Criptografía Para Principiantes

(Primera Versión)

José de Jesús Angel Angel

jesus@seguridata.com

Objetivo: *Este artículo tiene como propósito explicar las herramientas de seguridad informática más comunes, tratando de enfatizar la importancia de la criptografía, y dando una explicación lo más sencillo posible. Para un estudio más completo se recomienda ver la bibliografía.*

Indice:

0 Prefacio

El uso de técnicas criptográficas tiene como propósito prevenir algunas faltas de seguridad en un sistema computarizado. La seguridad en general debe de ser considerada como un aspecto de gran importancia en cualquier corporación que trabaje con sistemas computarizado. El hecho que gran parte de actividades humanas sea cada vez más dependiente de los sistemas computarizados hace que la seguridad juegue un papel importante [6].

Quizá antes sea importante mencionar algunos datos relacionados con la seguridad antes de comenzar con el desarrollo del tema

En el reporte reciente "Computer Crime Survey" del **FBI**, proporcionado por Secure Site E-News del 22 de mayo de 1999, de la compañía VeriSign, se dieron los siguientes datos:

Se estudiaron 521 compañías de varias ramas de la industria y de diferentes tamaños. Estas están actualmente trabajando para que su sistema computarizado sea seguro.

El 61% de estas compañías ha tenido experiencias de perdida debido al uso de no autorizado de su sistema computarizado.
El 32 % de estas organizaciones están usando ahora métodos de identificación segura en su sitio de internet.
El promedio de perdida de robo o perdida de información esta sobre $1.2 M de dólares.
El promedio de perdida por sabotaje esta sobre $1.1 M dólares.
El 50% de todas las compañías reportaron abuso de del uso de la red
El 94% de las organizaciones tiene actualmente un sitio en la web.

A la pregunta ¿qué tipo de tecnología de seguridad usa? Se contesto con lo siguiente:

Se cuenta con un control en el acceso, el 89%.
Cuenta con archivos cifrados, el 59%.
Cuanta con sistema de passwords, el 59%.
Usa Firewalls, el 88%.
Una sistema de log-in cifrados, el 44%.
Usa smart-cards, 37%.
Detención de intrusos, 40%.
Certificados digitales para la autenticación, 32%.

A la pregunta ¿ Cuál es más frecuente origen de un ataque?

Un "hacker" independiente, un 74%.
Un competidor, un 53%.
Un empleado disgustado, un 86%.
¿Su organización provee servicio de comercio electrónico?

Si, el 29%.

¿Su web-site ha tenido un acceso no autorizado en los últimos 12 meses?

Si, un 18%.
No, un 44%.
No sabe un 38%.

Enseguida damos un reporte que se tiene del tema en Europa, dado a conocer en unos cursos de criptografía industrial en Bélgica en junio de 1997. En donde se mide la frecuencia de incidentes de seguridad de la información relacionada con sus causas [7][8].

Frecuencia	**Razón**
50-60%	Errores debido a la inexperiencia, reacciones de pánico, mal uso,…
15-20%	Empleados disgustados, accidentes de mantenimiento,…
10-15%	Desastres naturales como inundaciones, incendios,…
3-5%	Causas externas: “hackers”

Otro aspecto importante a considerar es el crecimiento enorme que ha tenido la red internet, algunos datos importantes son los siguientes, proporcionados por Paul Van Oorschot de Entrust Technologies en una conferencia del ciclo The Mathematics of Public Key Cryptography en junio de 1999:

Se duplica el trafico de internet cada 100 días.
En enero de 1999 hubo 150 millones de personas en línea, 75 de ellas en USA.
El comercio sobre internet se duplica cada año.
Podría llegar a $1 trillón de dólares lo comercializado en internet en el año 2002.
A la radio le tomo 40 años, a la televisión 10 años para alcanzar 50 millones de usuarios a la red le ha tomado menos de 5.

Estos datos sólo son algunos de los que frecuentemente son dados a conocer por algún medio, y aunque algunos obedecen a intereses comerciales, lo que sí es verdadero es el enorme cambio que han tenido gran cantidad de actividades a raíz del uso de internet que incluso se ha considerado como el invento más importante de fin de siglo y de ahí lo primordial de todo lo relacionado con su seguridad.

Siempre podremos encontrar razones para reafirmar la trascendencia que tiene la seguridad en los sistemas computarizados, enseguida nos dedicamos a dar una introducción de cómo podemos atacar este problema.

El diseñar una estrategia de seguridad depende en general mucho de la actividad que se este desarrollando, sin embargo se pueden considerar los siguientes tres pasos generales: el **primero** crear una politica global de seguridad, el **segundo** realizar un análisis de riesgos y el **tercero** aplicar las medidas correspondientes [3][9][10].

Política global de seguridad: se debe de establecer el estatus de la información para la empresa o la organización, debe de contener un objetivo general, la importancia de la tecnología de la información para la empresa, el periodo de tiempo de validez de la política, los recursos con que se cuenta, objetivos especificos de la empresa.
Debe de establecerse la calidad de la información que se maneja según su objetivo, la calidad que debe tener la información quiere decir que se establezca cuando o para quien la información debe ser confidencial, cuando debe verificarse su integridad y cuando debe de verificarse su autenticidad tanto de la información como de los usuarios.

Análisis de riesgos: consiste en enumerar todo tipo de riesgos a los cuales esta expuesta la información y cuales son las consecuencias, los posibles atacantes entre persona empresas y dependencias de inteligencia, las posibles amenazas etc., enumerar todo tipo de posible perdida desde perdidas directas como dinero, clientes, tiempo etc., así como indirectas: créditos, perdida de imagen, implicación en un litigio, perdida de imagen, perdida de confianza etcétera.
El riesgo se puede calcular por la formula riesgo = probabilidad ×perdida, por ejemplo el riesgo de perder un contrato por robo de información confidencial es igual a la probabilidad de que ocurra el robo multiplicado por la perdida total en pesos de no hacer el contrato. El riesgo de fraude en transacciones financieras es igual a la probabilidad de que ocurra el fraude por la perdida en pesos de que llegara ocurrir ese fraude. Si la probabilidad es muy pequeña el riesgo es menor, pero si la probabilidad es casi uno, el riesgo puede ser casi igual a la perdida total. Si por otro lado la perdida es menor aunque la probabilidad de que ocurra el evento sea muy grande tenemos un riesgo menor. Por ejemplo la perdida de una transacción de 300 pesos con una probabilidad muy grande de que ocurra al usar criptografía débil, el riesgo llega a ser menor.
En el análisis de riesgo debe también incluirse los posibles ataques que puedan existir y su posible efectos.

Medidas de seguridad: esta parte la podemos plantear como la terminación de la toda la estructura de seguridad de la información. Una vez planteada una política de seguridad, decir cuanto vale la información, un análisis de riesgo, decir que tanto pierdo si le ocurre algo a mi información o que tanto se gana si se protege, debemos de establecer las medidas para que cumpliendo con la política de seguridad, las perdidas sean las menores posibles y que esto se transforme en ganancias ya sean materiales o de imagen.

Las posibles medidas que se pueden establecer se pueden dividir según la siguiente tabla:

tipos	**Protección Física**	**Medidas Técnicas**	**Medidas de Organización**
Preventivas	**PF**	**PT**	**PO**
Detectivas	**DF**	**DT**	**DO**
Correctiva	**CF**	**CT**	**CO**

PF: guardias a la entrada del edificio, control en el acceso de entrada, protección al hardware, respaldo de datos, …
DF: monitor de vigilancia, detector de metales, detector de movimiento, …
CF: respaldo de fuente de poder, …
PT: firewalls, criptografía, bitácora, …
DT: control de acceso lógico, sesión de autenticación, …
CT: programa antivirus, …
PO: cursos de actualización, organización de las claves, …
DO: monitoreo de auditoria, …
CO: respaldos automáticos, plan de incidentes (sanciones), …

En resumen debemos de mencionar que no existe un sistema computarizado que garantice al 100% la seguridad de la información debido a la inmensa mayoría de formas diferentes con que se puede romper la seguridad de un sistema [2]. Sin embargo una buena planeación de la estrategia para dar seguridad a la información puede resultar desde la salvación de una empresa hasta la obtención de grandes ganancias directas en pesos, o como ganancias indirectas mejorando la imagen y la seguridad de la empresa. Uno de los objetivos principales de estableces una política de seguridad es de reducir al mínimo los riegos posibles, implementando adecuadamente las diferentes medidas de seguridad [1][4][5].

Enseguida repasamos algunas de las técnicas de seguridad que pertenecen a la criptografía, se pretende exponer de una forma simple algunas de las partes mas conocidas de este amplio campo, para un estudio más profundo se puede recurrir a la amplia bibliografía.

1 Introducción

La palabra criptografía proviene del griego *kryptos*, que significa esconder y *gráphein*, escribir, es decir, escritura escondida. La criptografía ha sido usada a través de los años para mandar mensajes confidenciales cuyo propósito es que sólo las personas autorizadas lo puedan entender el mensaje.

Alguien que quiere mandar información confidencial aplica técnicas criptográficas para poder “esconder” el mensaje (lo llamaremos cifrar o encriptar), manda el mensaje por una línea de comunicación que se supone insegura y después solo el receptor autorizado pueda leer el mensaje “escondido” (lo llamamos descifrar o desencriptar)

CIFRAR **ENVIAR** **DESCIFRAR**

Desde sus inicios, la criptografía llegó a ser una herramienta muy usada en el ambiente militar, en la segunda gran guerra tuvo un papel determinante, una de las máquinas de cifrado y que tubo gran popularidad se llamó **ENIGMA**. Al terminar la guerra las agencias de seguridad de las grandes potencias invirtieron muchos recursos para su investigación. La criptografía como la conocemos hoy surgió con la invención de la computadora. Una buena referencia sobre la historia de la criptografía desde sus inicios hasta la 2° gran guerra se puede encontrar en [22], y en [19] se puede encontrar algo de la historia de la posguerra. También se pueden consultar [21][41].

La criptografía actual se inicia en la segunda mitad de la década de los años 70. No es hasta la invención del sistema conocido como **DES** (**D**ata **E**ncryption **S**tandard) en 1976 que se da a conocer mas ampliamente, principalmente en el mundo industrial y comercial. Posteriormente con el sistema **RSA** (**R**ivest, **S**hamir, **A**dleman) en 1978, se abre el comienzo de la criptografia en un gran rango de aplicaciones: en transmisiones militares, en transacciones financieras, en comunicación de satélite, en redes de computadoras, en líneas telefónicas, en transmisiones de televisión etcétera.

La criptografía se divide en dos grandes ramas, la criptografía de clave privada o simétrica y la criptografía de clave pública o asimétrica, **DES** pertenece al primer grupo y **RSA** al segundo. Las referencias más conocidas sobre criptografía de carácter general son [29][38][43][67][68][70][72][73][76].

Para poder entender un poco de la criptografía, es tiempo de plantear que tipo de problemas resuelve ésta. Los principales problemas de seguridad que resuelve la criptografía son: la privacidad, la integridad, la autenticación y el no rechazo.

La privacidad, se refiere a que la información sólo pueda ser leída por personas autorizadas.
Ejemplos: Si la comunicación se establece por teléfono y alguien intercepta la comunicación o escucha la conversación por otra línea podemos afirmas que no existe

privacidad. Si mandamos una carta y por alguna razón alguien rompe el sobre para leer la carta, podemos decir que se ha violado la privacidad. En la comunicación por Internet es muy difícil estar seguros que la comunicación es privada, ya que no se tiene control de la línea de comunicación. Por lo tanto si ciframos (escondemos) la información cualquier intercepción no autorizada no podrá entender la información confidencial. Esto es posible si se usan técnicas criptográficas, en particular la privacidad se logra si se cifra el mensaje con un método simétrico.

La integridad, se refiere a que la información no pueda ser alterada en el transcurso de ser enviada.

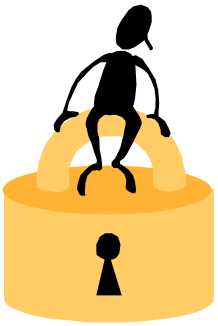
Ejemplos: Cuando compramos un boleto de avión es muy prudente verificar que los datos son los correctos antes de terminar la operación, en un proceso común esto se puede realizar al mismo tiempo de la compra, por internet como la compra se puede hacer desde dos ciudades muy distantes y la información tiene necesariamente que viajar por una línea de transmisión de la cual no se tiene control. Es muy importante estar seguros que la información transmitida no ha sido modificada (en tal caso se dice que hay integridad). Esto también se puede solucionar con técnicas criptográficas particularmente con procesos simétricos o asimétricos. La integridad es muy importante por ejemplo en las transmisiones militares ya que un cambio de información puede causar graves problemas.

