ANEXO 1 TÉRMINOS RELACIONADOS CON LA CRIPTOGRAFÍA

- Criptología

Estudio de la criptografía y el criptoanálisis. El criptoanálisis, es la ciencia que se encarga de descifrar los mensajes, mientras que la criptografía busca métodos más seguros de cifrado, y se puede clasificar en: Criptografía Clásica (basada en algoritmos sencillos y claves muy largas con cifrados rudimentarios basados en sustitución y transposición) y Criptografía moderna (que consiste básicamente en cifrados basados en algoritmos parametrizados en base a claves, aunque usa también transposición y sustitución). El cifrado moderno se divide actualmente en cifrado de clave privada y cifrado de clave pública. [11]

- Criptografía

La criptografía (del griego KRYPTOS (oculto) y GRAPHE (escrito)) es una ciencia que busca utilizar los conocimientos matemáticos (matemática realmente compleja) con el fin de crear algoritmos que permitan codificar (encriptar) la información antes de ser enviada a transitar por una red, y así dar un cierto nivel de seguridad en las transacciones electrónicas. Estos algoritmos pueden ser fuertes o débiles, esta característica de fortaleza se traduce en que tan difícil es descifrar el algoritmo (en cuanto a esto la última palabra la tienen los criptoanalistas quienes son científicos dedicados a la tarea de atacar y atacar estos algoritmos con el fin de descifrarlos). Hoy en día la criptografía es utilizada como medio para construir soluciones en seguridad electrónica que cumplan los siguientes aspectos:

- **IDENTIFICACIÓN:** Poder reconocer a un individuo en particular de otros.
- **AUTENTICACIÓN:** Poder comprobar y verificar cierta información.
- **AUTORIZACIÓN:** Establecer lo que puede hacer el usuario después de que un sistema lo autentica.
- **INTEGRIDAD:** Poder asegurar al usuario que la información no va a cambiar durante su tránsito por un sistema.
- **CONFIDENCIALIDAD:** Garantizar que la información que transita por un sistema va a permanecer en secreto.
- **ACEPTACIÓN:** Lograr que un usuario de un sistema no pueda negar algo que hizo sobre él.

Y para dar solución a lo anterior se debe indagar en lo que son la criptografía simétrica y asimétrica [2].

- Criptografía Simétrica

Este tipo de criptografía es tan antigua como el hombre mismo, se muestran antecedentes claros de ella incluso en el antiguo Egipto. Este tipo de criptografía busca el crear una clave, la cual se usa para cifrar la información; así como una única llave puede abrir y cerrar una puerta, solo con esa clave se podrá descifrar la información; una clave simétrica no es otra cosa que un número aleatorio que es generado por alguna fuente al azar y el cual se usa para cifrar y descifrar la información, la longitud de la clave la da el algoritmo, así por ejemplo si se trata de un algoritmo de cifrado simétrico de 128 bits, la clave simétrica tendrá también 128 bits de longitud. La característica más importante en el cifrado simétrico es el hecho de que el cifrado es muy rápido y la información se mantiene compacta, ya que el tamaño de la información cifrada es prácticamente el mismo que el de la información sin cifrar, pero no todo es bueno con la criptografía simétrica ya que se debe comprender que en una

comunicación de este tipo tanto el emisor como el receptor deben manejar la misma clave, el primero para cifrar la información y el segundo para descifrarla, pero si la clave cae en otras manos, la información podría ser descifrada por terceros; el otro problema tiene que ver con respecto al número de claves que deben usarse en un sistema de varios usuarios, ya que la clave simétrica debe ser única para cada comunicación; si por ejemplo tenemos un sistema de cinco usuarios se necesitarían cuatro claves por cada usuario para que él se pueda comunicar con los demás, resultando en un total de 20 claves por usuario, si fueran 30 usuarios necesitaría 870 claves, es decir por cada cantidad de usuarios que el sistema deba manejar se necesitaría manejar una cantidad de claves muy cercana al cuadrado del número de usuarios. Entre los algoritmos de cifrado simétrico se encuentran el Rivest Cipher en sus diferentes versiones (RC2, RC4, RC5), el algoritmo de cifrado estándar internacional (IDEA), el estándar de cifrado de datos (DES), entre otros [2][3].

Lo anterior se ilustra en la siguiente gráfica:

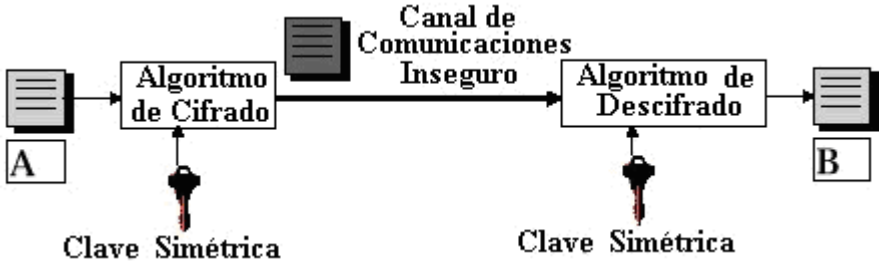

Figura 1: Modelo Básico de Criptografía Simétrica

- Criptografía Asimétrica

Con respecto a este tipo de criptografía existe un problema que de entrada es difícil de manejar, el cual es el del tamaño de las claves de cifrado; mientras que con la criptografía simétrica el valor típico máximo de longitud de las claves es de 128 bits, aquí se debe manejar una longitud de alrededor de 1620 bits para garantizar un nivel cercano de seguridad al simétrico. En la criptografía simétrica para generar la clave simétrica simplemente se genera un numero aleatorio, pero en la criptografía asimétrica el proceso es mucho más complejo, ya que aquí se usa una clave para cifrar la información y otra diferente para descifrarla, estas dos claves siempre son creadas al mismo tiempo y pese a ser independientes, matemáticamente están estrechamente relacionadas. Cuando se generan estas dos claves, una se rotula como clave privada y es la que solo va a conocer el usuario que generó las dos claves, y la otra se llamará clave pública que van a conocer todos los usuarios del sistema; lo interesante de esto es que la información que fue cifrada con la clave privada de un usuario solo podrá ser descifrada usando su clave pública y viceversa, es decir en este sistema de par de claves una clave no puede descifrar lo que cifró. Como se puede observar este sistema tiene grandes ventajas pero también grandes desventajas, la criptografía asimétrica soluciona el problema de la posible interceptación de la clave ya que en ésta solo se comparte una clave pública, en consecuencia en un sistema basado en este tipo de criptografía en lugar de necesitarse compartir un numero de claves de casi el total del cuadrado de usuarios, se necesitará compartir un total de claves públicas equivalente al mismo número de usuarios del sistema; otro gran beneficio es su propia naturaleza asimétrica, ya que el hecho de que todos los usuarios tengan una clave privada única, garantiza que cada usuario pueda hacer operaciones que nadie más en el sistema pueda llevar a cabo, y esto forma la base de lo que son las firmas digitales y soluciona el problema de la aceptación. Pero