La autenticidad, se refiere a que se pueda confirmar que el mensaje recibido haya sido mandado por quien dice lo mando o que el mensaje recibido es el que se esperaba.

Ejemplos: las técnicas necesarias para poder verificar la autenticidad tanto de personas como de mensajes usando quizá la más conocida aplicación de la criptografía asimétrica que es la firma digital, y de algún modo reemplaza a la firma autógrafa que se usa comúnmente, para autenticar mensajes se usa criptografía simétrica.

Por internet es muy fácil engañar a una persona con quien se tiene comunicación respecto a la identidad, resolver este problema es por lo tanto muy importante para efectuar comunicación confiable.

El no rechazo, se refiere a que no se pueda negar la autoría de un mensaje enviado.



Información Segura

Persona Autorizada

Cuando se diseña un sistema de seguridad una gran cantidad de problemas puedes ser evitados si se ponen en función de comprobar autenticidad, de garantizar privacidad, de asegurar integridad y evitar el no-rechazo.

La criptografía simétrica y asimétrica conjuntamente con otras técnicas, como el buen manejo de las claves y la legislación adecuada resuelven satisfactoriamente los anteriormente problemas planteados, como lo veremos en los capítulos posteriores.

2 Criptografía Simétrica

La criptografía simétrica se refiere al conjunto de métodos que permiten tener comunicación segura entre las partes siempre y cuando anteriormente se hayan intercambiado la clave correspondiente que llamaremos clave simétrica. La simetría se refiere a que las partes tienen la misma llave tanto para cifrar como para descifrar.

Este tipo de criptografía es conocida también como criptografía de clave privada o criptografía de llave privada.

Existe una clasificación de este tipo de criptografía en tres familias, la criptografía simétrica de bloques (block cipher), la criptografía simétrica de lluvia (stream cipher) y la criptografia simétrica de resumen (hash functions). Aunque con ligeras modificaciones un sistema de criptografía simétrica de bloques puede modificarse para convertirse en alguna de las otras dos formas, e inversamente, sin embargo es importante verlas por separado dado que se usan en diferentes aplicaciones.

La criptografía simétrica ha sido la más usada en toda la historia, ésta a podido ser implementada en diferente dispositivos, manuales, mecánicos, eléctricos, hasta los algoritmos actuales que son programables en cualquier computadora. La idea general es

aplicar diferentes funciones al mensaje que se quiere cifrar de tal modo que solo conociendo una clave pueda aplicarse de forma inversa para poder así descifrar.

Aunque no existe un tipo de diseño estándar, quizá el más popular es el de Fiestel, que consiste esencialmente en aplicar un número finito de interacciones de cierta forma, que finalmente da como resultado el mensaje cifrado. Este es el caso del sistema criptográfico simétrico más conocido, **DES**.

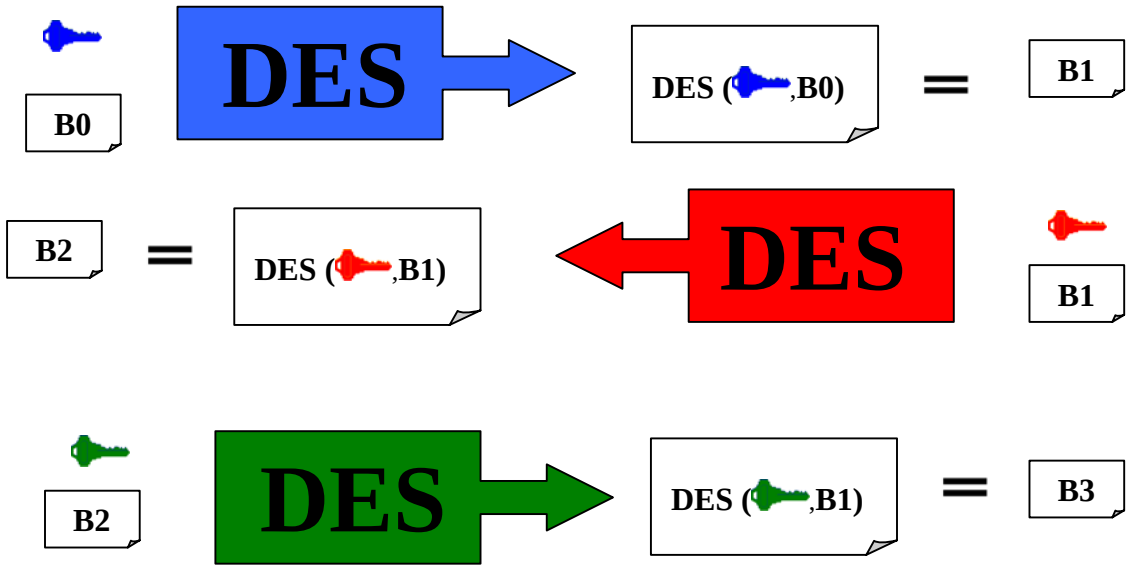
DES [47] es un sistema criptográfico que toma como entrada un bloque de 64 bits del mensaje y este se somete a 16 interacciones, con una clave de 56 bits. Para ver detalladamente su funcionamiento se puede consultar [29] o [52]. Este sistema fue tomado como estándar y ha sido uno de los mas conocidos, usados y estudiados.

DES opera con una llave de longitud de 56 bits, en la práctica el bloque de la clave tiene 64 bits, ya que a cada conjunto de 7 bits se le agrega un bit que puede ser usada como de paridad, pero en si la clave solo tiene 56 bits de longitud. Dependiendo de la naturaleza de la aplicación DES tiene 4 modos de operación [48][54][56] para poder implementarse: **ECB** (**E**lectronic **C**odebook **M**ode) para mensajes cortos, de menos de 64 bits, **CBC** (**C**ipher **B**lock **C**haining **M**ode) para mensajes largos, **CFB** (**C**ipher **B**lock **F**eedback) para cifrar bit por bit ó byte por byte y el **OFB** (**O**utput **F**eedback **M**ode) el mismo uso pero evitando propagación de error.

En la actualidad no se ha podido romper el sistema **DES** desde la perspectiva de poder deducir la clave simétrica a partir de la información interceptada, sin embargo con un método a fuerza bruta, es decir, probando todas las 2^56 posibles claves se ha podido romper **DES** en Enero de 1999. Lo anterior quiere decir que, es posible verificar todas las claves posibles en el sistema **DES** en un tiempo corto, lo que lo hace inseguro para propósitos de alta seguridad. La opción que se ha tomado para poder suplantar a **DES** ha sido usar lo que se conoce como cifrado múltiple, es decir aplicar varias veces el mismo algoritmo para fortalecer la longitud de la clave, esto a tomado la forma de un nuevo sistema de cifrado que se conoce actualmente como **triple-DES** o **TDES**.

TDES El funcionamiento de **TDES** [53] consiste en aplicar 3 veces **DES** de la siguiente manera: la primera vez se usa una clave **K1**(azul) junto con el bloque B0, de forma ordinaria **E** (de Encryption), obteniendo el bloque B1. La segunda ves se toma a B1 con la clave **K2** (roja), diferente a **K1** de forma inversa, llamada **D** (de Desencryption) y la tercera vez a B2 con una clave **K3** (verde) diferente a **K1** y **K2**, de forma ordinaria **E** (de Encryption), es decir, aplica de la interacción 1 a la 16 a B0 con la clave **K1**, después aplica de la 16 a la 1, a B1 con la clave **K2**, finalmente aplica una vez mas de la 1 a la 16 a B3 usando la clave **K3**, obteniendo finalmente a B3. En cada una de estas tres veces aplica el modo de operación más adecuado.

El proceso del cifrado con **TDES** se puede apreciar en las siguientes figuras:



Este sistema **TDES** usa entonces una clave de 168 bits, aunque se ha podido mostrar que los ataques actualmente pueden romper a **TDES** con una complejidad de 2^112, es decir efectuar al menos 2^112 operaciones para obtener la clave a fuerza bruta, además de la memoria requerida [44].

Se optó por TDES ya que es muy fácil interoperar con **DES** y proporciona seguridad a mediano plazo.

En los últimos 20 años se han diseñado una gran cantidad de sistemas criptográficos simétricos, entre algunos de ellos están: **RC-5** [37], **IDEA [**25], **FEAL** [40], **LOKI'91** [16], **DESX** [33], **Blowfish** [39], **CAST** [11]**, GOST** []18**,** etcétera. Sin embargo no han tenido el alcance de **DES**, a pesar de que algunos de ellos tienen mejores propiedades.

Podemos decir que el estado actual de la criptografía simétrica es la búsqueda de un nuevo sistema que pueda reemplazar a **DES** en la mayor parte de aplicaciones. Es así como se ha optado por convocar a un concurso de sistemas criptográficos simétricos y que este decida quien será el nuevo estándar al menos para los próximos 20 años.

AES

El **NIST** (**N**ational **I**nstitute of **S**tandards **T**echnology) [74] convocó a un concurso para poder tener un sistema simétrico que sea seguro y pueda usarse al menos en

los próximos 20 años como estándar. En la mitad del año de 1998 se aceptaron 15 candidatos, estos se han sometido a pruebas públicas y por parte del **NIST.** Actualmente se cuentan con 5 finalistas que son: **MARS**, **RC6**, **Rijndael**, **Serpent**, y **Twofish**, se espera que el candidato elegido se tenga a mediados del año 2000.

Las principales características que se pide a **AES** son que al menos sea tan seguro y rápido como **TDES**, es decir, que al menos evite los ataques conocidos. Además de que pueda ser implementado en una gran parte de aplicaciones. Una vez designado **AES** este podrá ser usado tanto como cifrador de bloques (block cipher), como cifrador de lluvia (stream cipher), como función resumen (hash function), y en el generador de números seudoaleatorios.

Los cifradores de lluvia o stream ciphers, son usados donde se cuanta con un ancho de banda restringido (el número de bits que se transmiten a la vez), además de que se requiere independencia en los bloques transmitidos, entonces la mejor opción es cifrar bit por bit o byte por byte, este tipo de cifradores tiene la característica además de ser muy rápido. Los algoritmos más conocidos de este tipo están **RC-4**, **SEAL** [66] y **WAKE.**

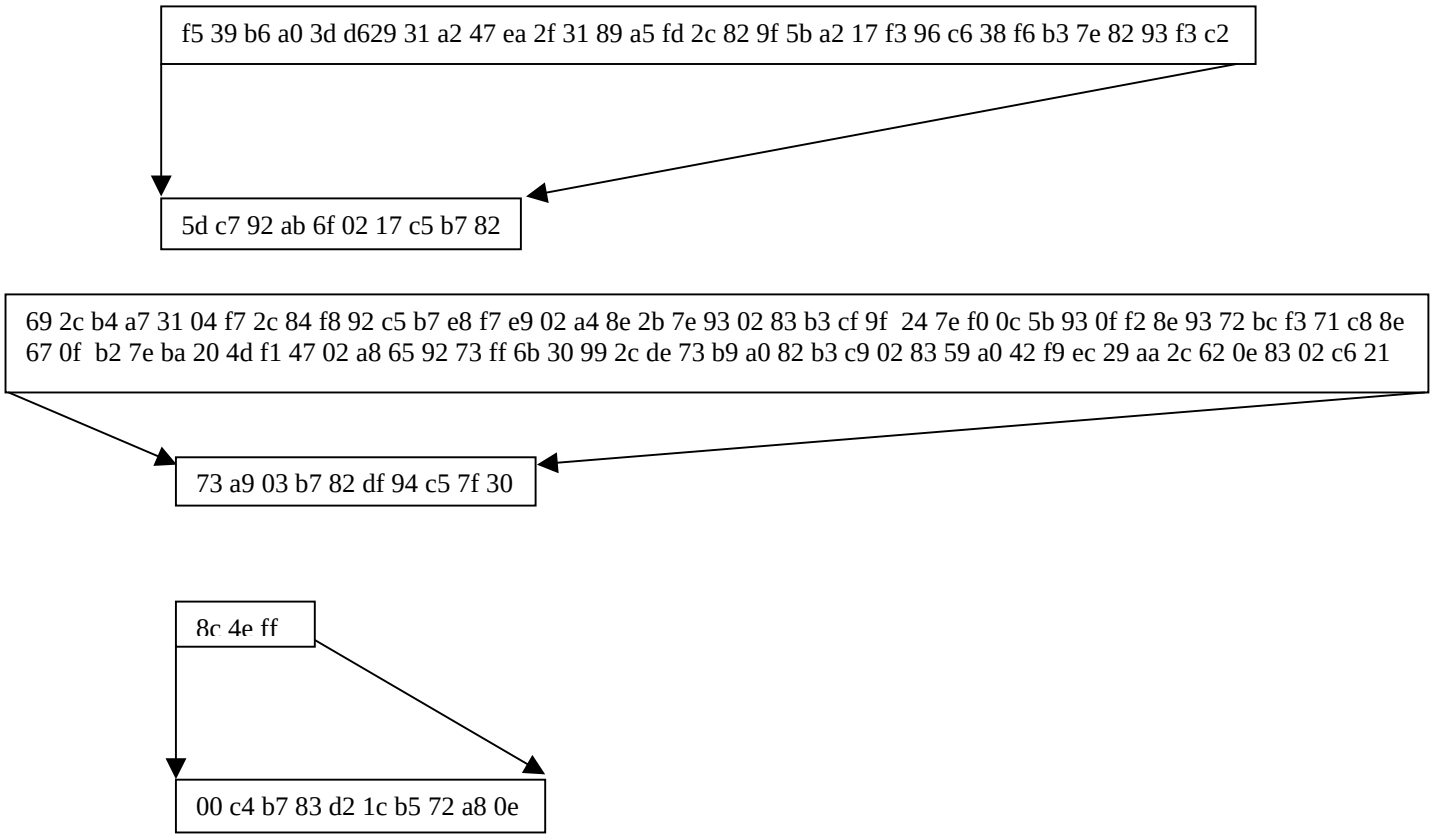
Entre los ataques más potentes a la criptografía simétrica están el criptoanálisis diferencial [12] y lineal [28], sin embargo no han podido ser muy eficientes en la práctica por lo tanto, por el momento después de que un sistema criptografíco es publicado y se muestra inmune a estos dos tipos de ataques (y algunos otros más) la mayor preocupación es la longitud de las claves [26].