obviamente no todo puede ser bueno, en contrariedad a su contraparte simétrica, los algoritmos asimétricos son mucho más lentos, lo cual podría ser un gran problema a la hora de tratar con volúmenes grandes de información, y esto no es todo, estos algoritmos también presentan el problema de que al cifrar la información la agrandan, es decir la información cifrada siempre es de mayor tamaño que la original. Entre los algoritmos de cifrado asimétrico se encuentran el de Rivest, Shamir y Adleman (RSA), el de Diffie y Hellman, entre otros [2][3].

Lo anterior se ilustra a continuación:

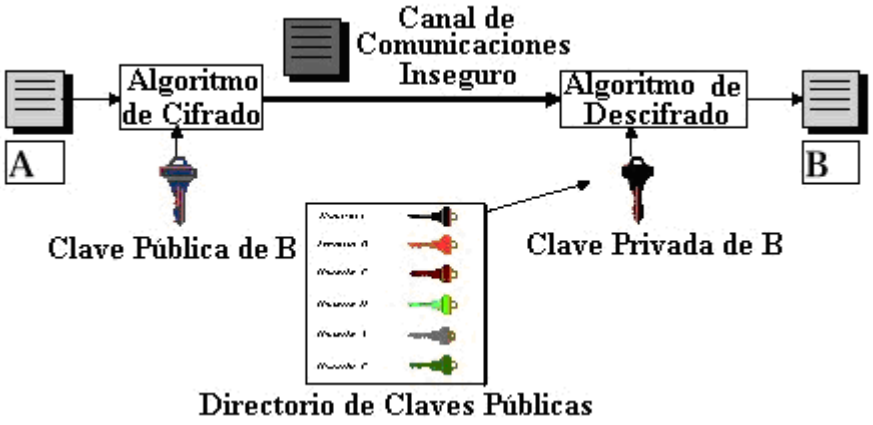

Figura 2: Modelo Básico de Criptografía Asimétrica

- Hashes

Un hash es sencillamente un algoritmo que a partir de un gran bloque de datos, saca una reseña (digest) mucho más pequeña que los datos originales, la manera en que hace esto es un proceso matemático muy complejo. Un algoritmo hash cumple con lo siguiente: no se puede usar el hash para generar el bloque de datos inicial a partir de la reseña, la reseña no dice nada del bloque de datos inicial, cualquier cambio por mínimo que sea en el bloque de datos inicial producirá una reseña diferente. Entre los algoritmos hash se encuentran el Message Digest en sus diferentes versiones (MD2, MD5) y el SHA [2].

- Firmas Digitales [2].

Firmar digitalmente un archivo electrónico no es otra cosa que generar su reseña con un hash y a continuación cifrarla usando la clave privada de quien lo va a enviar.

La validación de identificación y autenticidad de muchos documentos legales, financieros y de otros tipos se determina por la presencia o ausencia de una firma manuscrita autorizada. Para que los sistemas computerizados de mensajes reemplacen el transporte físico de papel y tinta, debe encontrarse una solución a estos problemas.

El problema de inventar un reemplazo para las firmas manuscritas es difícil. Básicamente, lo que se requiere es un sistema mediante el cual una parte pueda enviar un mensaje “firmado” a otra parte de modo que:

El receptor pueda verificar la identidad proclamada del transmisor.

El transmisor no pueda repudiar después el contenido del mensaje.

El receptor no haya podido confeccionar el mensaje él mismo.

El primer requisito es necesario, por ejemplo, en los sistemas financieros. Cuando la computadora de un cliente ordena a la computadora de un banco que compre una tonelada de oro, la computadora del banco necesita asegurarse de que la computadora que da la orden realmente pertenece a la compañía a la que se le aplicará el débito.

El segundo requisito es necesario para proteger al banco contra fraudes. Si el banco compra una tonelada de oro, e inmediatamente después cae el precio del oro. Un cliente deshonesto podría demandar al banco, alegando que nunca emitió una orden para comprar el oro. Cuando el banco presenta el mensaje ante el juez, el cliente niega haberlo enviado.

El tercer requisito es necesario para proteger al cliente en el caso de que el precio del oro suba y que el banco trate de falsificar un mensaje firmado en el que el cliente solicitó un lingote de oro en lugar de una tonelada.

Al igual que la criptografía, las firmas digitales se dividen en dos grandes grupos, firmas de clave secreta y firmas de clave pública.

Firmas de clave secreta [2].

Un enfoque de las firmas digitales sería tener una autoridad central que sepa todo y en quien todos confíen, por ejemplo X. Cada usuario escoge entonces una clave secreta y la lleva personalmente a las oficinas de X. Por tanto, sólo A y X conocen la clave secreta de A, KA, etc.

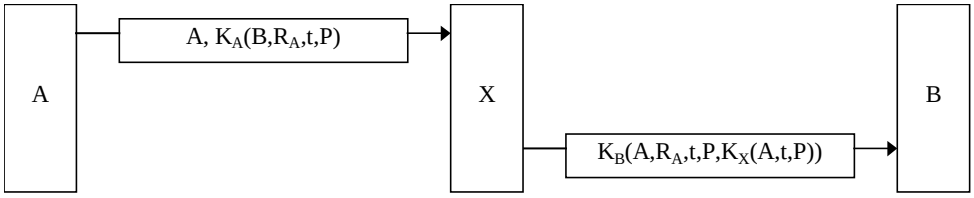

Figura 3: Firma con Clave Secreta Compartida con Centro de Distribución de Claves

Cuando A quiere enviar un mensaje de texto normal firmado, P, a B; genera KA(B,RA,t,P) y lo envía como se muestra en la figura anterior. X ve que el mensaje es de A, lo descifra y envía un mensaje a B como se muestra. El mensaje a B contiene el texto normal del mensaje de A y también el mensaje firmado KX(A,t,P), donde t es una marca de tiempo. Ahora B atiende la solicitud de A.