Funciones Hash

Una herramienta fundamental en la criptografía son las funciones hash [60], son usadas principalmente para resolver el problema de la integridad de los mensajes, así como la autenticidad de mensajes y de su origen.

Una función hash es también ampliamente usada para la firma digital, ya que los documentos a firmar pueden ser en general demasiado grandes la función hash les asocia una cadena de longitud 160 bits que son mas manejables para el propósito de firma digital.

De forma gráfica la función hash efectúa lo siguiente:



Esto es, un mensaje de longitud arbitraria lo transforma de forma “única” a un mensaje de longitud constante.

¿Cómo hace esto?

La idea general es la siguiente:

La función hash toma como entrada una cadena de longitud arbitraria, digamos 5259 bits, luego divide este mensaje en pedazos iguales, digamos de 160bits, como en este caso y en general el mensaje original no será un múltiplo de 160, entonces para completar un número entero de pedazos de 160 bits al último se le agrega un relleno, digamos de puros ceros. En nuestro caso en 5259 caben 32 pedazos de 160 bits y sobran 139, entonces se agregaran 21 ceros más.

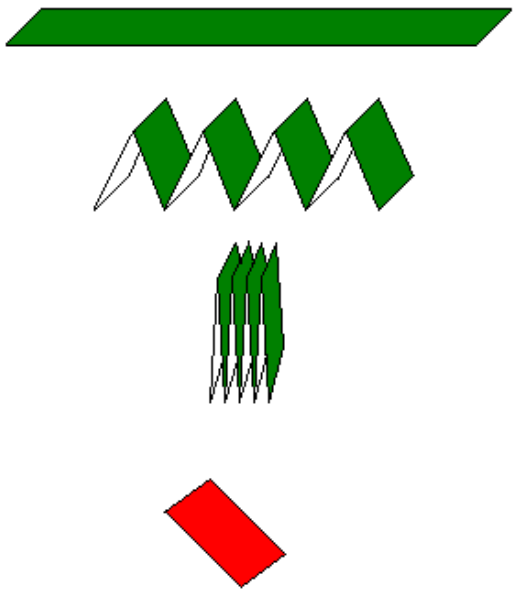
Entonces el mensaje toma la forma $x=x1, x2, x3,\ldots,xt$ donde cada xi tiene igual longitud (160bits por ejemplo).

Posteriormente se asocia un valor constante a un vector de inicialización *IV*, y se efectúan las siguientes interacciones

$$H_0 = IV$$
$$H_i = f(H_{i-1}, x_i) \quad 1 \le i \le t$$
$$h(x) = g(H_t)$$

donde *f* es una función que combina a dos cadenas de bits de longitud igual y fija, y *g* es una función de salida.

De alguna forma lo que se hace es tomar el mensaje partirlo en pedazos de longitud constante y combinar de alguna forma pedazo por pedazo hasta obtener un solo mensaje de longitud fija como muestra la figura siguiente:



Las funciones hash (o primitivas hash) pueden operar como: **MDC** (**M**odification **D**etection **C**odes) ó **MAC** (**M**essage **A**uthentication **C**odes) [57][64].

Los **MDC** sirven para resolver el problema de la integridad de la información, al mensaje se le aplica un **MDC** (una función hash) y se manda junto con el propio mensaje, al

recibirlo el receptor aplica la función hash al mensaje y comprueba que sea igual al hash que se envió antes.
Los **MDC**s son usados principalmente para resolver el problema de la integridad y lo hacen tomando el razonamiento siguiente:

Se aplica un hash $h(\mathbf{M})$ al mensaje **M** y se envía con el mensaje, cuando se recibe (**M**, $h(\mathbf{m})$) se le aplica una vez más el hash (que es público a **M**) obteniendo $h'(\mathbf{m})$, si $h(\mathbf{M})=h'(\mathbf{M})$, entonces se puede aceptar que el mensaje se a transmitido sin alteración

Los **MAC** sirven para autenticar el origen de los mensajes (junto con la integridad), un **MAC** es un mensaje junto con una clave simétrica que se les aplica un hash y se manda, al llegar la autenticidad del origen del mensaje se demuestra si la clave del receptor corresponde a la que se creo en el origen del mensaje.
Los **MAC**s son usados para resolver el problema de autenticar el origen del mensaje y tienes el argumento siguiente:

Se combina el mensaje **M** con una clave privada **K** y se les aplica un hash $h(\mathbf{M,K})$, si al llegar a su destino $h(\mathbf{M, K})$ se comprueba de integridad de la clave privada **K**, entonces se demuestra que el origen es solo el que tiene la misma clave **K**, probando así la autenticidad del origen del mensaje.

Las propiedades que deben de tener las primitivas hash son:

1) **Resistencia a la preimagen**: significa que dada cualquier imagen y, es computacionalmente imposible encontrar un mensaje *x*, tal que $h(x)=y$. Otra forma como se conoce esta propiedad es que *h* sea de un solo sentido.
2) **Resistencia a una 2° preimagen**: significa que dado *x*, es computacionalmente imposible encontrar una *x*' tal que $h(x)=h(x')$. Otra forma de conocer esta propiedad es que h sea resistente a una colisión suave.
3) **Resistencia a colisión**: significa que es computacionalmente imposible encontrar dos diferentes mensajes *x*, *x*' tal que $h(x)=h(x')$. Esta propiedad también se conoce como resistencia a colisión fuerte.

Para ilustrar la necesidad de estas propiedades veamos los siguientes ejemplos:

Consideremos un esquema de firma digital con apéndice, entonces la firma se aplica a $h(x)$, en este caso *h* debe ser un **MDC** con resistencia a una 2° preimagen, ya que de lo contrario un atacante **C** que conozca la firma sobre $h(x)$, puede encontrar otro mensaje *x*' tal que $h(x) = h(x')$ y reclamar que la firma es del documento *x*'.

Si el atacante **C** puede hacer que el usuario firme un mensaje, entonces el atacante puede encontrar una colisión (*x*, *x*') (en lugar de lo más difícil que es encontrar una segunda preimagen de *x*) y hacer firmar al usuario a *x* diciendo que firmo *x*'. En este caso es necesaria la propiedad de resistencia a colisión.

Por último si (*e*,*n*) es la clave pública **RSA** de **A**, **C** puede elegir aleatoriamente un *y* y calcular *z* = *y*^*e* mod *n*, y reclamar que *y* es la firma de *z*, si **C** puede encontrar una preimagen *x* tal que *z* = *h*(*x*), donde *x* es importante para **A**. Esto es evitable si *h* es resistente a preimagen.

Las funciones hash más conocidas son las siguientes: las que se crean a partir de un block cipher como **DES** [29], **MD5** [62], **SHA-1**, y **RIPEMD** [65].
Actualmente se ha podido encontrar debilidades en las funciones hash que tienen como salida una cadena de 128 bits, por lo que se ha recomendado usar salidas de 160bits. Así mismo se han encontrado ataques a **MD5** y **SHA-0** (antecesora de **SHA-1**), esto ha dado lugar que se dirija la atención sobre la función has **RIPEMD**.

3 Criptografía Asimétrica

La criptografía asimétrica es por definición aquella que utiliza dos claves diferentes para cada usuario, una para cifrar que se le llama clave pública y otra para descifrar que es la clave privada. El nacimiento de la criptografía asimétrica se dio al estar buscando un modo más práctico de intercambiar las llaves simétricas Diffie y Hellman [20], proponen una forma para hacer esto, sin embargo no fue hasta que el popular método de **R**ivest **S**hamir y **A**dleman **RSA** publicado en 1978 [36], cuándo toma forma la criptografía asimétrica, su funcionamiento esta basado en la imposibilidad computacional de factorizar números enteros grandes.

Actualmente la Criptografía asimétrica [32] es muy usada, sus dos principales aplicaciones son precisamente el intercambio de claves privadas [50] y la firma digital, una firma digital se puede definir como una cadena de caracteres que se agrega a un archivo digital que hace el mismo papel que la firma convencional que se escribe en un documento de papel ordinario. Los fundamentos de la criptografía asimétrica pertenecen a la teoría de números, algo de esto lo podemos ver en [23][24][34].

En la actualidad la criptografía asimétrica o de clave pública se divide en tres familias, según el problema matemático del cual basan su seguridad. La primera familias la que basa su seguridad en el **P**roblema de **F**actorización **E**ntera **PFE** [15], los sistemas que pertenecen a esta familia son, el sistema **RSA**, y el de **R**abin **W**illiams **RW** [46]. La segunda familia es la que basa su seguridad en el **P**roblema del **L**ogaritmo **D**iscreto **PLD**, a esta familia pertenece el sistema de **D**iffie **H**ellman **DH** de intercambio de claves y el sistema **DSA** [55] de firma digital. La tercera familia es la que basa su seguridad en el **P**roblema del **L**ogaritmo **D**iscreto **E**líptico **PLDE**, en este caso hay varios esquemas tanto de intercambio

de claves como de firma digital que existen como el **DHE** (Diffie Hellman Elíptico), **DSAE**, (**N**yberg-**R**ueppel) **NRE**, (**M**enezes, **Q**u, **V**anstone) **MQV** [30], etcétera.

Aunque a las familias anteriores pertenecen los sistemas asimétricos más conocidos, existen otro tipo de sistemas que basan su seguridad en otro tipo de problema como por ejemplo en el Problema del Logaritmo Discreto Hiperelíptico, sobre problemas de retículas y sobre subconjuntos de clases de campos numéricos reales y complejos.

RSA, en el caso de **RSA** [17] el problema matemático es el de la factorización de un número entero *n* grande (1024 bits), este número entero se sabe es producto de dos números primos *p,q* de la misma longitud, entonces la clave pública es el número *n* y la privada es *p,q*. El razonamiento del funcionamiento de **RSA** es el siguiente:

a) a cada usuario se le asigna un número entero *n*, que funciona como su clave pública
b) solo el usuario respectivo conoce la factorización de *n* (o sea *p,q*), que mantiene en secreto y es la clave privada

c) existe un directorio de claves públicas

d) si alguien quiere mandar un mensaje *m* a algún usuario entonces elige su clave pública *n* y con información adicional también pública puede mandar el mensaje cifrado **c**, que solo podrá descifrar el usuario correspondiente, el mensaje *m* convertido a número (codificación) se somete a la siguiente operación.

$$c = m^e \bmod n$$

e) Entonces el mensaje *c* puede viajar sin problema por cualquier canal inseguro

f) cuando la información cifrada llega a su destino el receptor procede a descifrar el mensaje con la siguiente fórmula

$$m = c^d \bmod n$$

g) Se puede mostrar que estas formulas son inversas y por lo tanto dan el resultado deseado, (*m*,*e*) son públicos y se pueden considerar como la clave pública, la clave privada es la pareja (*p*,*q*) o equivalentemente el número *d*. La relación que existe entre *d* y *e* es que uno es el inverso multiplicativo del otro módulo $\lambda(n)$ donde $\lambda(n)$ es el mínimo común múltiplo de *p*-1 y *q*-1, esto significa que la clave privada o el la pareja *p*,*q* o es el número *d*.