Si ahora A niega el envío del mensaje, cuando el caso llega al juez y A niegue haber enviado a B el mensaje, B indica que el mensaje vino de A y no de un tercero C pues X no hubiera aceptado un mensaje de A a menos que estuviese cifrado con KA, por lo que no hay posibilidad de que C enviará a X un mensaje falso en nombre de A. B además presenta la prueba KX(A,t,P). Entonces el juez pide a X (en quien todo el mundo confía) que descifre la prueba. Cuando X testifica que B dice la verdad el caso queda resuelto.

Un problema potencial del protocolo de firma anterior es que C repita cualquiera de los dos mensajes. Para minimizar este problema, se usan en todos los intercambios marcas de tiempo. Es más, B puede revisar todos los mensajes recientes para ver si

se usó RA en cualquiera de ellos. De ser así, el mensaje se descarta como repetición. Nótese que B rechazará los mensajes muy viejos con base en la marca de tiempo. Para protegerse contra ataques de repetición instantánea, B simplemente examina el RA de cada mensaje de entrada para ver si un mensaje igual se recibió de A durante el tiempo de validez de la marca temporal. Si no, B puede suponer con seguridad que ésta es una solicitud nueva.

Firmas de clave pública [2].

Un problema estructural del uso de la criptografía de clave secreta para las firmas digitales es que todos tienen que confiar en X. Es más, X lee todos los mensajes firmados. Los candidatos más lógicos para operar el servidor X son el gobierno, los bancos y los abogados. Estas organizaciones no tiene porqué inspirar confianza completa a todos los ciudadanos. Por tanto, sería bueno si la firma de documentos no requiriese una autoridad confiable.

Afortunadamente, la criptografía de clave pública puede hacer una contribución importante aquí. Supongamos que los algoritmos públicos de cifrado y descifrado tienen la propiedad de que E(D(P))=P además de la propiedad normal de D(E(P))=P (el RSA tiene esta propiedad por lo que el supuesto es razonable). Suponiendo que éste es el caso, A puede enviar un mensaje de texto normal firmado y cifrado, P, a B transmitiendo EB(DA(P)). Firmado por DA(P) y cifrado por EB(), de forma que sólo B podrá leerlos. Nótese que A conoce su propia clave de descifrado (privada), DA, así como la clave pública de B, EB, por lo cual la construcción de este mensaje es algo que A puede hacer.

Cuando B recibe el mensaje, lo transforma usando su clave privada, como es normal, produciendo DA(P), como se muestra en la figura siguiente:

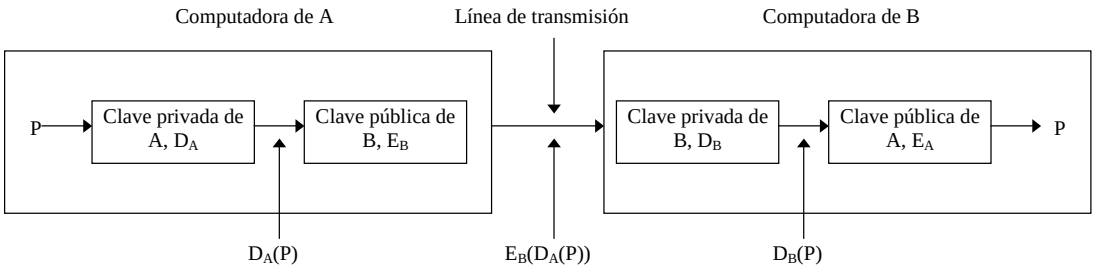

Figura 4: Firma Digital con Clave Pública

B almacena este texto en un lugar seguro y lo descifra usando EA para obtener el texto normal original.

En todo el proceso, en resumen lo que se ha realizado es EA(DB(EB(DA(P)))).

Para ver cómo funciona la propiedad de firma, supongamos que A niega haber enviado el mensaje P a B. Cuando el caso llega al juez, B puede presentar tanto P como DA(P). El juez puede comprobar fácilmente que B tiene un mensaje válido cifrado por DA con solo aplicarle EA. Puesto que B no conoce la clave privada de A, la única forma en que B pudo haber adquirido el mensaje cifrado con ella sería que A en efecto lo hubiera enviado.

Sin embargo existen dos problemas, por un lado B puede demostrar que un mensaje fue enviado por A siempre y cuando DA permanezca en secreto. Si A divulga su clave secreta, el argumento ya no se mantiene. Por otro lado, si A decide cambiar su clave,

algo legal y probablemente buena idea de forma periódica, cuando se aplique la actual EA a DA(P) no se obtiene P. En consecuencia, parece que sí que se requiere alguna autoridad para registrar todos los cambios de clave y sus fechas.

En principio cualquier algoritmo de clave pública puede usarse para firmas digitales. El estándar de facto de la industria es el algoritmo RSA y muchos productos de seguridad lo usan.

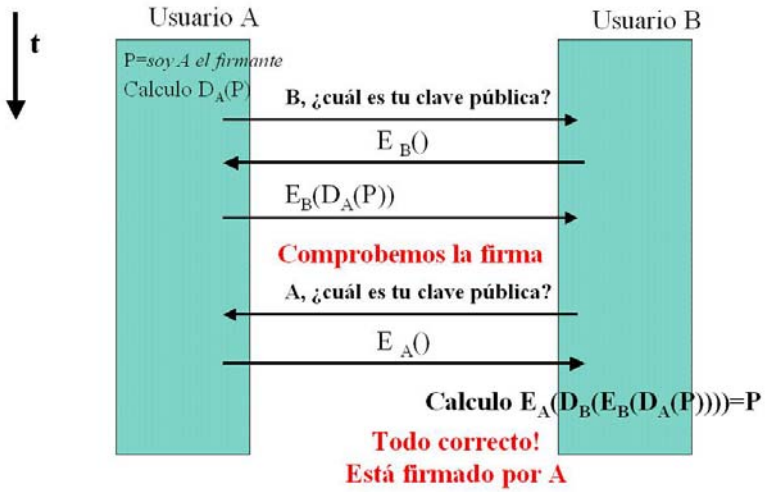

Figura 5: Protocolo de Firma Digital con Clave Pública

Con todo ello, el protocolo quedaría tal como se ve en la figura 20.