En términos muy generales es así como funciona el sistema **RSA**. Sin embargo en la realidad existen dos formas que son las más comunes, estas formas depende de la aplicación y se llaman el esquema de firma y el esquema de cifrado, cada una de estas dos diferentes aplicaciones consiste en una serie de pasos que a continuación se describen

Esquema de cifrado

Uso: este esquema se usa principalmente en cifrar claves de sistemas simétricos (claves de 128 bits aprox.)

1) Se toma el mensaje **M** (por ejemplo una clave simétrica de 128 bits), como en la practica actual es recomendable usar arreglos de longitud de 1024 bits, los complementa esos 128 bits con una serie de técnicas para obtener un arreglo de 1024

bits, que la computadora entiende como un número entero *m*, este proceso se llama codificación.

2) Se le aplica la formula de cifrado de **RSA** al entero *m*
3) Se envía el número entero *c*
4) Al recibir este número se aplica la formula de descifrado al entero *c* para obtener el entero *m*
5) Se decodifica m para obtener el mensaje **M**

Esquema de Firma Digital

Existen dos tipos de esquemas sobre firma digital, el que se denomina esquema de firma digital con apéndice [46][61] y el esquema de firma digital con mensaje recuperable. También cualquier esquema de firma cuenta con dos partes la primera parte se denomina proceso de firma (similar al cifrado) y la segunda parte proceso de verificación de la firma (similar al descifrado). Otros esquemas de firma digital se encuentran en [42].

El esquema más usado y conocido es el esquema de firma con apéndice y consiste en los siguientes puntos:

Proceso de Firma

1) El mensaje a firmar es *M*, se le aplica una función hash que reduce su longitud de forma única a un mensaje *H(M)* de longitud de 128 o 160 bits, lo que permite ver cualquier mensaje de cualquier longitud como una cadena de caracteres de longitud constante.

2) H(m) se somete también a un proceso de codificación, por lo tanto se obtiene un número h(m), al que se le aplica la formula con la potencia d, equivalentemente con la clave privada del firmante para obtener

$$s = h(m)^d \bmod n$$

3) Se envía entonces el mensaje firmado s

Proceso de Verificación

1) El que recibe s, se supone conoce el mensaje m, aplica la función para obtener
con la clave pública del que dice ser

$$h'(m) = s^e \bmod n$$

2) Aplica la función hash al mensaje *m* y si *h(m)=h'(m*) entonces acepta la firma

En un esquema con mensaje recuperable no es necesario saber el mensaje, después de que la firma es aceptada el mensaje puede recuperarse a partir de la firma.

Aspectos Importantes

1) La longitud de las claves

Existe una gran discusión [26], sobre este aspecto pero sin duda en la actualidad se acepta que es recomendable usar claves de longitud 768 para actividades personales,1024 bits para corporaciones y 2048 para actividades de alto riesgo. La longitud de las claves tiene que ver con la seguridad del sistema si el número *n* pudiese ser factorizado entonces sin mucha dificultad puede calcular a *d* a partir de *e*, *p*, y *q* por lo tanto descifrar cualquier mensaje. El último récord conocido sobre factorización de números enteros producto de dos primos es de 155 (512 bits) dígitos alcanzado en Jul de 1999.

2) La aleatoriedad de las claves

La generación de las claves **RSA** es muy importante, muchos ataques son evitados si las claves son elegidas de forma aleatoria [63], esto incrementara la seguridad del sistema.

3) método de codificación

El método que actualmente es usado para aplicaciones en el esquema de cifrado es el **OAEP** [], este resiste a los ataques que actualmente se conocen y el estándar más conocido sobre **RSA** es el **PKCS**#1 v.2 de la RSA Data Security.
En el caso de Esquemas de firma digital el método de codificación recomendable es **PSS** [], que esta descrito en **PKCS**#1 v 2.1

4) Elección de parámetros

La elección adecuada de los parámetros que se usan aumenta la seguridad del sistema asi como su fácil y rápida implementación. Como elegir a e=65537, que es el número 4 de Fermat. Esto implica que *d*, la clave privada sea de una longitud considerable, evitando el ataque de Wiener [45]. Por otro lado usar el método de descifrado por el teorema chino del residuo aumenta la rapidez de descifrado.

CCE otro tipo de criptografía de clave pública es el que usa curvas elípticas definidas en un campo finito. La diferencia que existe entre este sistema y **RSA** es el problema del cual basan su seguridad, mientras **RSA** razona de la siguiente manera: te doy el número 15 y te reta a encontrar los factores primos. El problema del cual están basados los sistemas que usan curvas elípticas que denotaremos como **CCE** es el problema del logaritmo discreto elíptico, en este caso su razonamiento con números sería algo como: te doy el número 15 y el 3 y te reta a encontrar cuantas veces tienes que sumar el mismo 3 para obtener 15.

En lo que sigue nos dedicaremos a explicar un poco mas lo más importante de los **CCE**

1) entenderemos como una curva elíptica a un conjunto finito de puntos *P,Q,…,S* donde cada punto en una pareja *P* = (*x,y*) y las coordenadas *x,y* satisfacen una ecuación de la siguiente forma:

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

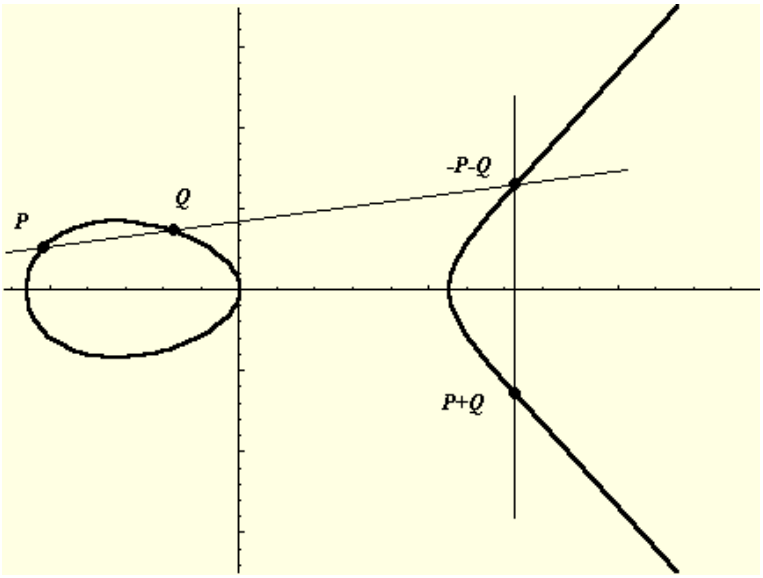
Donde las constantes *a,b,c,d* y e pertenecen a cierto conjunto llamado campo **F,** que para propósitos de la criptografía o es un campo primo (Z_p) o un campo de característica 2, o sea donde los elementos son n-adas de ceros y unos (F_2^n)

2) El conjunto de puntos que satisfacen a una ecuación similar a la de 1) lo podemos representar como

$$E : O, P_1, P_2, P_3, \ldots, P_n$$

Este conjunto de puntos puede sumarse y tiene las mismas propiedades que la suma de los números enteros, es decir lo que se conoce como un grupo abeliano, hay que hacer notar que en este caso el que hace el papel de cero (identidad aditiva) es un punto especial que no tiene coordenadas y se representa como **O** llamado punto al infinito

3) La suma de estos puntos tiene una explicación geométrica muy simple, en este caso la gráfica representa a todos los puntos que satisfacen la ecuación de 1), si suponemos que queremos sumar a *P* y *Q*, trazamos una línea recta que pase por *P* y *Q*, la ecuación de 1) es de grado 3 y la línea de grado 1, entonces existe siempre tres soluciones, en este caso la tercera solución esta dibujada como el punto *-P-Q*, enseguida se procede a dibujar una línea recta paralela al eje Y que pase por *-P-Q*, esta línea vertical también intercepta tres veces a la recta, todas las líneas verticales interceptan al punto especial llamado infinito y que geométricamente esta en el horizonte del plano, el tercer punto es por definición *P+Q*, como se muestra en la figura



4) La anterior forma de sumar puntos de una curva elíptica es un poco extraña sin embargo, es esta extrañeza lo que permita que sea un poco mas difícil romper los **CCE**. En el área de las matemáticas conocida como teoría de grupos se sabe que estos grupos son muy simples llamados grupo finitos abelianos lo que permite también que los **CCE** sean fácil de implementar, llamaremos al número de puntos racionales de la curva como el orden de la curva

5) Los **CCE** basan su seguridad en el Problema del Logaritmo Discreto Elíptico (**PLDE**), esto quiere decir que dados P,Q puntos de la curva hay que encontrar un número entero x tal que $xP = Q$ ($xP = P+P+\ldots+P$, x veces). Obsérvese que a diferencia del **PFE** (Problema de Factorización Entera) el **PLDE** no maneja completamente números, lo que hace más complicado su solución.

6) La creación de un protocolo con criptografía de curvas elípticas requiere fundamentalmente una alta seguridad y una buena implementación, para el primer punto se requiere que la elección de la curva sea adecuada, principalmente que sea no-supersingular y que el orden del grupo de puntos racionales tenga un factor primo de al menos 160 bits, además de que este orden no divida al orden de un número adecuado de extensiones del campo finito [13][27], para que no pueda ser sumergido en él, si el campo es Z_p, se pide que la curva no sea anómala. Todo esto con el fin de evitar los ataques conocidos.
   Para el caso de la implementación hay que contar con buenos programas que realicen la aritmética del campo finito, además de buenos algoritmos que sumen puntos racionales, si el campo es Z_p existen varios y si el campo el F_2^n entonces se toma una base polinomial que tenga el mínimo de términos por ejemplo un trinomio para generar los elementos del campo finito esto si la implementación es en software y se toma una base normal si es en hardware. Además de contemplar que las operaciones de puntos racionales pueden hacerse en el espacio proyectivo esto elimina el hacer divisiones ahorrando tiempo.

7) Lo anterior se ve reflejado en las ventajas que ofrecen los **CCE** en comparación con **RSA**, la principal es la longitud de la clave secreta. Se puede mostrar que mientras en **RSA** se tiene que usar una clave de 1024 para ofrecer una considerable seguridad, los **CCE** solo usan 163 bits para ofrecer la misma seguridad, así también las claves **RSA** de 2048 son equivalentes en seguridad a 210 de **CCE**. Esto se debe a que para resolver el **PLDE** el único algoritmo conocido toma tiempo de ejecución totalmente exponencial, mientras que el algoritmo que resuelve **PFE** incluso también el **PLD** en Z_p toman tiempo subexponencial.

8) Otra buena noticia sobre los **CCE** es que los elementos de los puntos racionales pueden ser elementos de un campo finito de característica 2, es decir pueden ser arreglos de ceros y unos de longitud finita (01001101110010010111), en este caso es posible construir una aritmética que optimice la rapidez y construir un circuito especial para esa aritmética, a esto se le conoce como Base Normal Optima.

9) Lo anterior permite con mucho que los **CCE** sean idóneos para ser implementados en donde el poder de computo y el espacio del circuito sea reducido, donde sea requerida una alta velocidad de procesamiento o grandes volúmenes de transacciones, donde el espacio de almacenamiento, la memoria o el ancho de banda sea limitado. Lo que permite su uso en Smart Cards, Teléfonos celulares, Fax, Organizadores de Palma, PCs, etcétera.

10) En la actualidad existen varios estándares que permiten el uso adecuado y óptimo de los **CCE**, entre los cuales se encuentran: **IEEE P1363** [75] (Institute of Electrical and Electronics Engineers), el **ANSI X9.62**, **ANSI X9.63**, **ANSI TG-17**, **ANSI X12** (American National Standards Institute), **UN/EDIFACT**, **ISO/IEC 14888**, **ISO/IEC 9796-4**, **ISO/IEC 14946** (International Standards Organization), **ATM** Forum (Asynchronous Transport Mode), **WAP** (Wireless Application Protocol). En comercio electrónico: **FSTC** (Financial Services Technology Consortion), **OTP 0.9** (Open Trading Protocol), **SET** (Secure Electronic Transactions). En internet **IETF** (The Internet Engineering Task Force), **IPSec** (Internet Protocol Security Protocol)

11) Los **CCE** están reemplazando a las aplicaciones que tienen implementado **RSA**, estas definen también esquemas de firma digital, Intercambio de claves simétricas y otros. Los **CCE** se pueden estudiar en [14][31][35][69][71].