Para resumir lo anterior se presenta la gráfica:

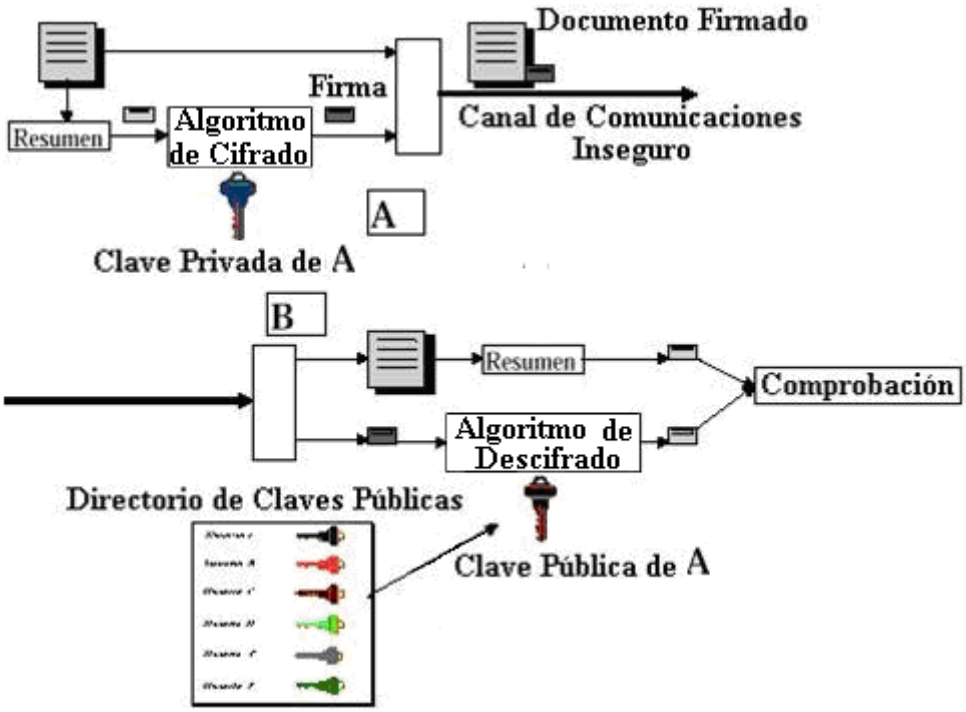

Figura 6: Modelo Básico de Firma Digital con Clave Pública

Aplicaciones seguras y uso de certificados [2].

Uno de los problemas que parecen en la distribución de claves públicas, es la propia distribución de dicha clave, que como hemos visto puede sufrir del ataque de alguien en medio. Una solución es la aplicación de métodos de interbloqueo, pero la solución más adoptada y estandarizada en la utilización de certificado.

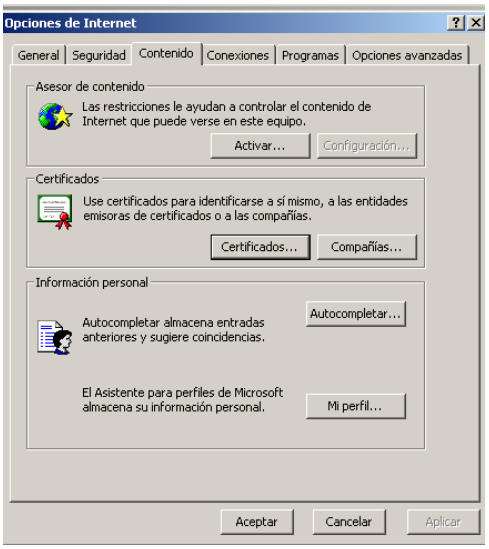

Figura 7: Configuración de Certificados en Herramientas: Opciones de Internet: Contenidos: Certificados en Internet Explorer 5.0

El certificado digital es un vínculo entre una clave pública y una identidad, que se consigue mediante una firma digital por una tercera parte o autoridad de certificación (CA: Certificaciont Authority o Autoridad de Certificación) comprobando la integridad. Se considera como un objeto firmado aquel que tiene: identidad del sujeto, clave, periodo de validez, identidad emisor, etc.; La Fábrica Nacional de Moneda y Timbre (FNMT), Verisign, Deutsche Telecom., Microsoft, Xcert, entre otras son diferentes autoridades de certificación.

A modo de ejemplo, en el caso de pedir un certificado para un ciudadano de algún país, su certificado sería un archivo, firmado con la clave privada de CA con la identidad, la clave pública del dicha identidad, atributos varios y compendio de dicha información:

DCA(identidad, clave, atributos, compendio{identidad, clave, atributos})

De esta forma, cualquiera que tenga el certificado raíz de la CA y pueda obtener la clave pública de la CA, podrá comprobar que dicha información es íntegra y válida, con lo cual permite autenticar a dicha identidad.

La mayoría de navegadores disponen ya en la distribución, en los propios CDs de instalación, un directorio lleno de certificados raiz, es decir, certificados de las propias autoridades de certificación firmados por ellas mismas.

La autoridad de certificación es reconocida y aceptada por todos, e imposible de suplantar. Por regla general, por seguridad no se trabaja directamente con la autoridad de certificación, si no con un intermediario o autoridad de registro.

La autoridad de registro atiende las peticiones de certificado y proporciona diferentes métodos de petición de certificados.

Para implementar el sistema de autenticación en base a certificados a través de Autoridad de Certificación (CA) han de considerarse los siguientes elementos:
Una política de certificación, incluyendo el ámbito de actuación (identificando los elementos a certificar, tanto personas como procesos, como servidores y/o clientes) así como la relación de la CA con otras CA, por regla general de forma jerárquica
Un certificado de la CA, de otra CA superior u homóloga que asegure quien dice ser y valida su clave pública.

Los certificados de los usuarios (p.ej X.509), así como el procedimiento de certificación y de revocación. La revocación permite consultar directamente qué certificados no son válidos.

Los protocolos de autenticación, gestión y obtención de certificados, así como de distribución de ellos son como veremos a continuación:
–o por bases de datos (p.ej directorio X.500) o comúnmente llamados servidores de directorios, que veremos a continuación
–o bien directamente del usuario en tiempo de conexión (WWW con SSL, Secure Socket Layer).

Por tanto para la puesta en marcha de una CA se requiere generar un par de claves (tanto privada como pública), proteger la clave privada y generar el certificado de la propia CA a través de otra CA.

El certificado raíz es un certificado emitido de la CA para sí misma con su clave pública, para comprobar certificados emitidos por ella. Se suele instalar previamente dicho certificado en el navegador para poder utilizar los certificados de dicha CA. Los navegadores llevan por defecto muchos de ellos.

Lista de certificados revocados (o CRL Certificate Revocation List) es una lista donde se recogen todos los certificados de la CA dados de baja por caducidad o por problemas varios como que se haya hecho pública la clave privada de un usuario, y por tanto cualquier firma emitida con posterioridad a la revocación no tiene validez. Este documento también es firmado por la propia CA [2].