3) Otras Herramientas criptográficas

En esta sección me dedicare principalmente a enumerar otro tipo de herramientas o técnicas que son usadas en criptografía, cada una de ellas tiene una gran aplicación y tienen un propósito muy especifico dentro del ámbito de la criptografía, sin embargo su descripción completa no es el propósito para un lector novato así que solo se mencionarán, para un mayor estudio puede consultarse la bibliografía.

A) Compartición de Secretos

La compartición de secretos [99][100], como su nombre lo dice es una técnica criptográfica que se dedica a partir un secreto, que puede ser una clave secreta, en la responsabilidad de varias personas y que solo con el número mínimo de personas se podrá reconstruir el secreto compartido. Por ejemplo si el secreto es el número 100 y este debe ser compartido por tres personas A1, A2 y A3 una forma de poder hacerlo es generar un número aleatorio menor a 100, digamos el 33 posteriormente se genera otro número aleatorio menor a 100-33, digamos el 27, y finalmente la tercera parte será *100-(21+33)=46*. Así el secreto 100 esta compartido por A1(33), A2(27) y A3(46) cada quien con su parte correspondiente. Como ninguno de ellos sabe las otras partes, solo los tres juntos podrán reconstruir el mensaje sumando sus partes. Claro esta este es solo un ejemplo para explicar el concepto.

La comparición de secretos puede ser usada para compartir digamos la combinación de una caja fuerte, la clave de lanzamiento de algún proyectil, la clave secreta de una autoridad certificadora, la clave de activación de algún dispositivo de alto riesgo, etc.,

Uno de los mejores métodos de comparición de secretos y mas conocido es el esquema (n,k) límite de Shamir. Este método consiste en partir una clave K en n partes, y se tiene como mínimo (límite) el número k de partes para reconstruir la clave, es decir cualquiera k de los n custodios pueden reconstruir la clave K, pero ningún subgrupo de k-1 custodios podrá hacerlo.

El esquema límite de Shamir se basa en lo siguiente:

1) Se define el número de custodios t, digamos t=2
2) Se generan aleatoriamente los coeficientes necesarios para construir un polinomio de t-1 grado, en nuestro caso

$$f(x) = s + a_1 x$$

donde el coeficiente es aleatorio y s el secreto a compartir

3) Evidentemente el secreto se recupera conociendo el polinomio y evaluando en cero $s = f(0)$
4) Para nuestro caso las partes serán

$$s_1 = f(1) \qquad s_2 = f(2)$$

Osea

$$s_1 = s + a_1 \qquad s_2 = s + 2a_1$$

En general

$$s_i = f(i)$$

El método para recuperar el secreto *s*, es reconstruir el polinomio $f(x)$ a partir de t partes cualquiera, esto se hace por medio de la interpolación de Lagrange [].
En nuestro caso el secreto se puede reconstruir de la siguiente formula:
$s = c_1 y_1 + c_2 y_2$
donde

y_1, y_2

son las partes, o sea:

$$y_1 = f(1) = s + a_1, \qquad y_2 = f(2) = s + 2a_2$$

y

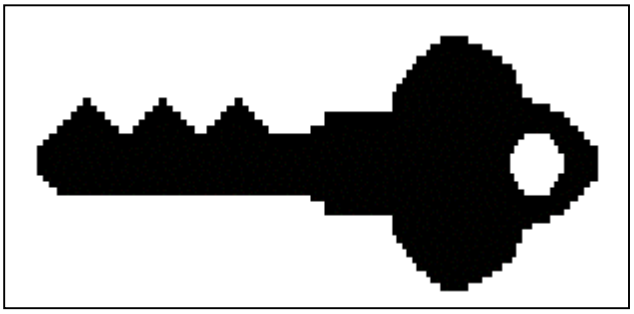
$$c_1 = \frac{2}{2-1}, \qquad c_2 = \frac{1}{1-2}$$

Por lo tanto

$$s = 2(s + a_1) - (s + 2a_1)$$

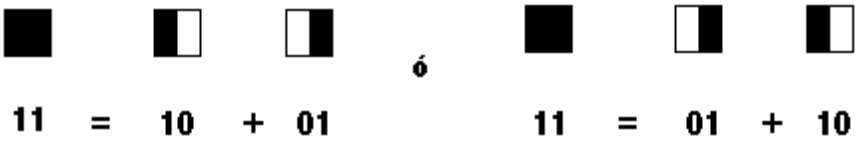
B) Criptografía Visual

Una idea ingeniosa de usar un método de comparición de secretos con un esquemas límite (*n*,*k*) es la criptografía visual [94][95][96][97][98], esto consiste en lo siguiente: una imagen es partida en *n* partes, y si se sobreponen al menos *k* de estas partes se puede reconstruir la imagen.
Veamos en ejemplo de un esquema (2,2), esto trabaja considerando que si la imagen es de banco y negro, entonces la imagen podrá ser un conjunto de cuadros completamente negros y completamente negros, por ejemplo la siguiente imagen



Ahora cada cuadro de la imagen podrá ser considerado como blanco o negro, equivalentemente con valores 0 y 1. Para partir esta imagen en dos partes *n*=2 y considerando el límite con *k*=2, se procede como sigue:

Cada cuadro que es completamente negro podrá ser partido en dos partes de la siguiente forma:

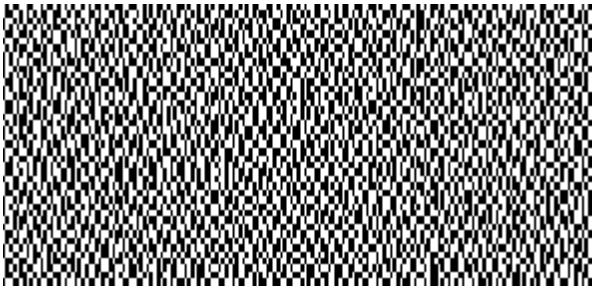


Y un cuadro completamente blando podrá ser partido en dos de la forma siguiente:

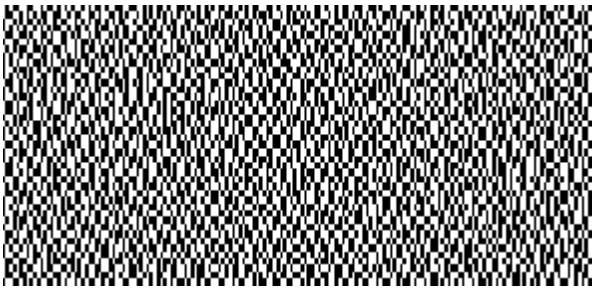
Que significa suma módulo 2, es decir 1+0=1, 0+1=1, 0+0=0 pero también 1+1=0, de este modo se pueden tomar cualquiera de las dos particiones de los cuadros de color blanco.

Para formar las dos partes de la figura en un acetato se elige aleatoriamente una de las combinaciones anteriores según se parta un cuadro blanco o uno negro

En el caso de nuestra figura una ves elegidas las partes, la figura partida en un esquema limite (2,2) queda así:



Parte 1



Parte 2

De esta forma se tiene partida la figura en dos partes y se recuperara solo sobreponiendo una sobre la otra.
Al sobreponer las dos partes se recupera la figura, de la siguiente forma:

En el caso general se parte los cuadros blancos y negros en *n* pedazos y hasta no tener *k* pedazos negros el cuadro reconstruido será siendo blanco, a partir de k pedazos negros hasta *n* el cuadro reconstruido será negro. En nuestro caso, un cuadro con solo la mitad negra será considerado blanco, es necesario que tenga dos mitades negras para que el cuadro reconstruido se considere negro, que es el caso del esquema (2,2).

C) Dinero Electrónico

Una aplicación más, que puede ser realidad gracias a la criptografía de clave pública es conocida como dinero electrónico [78], en términos sencillos el dinero electrónico es otra representación de lo que conocemos como dinero o valor, por ejemplo tenemos dinero en billetes emitidos por algún país, podemos tener cheques pagaderos en un banco, bonos, pagares pagaderos en algún plazo, en fin. El dinero electrónico es físicamente un número que se genera aleatoriamente se le asigna un valor, se cifra y firma y se envía al banco, ahí el banco valida el número y certifica el valor, y lo regresa al usuario firmado por el banco, entonces el usuario puede efectuar alguna transacción con ese billete electrónico.

= f5 6a 80 b2....03

Las principales propiedades del dinero electrónico son las siguientes:

1) **Independencia**: la seguridad del dinero digital no debe depender de la el lugar físico donde se encuentre, por ejemplo en el disco duro de una PC

2) **Seguridad**: el dinero digital (el número) no debe de ser usado en dos diferentes transacciones

3) **Privacidad**: el dinero electrónico debe de proteger la privacidad de su usuario, de esta forma cuando se haga una transacción debe de poder cambiarse el número a otro usuario sin que el banco sepa que dueños tuvo antes.

4) **Pagos fuera de línea**: el dinero electrónico no debe de depender de la conexión de la red, así un usuario puede transferir dinero electrónico que tenga en una “smart card” a

una computadora, el dinero digital debe ser independiente al medio de transporte que use.

5) **Transferibilidad**: el dinero electrónico debe de ser transferible, cuando un usuario transfiere dinero electrónico a otro usuario debe de borrarse la identidad del primero.

6) **Divisibilidad**: el dinero electrónico debe de poder dividirse en valores fraccionarios según sea el uso que se da, por ejemplo en valor de 100, 50 y 25

La serie de pasos que puede seguir una transacción que se realiza con dinero electrónico en un escenario simple es la siguiente:

Supóngase que el usuario **A** quiere mandar un cheque a **B**, usando ahora dinero electrónico.

1) **A** genera un número aleatorio grande N de digamos 100 dígitos y le da un valor digamos 1000 pesos
2) **A** cifra este número junto a su valor con su clave secreta asimétrica.
3) **A** firma este número y lo transmite a su banco.
4) El banco de **A** usa, la clave pública de **A** para descifrar el número y verificar la firma, así recibe la orden y sabe que es de **A**. El banco borra la firma de **A** del documento electrónico.
5) El banco revisa que **A** tenga en sus cuentas la cantidad pedida 1000 pesos y la debita de alguna de estas cuentas.
6) El banco firma el número que mando **A**, con el valor asignado de 1000 pesos
7) El banco regresa el número que ya es dinero a, **A**
8) **A** envía este dinero a **B**
9) **B** verifica la firma del banco de **A**, que esta en N
10) **B** envía N a su banco
11) EL banco de **B** re-verifica la firma del banco de **A** en N
12) El banco de **B** verifica que N no este en la lista de números “ya usados”
13) El banco de **B** acredita la cantidad de 1000 pesos a la cuenta de **B**
14) El banco de **B** pone a N en la lista de números “ya usados”
15) Finalmente el banco de **B** envía un recibo firmado donde establece que tiene 1000 pesos más en su cuenta

En el mundo comercial existen varias empresas privadas que proveen el servicio de dinero electrónico en diferentes modalidades entre ellas están: CheckFree, CyberCash, DigiCash, First Virtual, Open Market, NetBill y Netscape.

En http://www.ecashtechnologies.com/ pueden encontrarse algunos ejemplos interactivos de cómo trabaja el dinero electrónico en la práctica

5 Certificados digitales

Los certificados digitales [84][85], tienen una similitud con las licencias de conducir, las primeras permiten viajar por las carreteras, los certificados digitales permiten navegar por la red Internet, la principal característica es que da identidad al usuario y puede navegar con seguridad. De igual forma que la sola licencia de conducir o un pasaporte sirve para dar identidad a quien la porta en ciertos casos, el certificado digital da identidad a una clave pública y se comparta como una persona en el espacio cibernético.

El nacimiento del certificado digital fue a raíz de resolver el problema de administrar las claves públicas y que la identidad del dueño pudiera ser falsa. La idea es que una tercera entidad intervenga en la administración de las claves públicas y asegure que las claves públicas tengan asociado un usuario claramente identificado. Esto fue inicialmente planteado por Kohnfelder del MIT en su tesis de licenciatura [86].

Las tres partes más importantes de un certificado digital son:

1) Una clave pública
2) La identidad del implicado: nombre y datos generales,
3) La firma privada de una tercera entidad llamada autoridad certificadora que todos reconocen como tal y que válida la asociación de la clave pública en cuestión con el tipo que dice ser.

En la actualidad casi todas las aplicaciones de comercio electrónico y transacciones seguras requieren un certificado digital, se ah propagado tanto su uso que se tiene ya un formato estándar de certificado digital, este es conocido como X509 v. 3 [90]

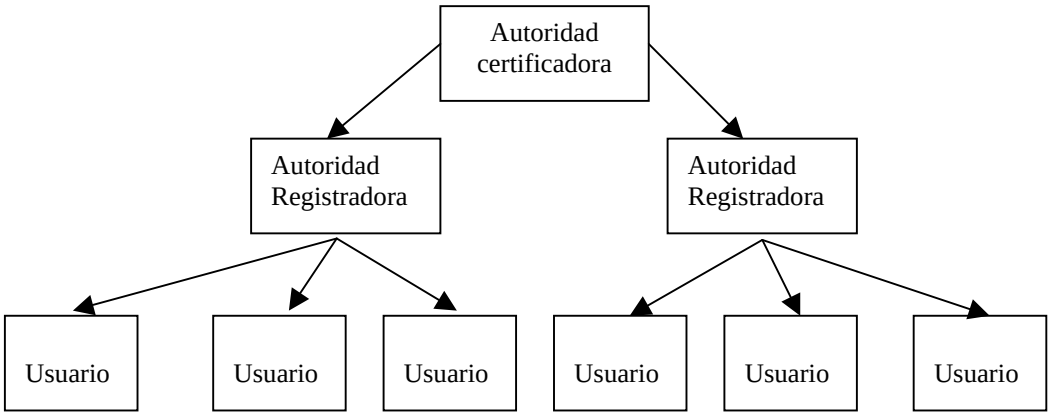
Algunos de los datos mas importantes de este formato son los siguientes:

Versión: 1,2 o 3
Número de Serie:
Emisor del Certificado: VeriMex
Identificador del Algoritmo usado en la firma: RSA, DSA o CE
Periodo de Validez: De Enero 2000 a Dic 2000
Sujeto: Jesús Angel
Información de la clave pública del sujeto: la clave, longitud, y demás parámetros
Algunos datos opcionales, extensiones que permite la v3
Firma de la Autoridad Certificadora

Un certificado digital entonces se reduce a un archivo de uno o dos k de tamaño, que autentica a un usuario de la red.

6 Infraestructura de claves públicas

Teniendo ya un certificado digital que es generado con la ayuda de un algoritmo de clave pública ahora el problema es como administración todos estos [51][58][59], la estructura más básica es la siguiente:



El papel de la Autoridad certificadora (**AC**) es de firmar los certificados digitales de los usuarios, generar los certificados, mantener el status correcto de los certificados, esto cumple el siguiente ciclo:

1) La generación del certificado se hace primero por una solicitud de un usuario, el usuario genera sus claves pública y privada y manda junto con los requerimientos de la solicitud su clave pública para que esta sea certificada por la **AC**.
2) Una vez que la **AR** (es la **AC** regional) verifica la autenticidad del usuario, la **AC** vía la AR firma el certificado digital y es mandado al usuario
3) El status del usuario puede estar en: activo, inactivo o revocado. Si es activo el usuario puede hacer uso del certificado digital durante todo su periodo válido
4) Cuando termina el período de activación del certificado el usuario puede solicitar su renovación.

Entre las operaciones que pudiera realizar una **AC** están:

Generar certificados
Revocar certificados
Suspender certificados
Renovar certificados
Mantener un respaldo de certificados…..

Entre las que pudiera realizar una AR están:

Recibir las solicitudes de certificación
Proceso de la autenticación de usuarios
Generar las claves
Respaldo de las claves
Proceso de Recobrar las claves
Reportar las revocaciones….

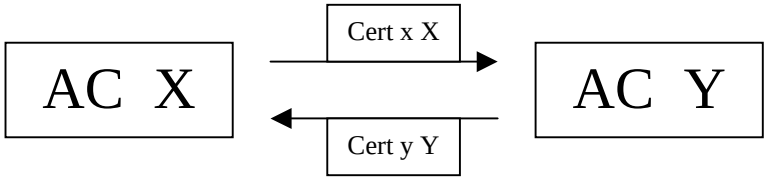
Y las actividades de los usuarios:

Solicitar el certificado
Solicitar la revocación del certificado
Solicitar la renovación del certificado….

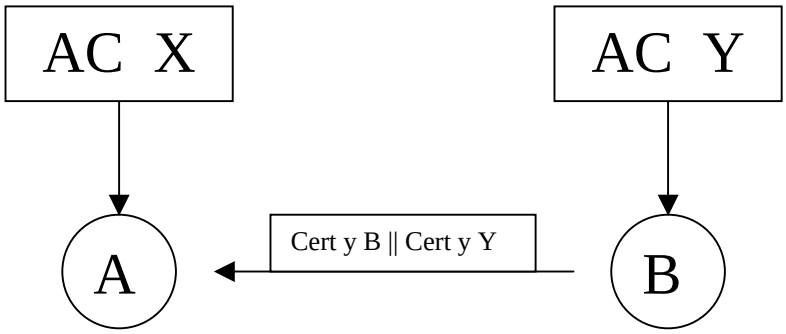
Una ves que algún usuario tiene un certificado digital este puede usarlo para poder navegar por la red con nombre y apellido en forma de bits, esto permite entrar al mundo del comercio electrónico, al mundo de las finanzas electrónicas y en general a la vida cibernética con personalidad certificada. El usuario dueño de un certificado digital tiene la potencialidad de poder autentificarse con cualquier otra entidad usuaria, también puede intercambiar información de forma confidencial y estar seguro de que esta es integra, así estar seguro que contactos vía el certificado digital no serán rechazados. Los primeros usuarios de certificados digitales fueron los servidores, actualmente son quienes mas los usan, sin embargo también se ha incrementado el número de personas que los usan.

Si suponemos que algún tipo de aplicación funciona ya con certificados digitales, esta tendrá una **AC** y las correspondientes **AR**, sin embargo es común que haya mas autoridades certificadoras y que sus usuarios puedan interoperar con sus respectivos certificados, a esto se le conoce como certificación cruzada y opera de la siguiente forma:

1) Las diferentes **AC** pueden estar certificadas enviándose una a otra sus respectivos certificados que ellas mismas generan



2) Entonces la **AC X** tendrá el certificado de la **AC Y** y viceversa, pudiendo generar un certificado para **Y** que genera **X** y otro para **X** que genera **Y**
3) Ahora como un usuario **A** de la **AC X** puede comunicarse con un usuario **B** de la **AC Y**



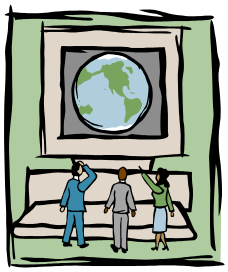
4) El usuario **B** envía a **A** el certificado de **B** que genera **Y** (**Cert y B**) junto con el certificado de **Y** que el mismo se genera (**Cert y Y**)
5) Ahora **A** puede validar a **B** (**Cert y B**) usando el certificado de **Y** que genera **X**

En la práctica se ha demostrado que el estatus de un certificado cambia con gran frecuencia, entonces la cantidad de certificados digitales revocados crece considerablemente, el problema esta en que cada vez que se piensa realizar una comunicación y es necesario validar un certificado se debe de comprobar que este no esta revocado. La solución que se ha venido usando es la de crear una lista de certificados revocados **LCR** y así verificar que el certificado no esta en esa lista, para poder iniciar la comunicación. El manejo de las listas de certificados revocados ha llegado a tener un gran costo que sin embargo aún no se ha reemplazar por otra técnica a pesar que se han propuesto ya salidas al problema.

Las operaciones de la administración de los certificados digitales puede cambiar de acuerdo a las leyes particulares de cada país o entidad. Más información sobre la infraestructura de certificados digitales se puede encontrar en [87][89][90][91][92][93].

7 Comercio electrónico

Hoy en día, gran parte de la actividad comercial ha podido transformarse gracias a redes de conexión por computadoras como Internet, esta transformación facilita hacer transacciones en cualquier momento de cualquier lugar del mundo. Todo lo que esta alrededor de esta nueva forma de hacer negocios es lo que se ha llamado comercio electrónico, sin duda la gran variedad de actividades que giraban alrededor del quehacer comercial se han tenido que conjuntar con las nuevas técnicas cibernéticas. Así hoy tanto un comerciante, un banquero, un abogado o una matemático puede hablar de comercio electrónico enfocándose a la parte que le corresponde.



Existen diferentes niveles de hacer comercio electrónico, y su clasificación aún esta por formarse, sin embargo, la parte más visible es la que cualquier usuario en una computadora personal puede ver, esto es hacer comercio electrónico se convierte a comprar o vender usando una conexión por internet en lugar de ir a la tienda. La forma de hacer esto es muy similar a lo que tradicionalmente se hace, por ejemplo: en la tienda uno entra al establecimiento, de forma electrónica se prende la computadora y una ves conectado a internet entra a la página del negocio, enseguida un comprador revisa los productos que posiblemente compre y los coloca en una carrito, de la misma forma en la computadora se navega por la página del negocio y con el browser se revisan los productos que éste vende, al escoger éstos se colocan en un carrito virtual, que no es nada mas que un archivo del usuario. Una vez elegido bien los productos de compra se pasa a la caja, donde se elige un sistema de pago y se facturan los productos al comprador. De forma similar en la computadora se pueden borrar productos que no se quieren comprar o añadir nuevos, una

ves elegidos éstos se procede a una parte de la pagina que toma los datos y solicita el método de pago, generalmente se lleva a cabo con tarjeta de crédito.

En la parte tradicional de comprar al pagar en la caja termina el proceso, en la parte por computadora aún tiene que esperarse que sean enviados los productos. A pesar de esto las ventajas que ofrece el comercio electrónico son magníficas, ya que es posible comprar en un relativo corto tiempo una gran cantidad de productos sin necesidad de moverse de lugar, es decir al mismo tiempo se puede comprar una computadora, un libro, un regalo, una pizza, hacer una transacción bancaria etc., de la forma tradicional se llevaría al menos un día completo y eso si los negocios esta en la misma ciudad, si no, el ahorro de tiempo que representa comprar por internet es incalculable.

Al efectuar una operación comercial por internet se presentan nuevos problemas, por ejemplo cómo saber que la tienda virtual existe verdaderamente, una vez hecho el pedido cómo saber que no se cambia la información, cuando se envía el número de tarjeta de crédito cómo saber si este permanecerá privado, en fin, para el comerciante también se presentan problemas similares, cómo saber que el cliente es honesto y no envia información falsa, etc. Todos estos problemas pueden ser resueltos de manera satisfactoria si se implementan protocolos de comunicación segura usando criptografía. En la siguiente sección nos dedicamos a describir como es que estos protocolos resuelven los problemas planteados.

8 Protocolos de seguridad

Un protocolo de seguridad es la parte visible de una aplicación, es el conjunto de programas y actividades programadas que cumplen con un objetivo especifico y que usan esquemas de seguridad criptográfica.

El ejemplo mas común es **SSL** (**S**ecure **S**ockets **L**ayer) (que vemos integrado en el Browser de Netscape y hace su aparición cuando el candado de la barra de herramientas se

cierra y también si la dirección de internet cambia de http a https, otro ejemplo es **PGP** que es un protocolo libre ampliamente usado de intercambio de correo electrónico seguro, uno mas es el conocido y muy publicitado **SET** que es un protocolo que permite dar seguridad en las transacciones por internet usando tarjeta de crédito, **IPsec** que proporciona seguridad en la conexión de internet a un nivel mas bajo.
Estos y cualquier protocolo de seguridad procura resolver algunos de los problemas de la seguridad como la integridad, la confidencialidad, la autenticación y el no rechazo, mediante sus diferentes características
Las características de los protocolos se derivan de las múltiples posibilidades con que se puede romper un sistema, es decir, robar información, cambiar información, leer información no autorizada, y todo lo que se considere no autorizado por los usuarios de una comunicación por red.

Enseguida vemos un escenario donde puede ocurrir algo de esto:

Por ejemplo sobre la seguridad por Internet se deben de considerar las siguientes tres partes: seguridad en el browser (Netscape o Explorer), la seguridad en el Web server (el servidor al cual nos conectamos) y la seguridad de la conexión.

Un ejemplo de protocolo es **SET**, objetivo efectuar transacciones seguras con tarjeta de crédito, usa certificados digitales, criptografia de clave pública y criptografía clave privada.

SSL Es el protocolo de comunicación segura mas conocido y usado actualmente, **SSL** [81][82] actúa en la capa de comunicación y es como un túnel que protege a toda la información enviada y recibida.