Pero, esta infraestructura de autoridades en muchas ocasiones no se dispone, ni de Autoridades de Certificación ni de Registro. Con este planteamiento inicial una solución tomada estriba en la confianza de los propios usuarios entre ellos. Este tipo de redes se llaman redes de confianza.

- Certificados Digitales [2].

Un certificado digital es simplemente un documento electrónico que presenta quien es el propietario de la clave pública que hay en él y valga la redundancia certifica que esa clave pública pertenece solo a esa persona en particular, para validar este documento una autoridad de certificación de confianza lo firma digitalmente (esto es sacarle la reseña y cifrarla con su clave privada) y adjunta esta firma al mismo, por lo cual para verificar su validez lo único que se debe hacer es generar su reseña y compararla con la reseña que se obtiene al aplicar la llave pública de la autoridad de certificación a la firma digital que este tiene, si corresponden sabemos de la autenticidad del certificado y la clave pública que hay en él (establecimiento de una relación de confianza).

Certificado X.509 [2].

X.509 es el protocolo que se utiliza para certificar las claves públicas, con lo que los usuarios pueden intercambiar datos de manera segura. X.509 está basado en criptografía asimétrica y firma digital y se emplea para autentificar la información en redes externas, en redes internas, en el correo electrónico y en general en aplicaciones que requieran autenticación.

Para ello, el certificado se cifra con la clave privada de la CA y todos los usuarios poseen la clave pública del CA.

El protocolo X.509 fue creado por la UIT para servir al X.400 (correo electrónico de OSI) y su origen se encuentra en el servicio de directorio X.500. En X.509 se define un framework (una capa de abstracción) para suministrar servicios de autenticación a los usuarios del directorio X.500.

El objetivo de este protocolo es asegurar la integridad, la privacidad y el no repudio de los mensajes.

X.509 se utiliza en SSL en los navegadores (https), PEM en el correo electrónico seguro, S/MIME, SET (Secure Electronic Transaction) que define una arquitectura para E-Commerce

Los campos del X.509 escritos en ASN1 son:
Versión: La del protocolo X.509 (actualmente versión 3)
Número de serie: Es un número asignado por el CA y que identifica de manera única el certificado.
Algoritmo de la firma del certificado: Identifica el algoritmo utilizado para firmar el certificado.
Autoridad de certificación (CA): Es el nombre de la CA.
Fecha de inicio y final: tiempo de validez
Usuario: Es el nombre del usuario.
Clave pública: Es la clave del usuario.
Identificador único del CA: Es el número que identifica al CA. Es único en el mundo.
Identificador único del usuario: Es el número que identifica al usuario para todos sus certificados.
Extensiones: Si hay extensiones de la información
Firma de la CA: Firma todos los campos anteriores empleando, para ello, su clave privada.

- Resolución del problema de seguridad del secreto [2].

La resolución del problema del secreto en la red (y del secreto de los mensajes en cualquier sistema de comunicación), ha estado siempre unido al cifrado (codificación) de los mensajes. Hasta la llegada de las computadoras, la principal restricción del cifrado consistía en la capacidad del empleado encargado de la codificación para realizar las transformaciones necesarias. Otra restricción adicional consistía en la dificultad de cambiar rápidamente el método de cifrado, pues esto implicaba entrenar a una gran cantidad de empleados. Sin embargo, el peligro de que un empleado fuera capturado por el enemigo, etc., ha hecho indispensable la capacidad de cambiar el método de cifrado al instante. De estos requisitos se deriva el modelo de la figura siguiente:

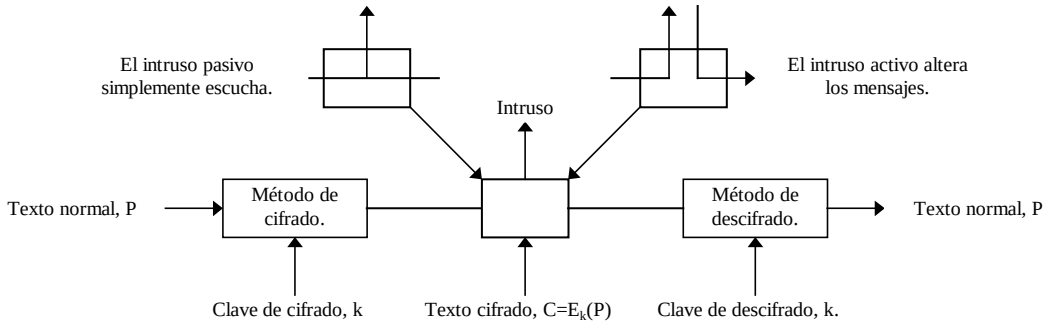

Figura 8: Modelo de Cifrado de Datos

Los mensajes a cifrar, conocidos como texto normal, se transforman mediante una función parametrizada por una clave. La salida del cifrado, conocida como texto cifrado, es transmitida después. Si un intruso escucha y copia el texto cifrado, a diferencia del destinatario original, no conoce la clave de cifrado y no puede descifrar fácilmente el texto cifrado.

Se denotará C=Ek(P) para indicar que el cifrado del texto normal P usando la clave K da el texto cifrado C. Del mismo modo P=Dk(C) representará el descifrado de C para obtener el texto normal nuevamente. Por tanto, $D_k(E_k(P))=P$. Esta notación sugiere que E y D son sólo funciones matemáticas de dos parámetros, de los cuales hemos escrito uno (la clave) como subíndice, en lugar de como argumento, para distinguirlo del mensaje [2].

Una regla fundamental de la criptografía es que se debe suponer que el criptoanalista conoce el método general de cifrado usado, esto es, el criptoanalista conoce E, pues la cantidad de esfuerzo necesario para inventar, probar e instalar un método nuevo cada vez que el viejo es conocido siempre hace impracticable mantenerlo en secreto. Aquí es donde entra la clave. La clave consiste en una cadena relativamente corta que selecciona uno de los muchos cifrados potenciales; en contraste con el método general, que tal vez se cambie cada cierto número de años, la clave puede cambiarse con la frecuencia que se requiera, por lo cual nuestro modelo es un método general estable y conocido públicamente pero parametrizado por una clave secreta y fácilmente cambiable. Un ejemplo de esto es una cerradura de combinación. Todo el mundo conoce como funciona, pero la clave es secreta. Una longitud de clave de tres dígitos significa que existen 1000 posibilidades, una longitud de clave de seis dígitos implica un millón de posibilidades [2].

- Principios criptográficos fundamentales [2].