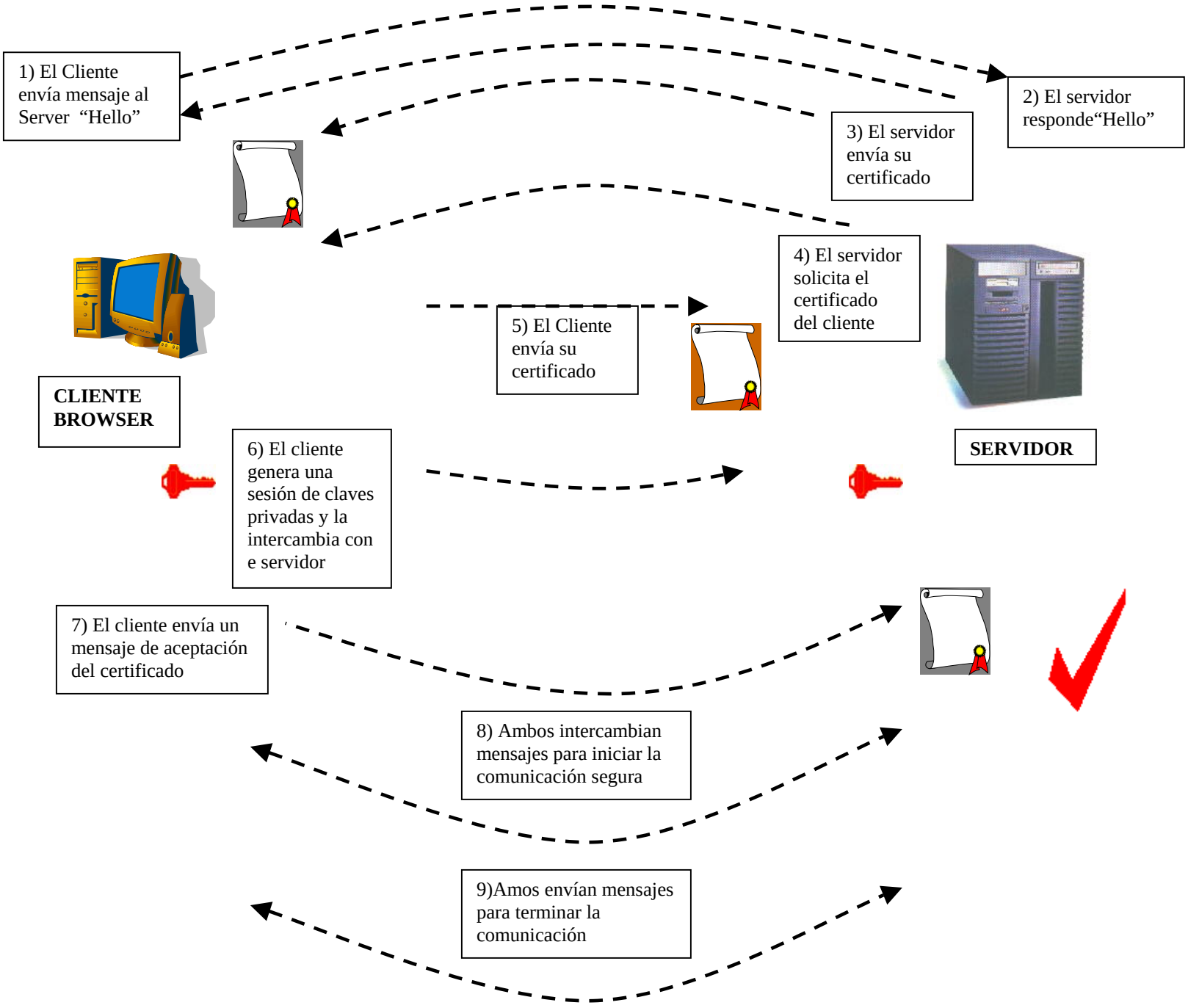
Con **SSL** se pueden usar diferentes algoritmos para las diferentes aplicaciones, por ejemplo usa **DES**, **TDES**, **RC2**, **RC4**, **MD5**, **SHA-1**, **DH** y **RSA**, cuando una comunicación esta bajo **SSL** la información que es cifrada es:

El URL del documento requerido
El contenido del documento requerido
El contenido de cualquier forma requerida
Los “cookies” enviados del browser al server
Los “cookies” enviados del server al browser
El contenido de las cabeceras de los http

El procedimiento que se lleva acabo para establecer una comunicación segura con **SSL** es el siguiente:

1) El cliente (browser) envía un mensaje de saludo al Server “ClientHello”
2) El server responde con un mensaje “ServerHello”
3) El server envía su certificado
4) El server solicita el certificado del cliente
5) El cliente envía su certificado: si es válido continua la comunicación si no para o sigue la comunicación sin certificado del cliente
6) El cliente envía un mensaje “ClientKeyExchange” solicitando un intercambio de claves simétricas si es el caso
7) El cliente envía un mensaje “CertificateVerify” si se ha verificado el certificado del server, en caso de que el cliente este en estado de autenticado
8) Ambos cliente y server envían un mensaje “ChangeCipherSpec” que significa el comienzo de la comunicación segura.
9) Al término de la comunicación ambos envían el mensaje “finished” con lo que termina la comunicación segura, este mensaje consiste en un intercambio del hash de toda la conversación, de manera que ambos están seguros que los mensajes fueron recibidos intactos

La versión más actual de **SSL** es la v3, existen otro protocolo parecido a **SSL** solo que es desarrollado por **IETF** que se denomina **TLS** (Transport Layer Security Protocol) y difiere en que usa un conjunto un poco mas amplio de algoritmos criptográficos. Por otra parte existe también **SSL** plus, un protocolo que extiende las capacidades de **SSL** y tiene por mayor característica que es interoperable con **RSA**, **DSA/DH** y **CE** (**C**riptografía **E**líptica).



El protocolo SSL

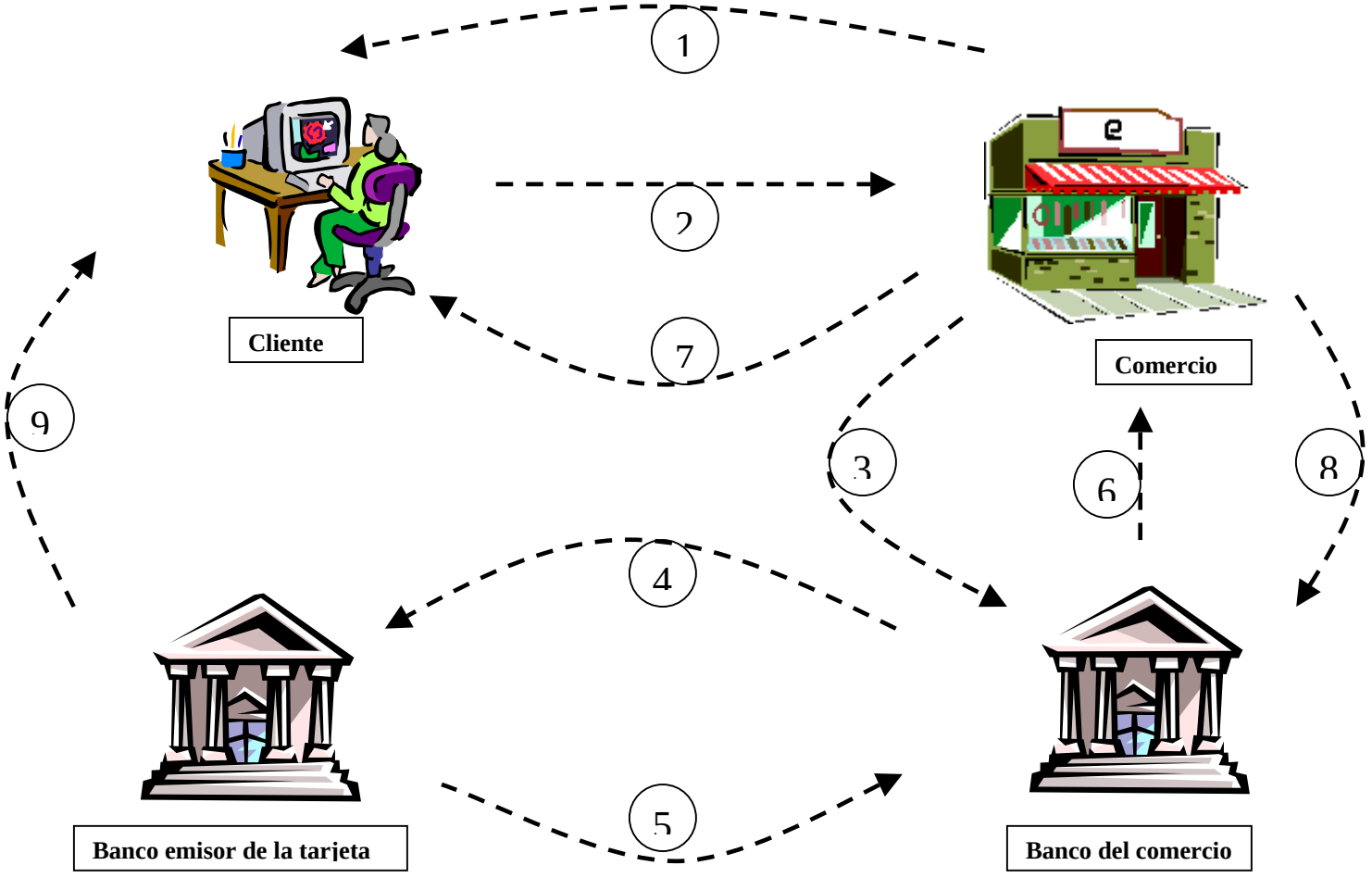
SET este protocolo esta especialmente diseñado para asegurar las transacciones por internet que se pagan con tarjeta de crédito. Esto es debido a que una gran cantidad de transacciones de compra por internet son efectuadas con tarjeta de crédito, por otro lado SSL deja descubierto alguna información sensible cuando se usa para lo mismo. La principal característica de **SET** [77][79][80][83], es que cubre estos huecos en la seguridad que deja **SSL**.

Por ejemplo con **SSL** solo protege el número de tarjeta cuando se envía del cliente al comerciante, sin embargo no hace nada para la validación del número de tarjeta, para chequear sí el cliente esta autorizado a usar ese número de tarjeta, para ver la autorización de la transacción del banco del comerciante etc., Además que el comerciante puede fácilmente guardar el número de tarjeta del cliente. En fin todas estas debilidades son cubiertas por **SET**, éste permite dar seguridad tanto al cliente, al comerciante como al banco emisor de la tarjeta y al banco del comerciante.

El proceso de **SET** es mas o menos el siguiente:

1) **El cliente inicializa la compra**: consiste en que el cliente usa el browser para seleccionar los productos a comprar y llena la forma de orden correspondiente. **SET** comienza cuando el cliente hace clic en "pagar" y se envía un mensaje de iniciar **SET**.
2) **El cliente usando SET envía la orden y la información de pago al comerciante**: el software **SET** del cliente crea dos mensajes uno conteniendo la información de la orden de compra, el total de la compra y el número de orden. El segundo mensaje contiene la información de pago, es decir, el número de la tarjeta de crédito del cliente y la información del banco emisor de la tarjeta. El primer mensaje es cifrado usando un sistema simétrico y es empaquetada en un sobre digital que se cifra usando la clave pública del comerciante. El segundo mensaje también es cifrado pero usando la clave pública del banco (esto previene que el comerciante tenga acceso a los números de tarjetas de los clientes). Finalmente el cliente firma ambos mensajes.
3) **El comerciante pasa la información de pago al banco**: el software **SET** del comerciante genera un requerimiento de autorización, éste es comprimido (con un hash) y firmado por el comerciante para probar su identidad al banco del comerciante, además de ser cifrado con un sistema simétrico y guardado en un sobre digital que es cifrado con la clave pública del banco.
4) **El banco verifica la validez del requerimiento**: el banco descifra el sobre digital y verifica la identidad del comerciante, en el caso de aceptarla descifra la información de pago del cliente y verifica su identidad. En tal caso genera una requerimiento de autorización lo firma y envía al banco que genero la tarjeta del cliente.
5) **El emisor de la tarjeta autoriza la transacción**: el banco del cliente (emisor de la tarjeta) confirma la identidad del cliente, descifra la información recibida y verifica la cuenta del cliente en caso de que no haya problemas, aprueba el requerimiento de autorización, lo firma y lo regresa al banco del comerciante.