El primer principio es que todos los mensajes cifrados deben contener redundancia, es decir, información no necesaria para entender el mensaje. Un ejemplo puede dejar claro la necesidad de esto. Considere una compañía de compras por red que tiene 60000 productos. Pensando en la eficiencia, los programadores han decidido que los mensajes de pedidos deben consistir en el nombre del cliente de 16 bytes seguido de un campo de datos de 4 bytes (2 para la cantidad y 2 para el número del producto). Los últimos 4 bytes deben cifrarse usando una clave muy grande conocida solo por el cliente y por la compañía.

Inicialmente esto puede parecer seguro, y lo es, porque los intrusos pasivos no pueden descifrar los mensajes. Sin embargo, también tiene un fallo que lo vuelve inútil. Supóngase que alguien consigue una lista de compradores de productos (los 16 bytes del número de cliente), entonces es fácil trabajar en un programa para generar pedidos ficticios usando nombres reales de los clientes. Dado que no tiene una lista de claves, pone números aleatorios en los últimos 4 bytes y envía cientos de pedidos. Al llegar estos mensajes, la computadora usa el nombre del cliente para localizar la clave y descifrar el mensaje. Casi todos los mensajes de 4 bytes descifrados son válidos, pues excepto que el pedido sea de cantidad 0 al descifrarlo o corresponda a un producto cuyo código no existe (cuya probabilidad solo es de un 8.5%), el pedido será atendido.

Este problema puede resolverse agregando redundancia a todos los mensajes. Por ejemplo, si se extienden los mensajes de pedido a 12 bytes, de los cuales los 8 primeros deben ser ceros, entonces este ataque ya no funciona, pues la probabilidad de generar un mensaje valido es prácticamente cero. Sin embargo la adición de redundancia simplifica a los criptoanalistas el descifrado de los 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 (Cyclic Redundant 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.

El segundo principio criptográfico es que deben tomarse algunas medidas para evitar que los intrusos activos reproduzcan mensajes viejos. Si no se toman tales medidas, alguien puede conectarse a la línea telefónica de la empresa de pedidos por red y simplemente continuar repitiendo mensajes válidos enviados previamente. Una de tales medidas es la inclusión en cada mensaje de una marca de tiempo válida durante, digamos, 5 minutos. El receptor puede entonces guardar los mensajes unos 5 minutos, para compararlos con los mensajes nuevos que lleguen y filtrar los duplicados. Los mensajes con mayor antigüedad que 5 minutos pueden descartarse, dado que todas las repeticiones enviadas más de 5 minutos después también se rechazarán como demasiado viejas.

PKI: Public Key Infraestructura [2].

Una PKI se define como la infraestructura de red formada por servidores y servicios que en base a claves públicas gestionan de forma segura todas las transacciones realizadas a través de la red. Actualmente está en fase de estandarización con un RFC. Las operaciones básicas realizadas en una PKI son la certificación (medio por el cual las claves se publican) y la validación, a través de revocación y autenticación. Las claves públicas de los servicios, usuarios, aplicaciones, clientes, etc.; pueden ubicarse en servicios de directorios, en autoridades de certificación, en tarjetas inteligentes, o de otras formas. Los fabricantes de PKI más representativos son RSA Labs, Verisign GTE Cyber Trust, Xcert, Netscape, entre otros.

La PKI como su nombre indica hace uso de los algoritmos de cifrado con clave pública y las aplicaciones (u operaciones) que se realizan con ellas son: para cifrar, por ejemplo en el caso que A quiere mandar P a B, para ello utiliza la clave pública de B, enviando EB(P) y B utilizando su clave privada realiza el paso inverso DB (EB(P)) para firmar y autenticar, por ejemplo en el caso que A quiere mandar P a B y firmarlo.

Entonces para ello A manda {P, DA(P)} utiliza su clave privada y B utilizando la clave pública de A, realiza la comprobación EA (DA(P)).

Una de las funciones importantes de la PKI es la validación de la información de los certificados. Para ello el usuario dispone de dos formas principales de validación, o preguntar por la validez a la CA (validación conocida como on line) y/o examinar el periodo de validez incluido en el certificado (off-line). El problema en esta última forma de validación, estriba en la complejidad de gestionar la revocación de certificados. La información del certificado es inválida si, la clave privada de la entidad se compromete, o si los datos de la entidad cambian.

Si la validación se realiza on-line, la revocación resulta sencilla. Si la validación se realiza off-line, la revocación se realiza mediante métodos como las listas de certificados revocados o CRLs (Certificate Revocated List), que es una lista de los certificados revocados, publicada y firmada periódicamente por la CA. El usuario chequea esta lista para comprobar la validez de los certificados.

Las distribución y gestión de las claves dentro de las PKI es lo que se llama repositorios (llaveros) de Certificados y son los elementos necesarios para almacenar los certificados de los usuarios. Para poder almacenarlos se suele utilizar servidores de directorios basados en X.500 y LDAP. Cabe destacar que los principales navegadores llevan soporte para servidores de directorio y también pueden gestionar las claves directamente con SSL (Secure Socket Layer) con total seguridad.

Las PKI pueden estar basadas en certificados X.509 y/o el soporte de repositorio de llaves (llaveros) de PGP, ya que éste incluye opción para clave pública, bien por confianza directa, es decir reside en los propios usuarios y confianza jerárquica, a través de una CA.

ANEXO 2 TÉCNICAS DE CIFRADO

Criptografía clásica [2].

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 transposición. Se tomará como texto normal aquel que se encuentra representado por letras minúsculas y como texto cifrado el que se encuentre representado por letras mayúsculas.

Cifrado por sustitución [2].

El cifrado por sustitución se basa en la sustitución de cada letra o grupo de letras por otra letra o grupo de letras para disfrazarla. Uno de los cifrados por sustitución más antiguos conocidos es el cifrado de Cesar, atribuido al emperador romano Julio Cesar. En este método la letra a se convierte en D, la b en E, la c en F, ... , y z se vuelve C. Así, el mensaje ataque se convierte en DWDTXH. Una generalización del cifrado de Cesar permite que el alfabeto de texto cifrado se desplaza k letras en lugar de siempre 3, con lo cual k se convierte en la clave de cifrado.