6) **El banco del comerciante autoriza la transacción**: una ves recibida la autorización del banco emisor, el banco del comerciante autoriza la transacción la firma y la envía al servidor del comerciante.
7) **El servidor del comerciante complementa la transacción**: el servidor del comerciante da a conocer que la transacción que la tarjeta fue aprobada y muestra al cliente la conformidad de pago, y procesa la orden que pide el cliente terminado la compra cuando se le son enviados los bienes que compró el cliente.
8) **El comerciante captura la transacción**: en la fase final de SET el comerciante envía un mensaje de "captura" a su banco, esto confirma la compra y genera el cargo a la cuenta del cliente, así como acreditar el monto a la cuenta del comerciante.
9) **El generador de la tarjeta envía el aviso de crédito al cliente**: el cargo de SET aparece en estado de cuenta del cliente que se le envía mensualmente.



SET requiere un certificado digital en cada paso de autenticación y usa dos pares de claves, una para el cifrado del sobre digital y otra para la firma, (**SSL** solo usa un par de claves), actualmente **SET** usa la función hash **SHA-1**, **DES** y **RSA** de 1024 bits, estos parámetros fueron tomados para ser compatible con los certificados existentes, aunque el piloto de **SET** usó el sistema asimétrico de cifrado con curvas elípticas y se piensa que soporte también curvas elípticas en la próxima versión de **SET**.

8 Bibliografía

SEGURIDAD en COMPUTO

1) **E. Amoroso,** Fundamentals of computer Security Technology, Prentice Hall Inc., 1994
2) **E. Felten, D. Balfans, D. Dean, D. Wallach,** Web spoofing: An Internet congame. Technical Report 560-96, Dep. of Computer Science Princeton University 1996
3) **R. Focardi, R. Gorrieri,** A classification of security properties, Journal of Computer Security, 3 (1) 1995
4) **G.T. Gangemi, D. Russell**, Computer Security Basic, O'Reilly 1991
5) **S. Garfinkel, G. Spafford**, Web Security and Commerce, O'Reilly 1997
6) **D. Icove, K. Seger, W. VonStorch**, Computer Crime, O'reilly 1995
7) **P.E. Neumann,** Computer Related Risks, Addison Wesley Reading MA, 1995
8) **B. Preneel V. Rijmen (eds),** State of the Art in Applied Cryptography, LNCS 1528, 1998
9) **L.D. Stein,** Web Security, Eddison Wesley 1997
10) **W. Stallings,** Mecklermedia's official Internet world Internet security handbook, IDG Books, San Mateo, CA USA 1995

CRIPTOGRAFIA Y TEORIA DE NUMEROS

11) **C.M. Adams**, **S.E. Tavares**, The structured design of cryptographically good S-boxes, Jornal of cryptology V. 3 n 1, pp 27-42, 1990 (CAST)
12) **E. Biham, A. Shamir**, Differential cryptanalysis of the Data Encryption Standard Springer Verlag 1993
13) **I. Blake, X. Gao, A. Menezes, R. Mullin, S. Vanstone, T. Yaghoobian**, Applications of Finite Fields, Kluwer Academic Publishers 1992
14) **I. Blake, G. Seroussi, N. Smart**, Elliptic Curves in Cryptography LMS 265, Cambridge University Press 1999
15) **D.M. Bressoud**, Factorization and primality testing, UTM Springer-Verlag 1989
16) **L. Brown, M. Kwan, J. Pieprzyk, J. Seberry**, Improving resistance to differential cryptanalysis and the redesign of LOKI, Advances in Cryptology ASIACRYPT'91 LNCS 739, pp 36-50, 1993
17) **S.C. Coutinho**, The Mathematics of Ciphers, A.K. Peters 1998
18) **C. Charnes, L. O'Connor, J. Pieprzyk, R. Safavi-Naini, Y. Zheng**, Coments on Soviet encryption algorithm, Advances in Cryptology Eurocrypt'94, LNCS 950, pp 433-438, 1995 (GOST)
19) **C.A. Deavours, L. Kruh**, Machine Cryptography and Modern Cryptanalysis, Artech House Inc. 1985
20) **W. Diffie, M.E. Hellman,** New Directions in Cryptography, Transactions on Information Theory Vol IT22 No 6, pp 644-654 1976
21) **W. Friedman**, Cryptology, Encyclopedia Brittanica, 6, pp 844-851 1967
22) **D. Kahn**, The Codebreakers, the Story of Secret Writing, Macmillan Publishing Co. NY 1967
23) **N. Koblitz**, A course in Number Theory and Cryptography, Springer Verlag 1994
24) **N. Koblitz**, Algebraic Aspects of Cryptography, Springer Verlag 1998
25) **X. Lai, J.R. Massey**, A proposal for a new block encryption standard, Advances in Cryptology EUROCRYPT'90, LNCS 473, pp 389-404, 1991
26) **A. K. Lenstra**, **E. R. Verheul**, Selecting Cryptographic Key Sizes, 1999
27) **R. Lidl, H. Niederreiter**, Encyclopedia of Mathematics and Its Applications Vol. 20, Addison Wesley 1983
28) **M. Matsui**, Linear cryptanalysis method for DES cipher, Advances in Cryptology EUROCRYPT'93, LNCS 765, pp 386-397, 1994
29) **A.J. Menezes, P.C. van Oorschot, S.A. Vanstone**, Handbook of Applied Cryptography, CRC Press 1996
30) **A.J. Menezes, M. Qu, S. Vanstone**, Some New key agreement protocols providing implicit authentication, SAC'95, pp 18-19, 1995
31) **A.J. Menezes**, Elliptic Curve Key cryptosystems, Kluwer Academic Publishers 1993
32) **A.M. Odlyzko**, Public Key Cryptography, AT&T Bell Laboratories, Murray Hill, New Jersey 0797, 1993
33) **P. Rogaway**, The Security of DESX, CryptoBytes Vol 2, No 2, pp 8-10, 1996
34) **K.H. Rosen**, Elementary Number Theory and Its Applications, Addison Wesley 1988
35) **M. Rosing**, Implementing Elliptic Curve Cryptography, Manning Publications Co. 1998

36) **R.L. Rivest, A. Shamir, L. Adleman**, A Method for Obtaining Digital Signature and Public-Key Cryptosystems, Communication of the ACM Vol 21 No 2 pp 120-126, 1978
37) **R.L. Rivest**, The RC5 encryption algorithm, Fast software Encryption LNCS 1008, pp 86-96, 1995
38) **B. Schneier,** Applied Cryptography, John Wiley & Sons, Inc. 1996
39) **B. Schneier**, Description of a new variable-length key, 64-bit block cipher (Blowfish) FSE, LNCS 809, pp 191-204, 1994
40) **A. Shimizu, S. Miyaguchi**, Fast data encipherment algorithm FEAL, Advances in Cryptology Eurocrypt'87¸ LNCS 304, pp 267-278, 1988
41) **G.J. Simmons**, Cryptology, The new Encyclopaedia Britannica, Macropaedia Vol 16, pp 913-924B
42) **C.P. Schnorr**, Efficient identification and signatures for smart cards, Advances in cryptology CRYPTO'89, LNCS 435, pp 239-252, 1990
43) **D.R. Stinson,** Cryptography Theory and Practice, CRC Press Inc. 1995
44) **P. van Oorschot, M. Wiener**, A Known-plaintext attack on two-kwy triple encryption, Advances in Cryptology EUROCRYPT '90, LNCS 473, pp 318-325, 1991
45) **M.J. Wiener**, Cryptanalysis of short RSA secret exponents, IEEE Transactions on Information theory, 32, pp 553-558, 1990
46) **H.C. Williams**, A Modification of the RSA public-key encryption procedure, IEEE Transaction on Information Theory, No 26, pp 726-729 1980
47) **ANSI X3.92** "American National Standard- Data Encryption Algorithm" American National Standards Institute 1981
48) **ANSI X3.106** "American National Standard- Data Encryption Algorithm- Modes of Operation", American National Standards Institute 1983
49) **ANSI X9.31** "American National Standard for Finalcial Services- Public key cryptography using RSA for financial services industry" The RSA signature algorithm 1995 (part 1), hash algorithm for RSA (part 2) 1995
50) **ANSI X9.42** "Public key cryptography for financial services industry: Management of symmetric algorithm keys using Diffie-Hellman" 1995
51) **ANSI X9.57** "Public key cryptography for financial services industry" Certificate management" 1995
52) **FIPS 46-2** (1993) "Data Encryption Standard"
53) **FIPS 46-3** (1999) "TDES"
54) **FIPS 81** "DES modes of operation" 1980
55) **FIPS 186** "Digital signature standards" 1994
56) **ISO 8372** "Information processing – Modes of operation for a 64-bit block cipher algorithm", 1997, 1992
57) **ISO 9731**-1,2 "Banking – Approved algorithms for message authentication" 1987,1992
58) **ISO 11166**-1,2 "Banking – Key management by means of asymmetric algorithms 1994, 1995
59) **ISO 11568**-1,2,3,4,5,6 "Banking – Key management" 1994, 1996
60) **ISO 10118**-1,2,3,4 "Information technology- security techniques- hash functions" 1994, 1996
61) **ISO/IEC 14888**-1,2,3 "Information technology – security techniques- digital signature with appendix" 1996

62) **RFC 1321** "The MD5 message digest algorithm", Internet Request for Comments 1992
63) **RFC 1750** "Randomness requirements for security", Internet Request for Comments 1994
64) **RFC 2104** "HMAC: keyed-Hashing for Message Authentication", Internet Request for Comments 1997

65) http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html
66) http://www.cs.ucdavis.edu/~rogaway/papers/
67) http://theory.lcs.mit.edu/~rivest/
68) http://www.rsasecurity.com
69) http://www.certicom.com
70) http://www.counterpane.com
71) http://www.cacr.math.uwaterloo.ca/
72) http://www.cryptography.com/
73) http://www.zurich.ibm.com/Technology/Security/
74) http://csrc.nist.gov/encryption/aes/aes_home.htm
75) http://grouper.ieee.org/groups/1363/index.html
76) http://www.cacr.math.uwaterloo.ca/hac/

PROTOCOLOS (SSL, SET)

77) **G. N. Drew,** Using Set for Secure Electronic Commerce**,** , Prentice Hall, NJ 1999
78) **D.C. Lynch, L. Lundquist**, Digital Money, John Wiley & Sons 1996
79) **L. Loeb**, Secure Electronic Transactions, Introduction and Technical Reference, Artech House, 1998
80) **M.S. Merkow**, **J. Breithaupt**, Building SET Applications for Secure Transactions, John Wiley &Sons 1998
81) http://directory.netscape.com/Computers/Security/Internet/SSL-TLS
82) http://www.ietf.org/html.charters/tls-charter.html
83) http://www.setco.org/

INFRAESTRUCTURAS DE CLAVES PUBLICAS Y CERTIFICADOS DIGITALES

84) **W. Ford, M. S. Baum**, Secure Electronic Commerce: Building the Infrastructure for Digital Signature and Encryption, Prentice-Hall, Englewood Cliffs, NJ 1997
85) **J. Feghhi, J. Feghhi, P. Williams**, Digital Certificates Applied Internet Security, Addison Wesley, 1999
86) **L.M. Kohnfelder**, Toward a practical public-key cryptosystem, B.Sc. thesis, MIT Department of Electrical Engineering, 1978
87) http://www-08.nist.gov/pki/program/welcome.html

88) http://www.ietf.org/html.charters/pkix-charter.html
89) http://theory.lcs.mit.edu/~rivest/sdsi10.html
90) http://www.certco.com/
91) http://www.verisign.com/
92) http://www.entrust.com/
93) http://www.xcert.com/

CRIPTOGRAFIA VISUAL

94) **M.Naor** and **B.Pinkas**, Visual authentication and identification, in "Advances in Cryptology -- CRYPTO '97", B. Kaliski, Jr., ed., Lecture Notes in Computer Science 1294 (1997), 322-336
95) **M.Nao**r and **A.Shamir**, Visual cryptography, in "Advances in Cryptology -- EUROCRYPT '94", A. De Santis, ed., Lecture Notes in Computer Science 950 (1995), 1-12.
96) **M. Naor** and **A. Shamir**, Visual cryptography II: improving the constrast via the cover base, in "Security Protocols", M. Lomas, ed., Lecture Notes in Computer Science 1189 (1997), 197-202.
97) **D. R. Stinson**, Visual cryptography and threshold schemes, Dr. Dobb's Journal,,(1998), 36-43
98) http://www.cacr.math.uwaterloo.ca/~dstinson/visual.html

COMPARTICION DE SECRETOS

99) **A. Shamir**, How to share a secret, Communications of the ACM V. 22 1979, 612-613
100) http://www.cacr.math.uwaterloo.ca/~dstinson/ssbib.html