La siguiente mejora es hacer que cada uno de los símbolos del texto normal, por ejemplo las 26 letras del alfabeto inglés, tengan una correspondencia biunívoca con alguna otra letra. Por ejemplo:

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

Este sistema general se llama sustitución monoalfabética, siendo la clave la cadena de 26 letras correspondiente al alfabeto completo. A primera vista, esto podría parecer un sistema seguro, porque aunque el criptoanalista conoce el sistema general (sustitución letra por letra), no sabe cuál de las 26!=4x1026 claves posibles se está usando. Sin embargo, si se cuenta con una cantidad pequeña de texto cifrado, el cifrado puede descifrarse fácilmente. El ataque básico aprovecha las propiedades estadísticas de los lenguajes naturales. En inglés, la letra e es la más común, seguida de t, o, a, n, i, etc. Las combinaciones de letras más comunes o digramas son th, in, er, re y an. Las combinaciones de tres letras más comunes o trigramas son the, ing, and e ion.

Un criptoanalista que intenta descifrar una codificación monoalfabética comenzaría por contar la frecuencia relativa de todas las letras del texto cifrado. Entonces podría asignar tentativamente la más común a la letra e y la siguiente más común a la letra t. Vería entonces los trigramas para encontrar uno común de la forma tXe, lo que sugerirá que X es h. De la misma manera si el patrón thYt ocurre con frecuencia, Y probablemente representa a la letra a. Con esta información puede buscar trigramas frecuentes de la forma aZW, que con probabilidad es and, y acabar de forma similar descifrando el texto cifrado.

Otros métodos de cifrado por sustitución son:

Cifrado de Polybius [2].

Se logra introduciendo el alfabeto dentro de una tabla por filas, donde la tabla tiene un número de columnas determinado, este número de columnas es la clave conociendo el algoritmo. El texto normal se codifica en base a las coordenadas de las letras del

texto normal dentro de dicha tabla. La clave de este cifrado está en la disposición del alfabeto en la tabla, es decir el número de columnas de la tabla.

	1	2	3	4	5	6	7
1	A	B	C	D	E	F	g
2	H	I	J	K	L	M	N
3	Ñ	O	P	Q	R	S	t
4	U	V	W	X	Y	Z	+

Si P=HOLA, entonces el cifrado consiste en codificar (fila, columna) por tanto el cifrado es (2,1), (3,2), (2,5), (1,1)

Cifrado de Trithemius [2].

Es un método de sustitución progresivo, basado en el cifrado de Julio Cesar, donde el valor de k varía incrementalmente de forma conocida en los extremos. Ejemplo: clave k=+2 y texto normal P="Hola" =>H(+2)=J, o(+3)=r, l(+4)=o, a(+5)=f:: por tanto el texto cifrado sería C="Jrof"

Cifrado por transposición [2].

Los cifrados por sustitución conservan el orden de los símbolos de texto normal, pero los disfrazan. Los cifrados por transposición en contraste, reordenan las letras pero no las disfrazan. Un ejemplo de cifrado por transposición es la transposición columnar. La clave del cifrado es una palabra o frase que no contiene letras repetidas. En este ejemplo, la clave es MEGABUCK. El propósito de la clave es numerar las columnas, estando la columna 1 bajo la letra clave más cercana al inicio del alfabeto, y así sucesivamente. El texto normal se escribe horizontalmente en filas, el texto cifrado se lee por columnas (o también por filas según el método empleado), comenzando por la columna cuya letra clave es la más baja.

M E G A B U C K	Texto normal:
7 4 5 1 2 8 3 6	
p l e a s e t r	pleasetransferonemilliondollarstomyswissb
a n s f e r o n	ankaccountsixtwotwo
e m i l l i o n	
d o l l a r s t	Texto cifrado:
o m y s w i s s	
b a n k a c c o	AFLLSKSOSELAWAIATOOSSCTCLNMO
u n t s i x t w	MANTESILYNTWRNNTSOWDPAEDOBU
o t w o a b c d	OERIRICXB

Para descifrar un cifrado por transposición, el criptoanalista debe primero ser consciente de que está tratando con un cifrado por transposición. Al observar la frecuencia de E, T, A, O, I, N, etc., es fácil ver si se ajustan al patrón usual del texto normal. De ser así, es evidente que se trata de un cifrado por transposición, pues en tal cifrado cada letra se representa a sí misma.

El siguiente paso es adivinar la cantidad de columnas. En muchos casos, puede adivinarse una palabra o frase probable por el contexto del mensaje. Por ejemplo, supóngase que nuestro criptoanalista sospecha que la frase de texto normal milliondolars aparece en algún lugar del mensaje. Observe que los diagramas MO, IL, LL, LA, IR y OS ocurren en el texto cifrado como resultado de que esta frase da la vuelta. Si se hubiera usado una clave de longitud siete, habrían ocurrido los diagramas

MD, IO, LL, LL, IA, OR y NS. De hecho, para cada longitud de clave, se produce un grupo diferente de diagramas de texto cifrado. Buscando las diferentes posibilidades, el criptoanalista con frecuencia puede determinar fácilmente la longitud de la clave.

El paso restante es ordenar las columnas. Cuando la cantidad de columnas, k, es pequeña, puede examinarse cada uno de los pares de columnas k(k-1) para ver si la frecuencia de sus diagramas es igual a la del texto normal. El par con mejor concordancia se supone correctamente ubicado. Ahora cada columna restante se prueba tentativamente como el sucesor de este par. La columna cuyas frecuencias de digramas y trigramas produce la mejor concordancia se toma tentativamente como correcta. La columna antecesora se encuentra de la misma manera. El proceso completo se repite hasta encontrar un orden potencial. Es probable que el texto normal sea reconocible en algún punto (por ejemplo, sí aparece milloin, quedará claro dónde está el error).

Rellenos de una sola vez [2].

La construcción de un cifrado inviolable en realidad 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, conocido como XOR y cuya tabla de valores lógicos puede verse a continuación.

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

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 digrama (combinación de dos letras) y cada trigrama (combinación de tres letras). Como ejemplo de cifrado basado en relleno de una sóla vez, 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

Si se procede ahora a descifrarlo con la clave de codificación, se obtiene el mensaje original ya que aplicamos la función XOR 2 veces:

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

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 moderna [2].

La criptografía moderna usa las mismas ideas básicas que la criptografía tradicional, la transposición y la sustitución, pero su orientación es distinta. Mientras la criptografía tradicional usaba algoritmos sencillos y claves muy largas para la seguridad, hoy en día es cierto la afirmación contraria: el objetivo es hacer algoritmos de cifrado tan complicados y rebuscados que incluso si el criptoanalista obtiene cantidades enormes de texto cifrado a su gusto, no será capaz de entender nada y por tanto de descifrarlo.

Las transposiciones y sustituciones pueden implantarse mediante circuitos sencillos. En la figura siguiente se muestran dos dispositivos conocidos como caja P, que se usa para efectuar una transposición de una entrada de 12 bits; y otro dispositivo conocido como caja S, en el cual ingresa un texto normal de 3 bits y sale un texto cifrado de 3 bits. La potencia real de estos elementos básicos sólo se hace aparente cuando ponemos en cascada una serie completa de estas cajas para formar un cifrado de producto como podemos ver en la figura siguiente.

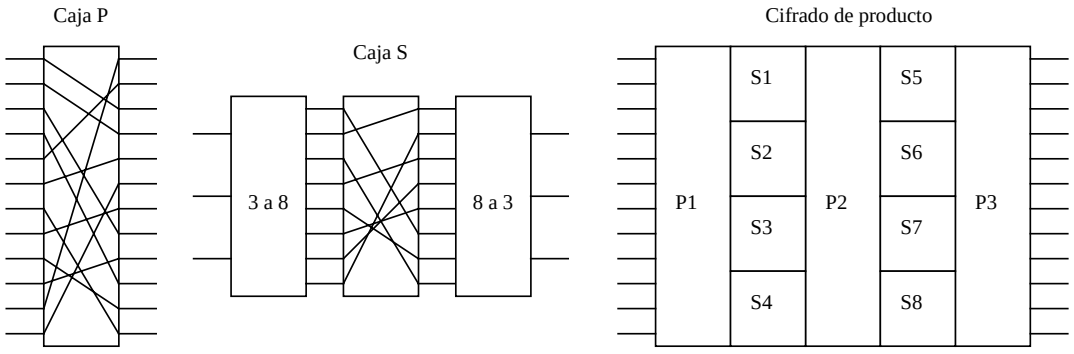

Figura 9: Ejemplo de Cifrado de Producto mediante Cajas P y Cajas S.

El cifrado moderno se divide actualmente en cifrado de clave privada y cifrado de clave pública:

En el cifrado de clave privada las claves de cifrado y descifrado son la misma (o bien se deriva de forma directa una de la otra), debiendo mantenerse en secreto dicha clave. Ejemplo: DES (Data Encryption Standar), DES triple e IDEA (International Data Encryption Algorithm). El cifrado de clave privada, es más rápido que el de clave pública (de 100 a 1000 veces), y por tanto se utiliza generalmente en el intercambio de información dentro de una sesión. Estas claves también son conocidas como claves de sesión o de cifrado simétricas, ya que en ambos extremos se posee la misma clave.

En el cifrado de clave pública, las claves de cifrado y descifrado son independientes, no derivándose una de la otra, por lo cual puede hacerse pública la clave de cifrado siempre que se mantenga en secreto la clave de descifrado. Ejemplo: Cifrado RSA (Rivest, Shamir, Adleman).

El cifrado de clave pública es más lento y por tanto se utiliza para intercambiar las claves de sesión. Como este algoritmo utiliza dos claves diferentes, una privada y otra pública el cifrado se conoce como cifrado asimétrico

Algoritmos de Curva Elíptica

Son los más recientes dentro del campo de los sistemas de clave pública. En general se creen que son bastante seguros, pero no ha sido demostrado. Existe un tipo de curvas que recientemente se ha revelado extremadamente vulnerable, por lo que éstas no deben usarse en criptografía. Existen muchos otros tipos de curvas, pero han de ser cuidadosamente examinadas para comprobar su idoneidad como base para un código de cifrado de datos. La valía de los sistemas de curvas elípticas permanece hoy por hoy bajo dudas.[11]

Un **CCE** basa su seguridad en el **P**roblema del **L**ogaritmo **D**iscreto **E**líptico (**PLDE**), es decir, en el **P**roblema del **L**ogaritmo **D**iscreto (**PLD**) definido en el grupo de puntos racionales de una curva elíptica. El **PLD** es encontrar un número entero *x* tal que *ax* = *b* donde *a* Î á*b*ñ es el subgrupo generado por *b*, los múltiplos de *b*, y *b* Î *G* un grupo. Entonces cuando tomamos como grupo *G* a *E*(*K*) el anterior problema se denomina **PLDE** [9].

La forma más simple de ver cómo trabaja un **CCE** basado en el **PLD** es usando el esquema de ElGamal que consiste en los siguientes pasos: se tienen como conocidos a una curva *E* definida sobre el campo *K*, y *n* = #(*E*(*K*)) y un punto racional *P* (además supongamos a el grupo cíclico).

a) **Generación de llaves**

1.- Se elige un número $x \in [1, n - 1]$
2.- Se calcula el punto $H = xG \in E(K)$
3.- La llave pública será el punto *H*
4.- La llave privada será el número *x*

b) **Proceso de Cifrado**

1.- Se elige un número aleatorio $k \in [1, n - 1]$
2.- Se calcula el punto $c = (kG, P_m + kH)$
3.- Se transmite el punto $c = (c_1, c_2)$

c) **Proceso de Descifrado**

1.- El receptor recobra a *Pm* calculando $c_2 - xc_1 = P_m + kH - xkG = P_m$
Observe que la seguridad del sistema está basada en la dificultad de calcular *x* a partir de *H* y *G*, es decir, que el sistema será roto si se resuelve el **PLDE**. Respecto a la implemetación la dirección es encontrar eficaces formas de calcular múltiplos de puntos racionales *kP*.

El **PLDE** ha sido intensamente estudiado los últimos años, conjuntamente con el **P**roblema de la **F**actorización **E**ntera **PFE**, atraen gran parte de la atención de la comunidad criptográfica, en lo que se refiere a sistemas de llave pública.

El primer gran paso que se dio en la solución del **PLDE** fue en 1993, donde el método **MOV** (**M**enezes, **O**kamoto, **V**anstone) reduce el **PLDE** definido sobre una curva elíptica sobre F_{2n} a el **PLD** en el campo $F_{(2n)k}$. Este método es eficiente sobre las curvas elípticas supersingulares, lo que causó que éstas se desecharan para

propósitos criptográficos. Esencialmente el **MOV** consiste en incluir a el grupo de puntos racionales $E(F_{2n})$ dentro de una extensión $F_{(2n)k}$ del campo finito F_{2n} . El método es totalmente impráctico sobre curvas no supersingulares [9].

Otro acercamiento en la solución del **PLDE** fue alrededor de Sep. de 1997, donde el **PLDE** fue resuelto para una clase del tipo de curvas elípticas llamadas anómalas, es decir cuando su traza es ±1. Este método consiste en incluir a el grupo $E(Z_p)$ en el grupo aditivo Z_p, usando al conjunto de puntos del grupo Z_{p2} cuya reducción módulo p es el punto al infinito y no es factible su generalización a otro tipo de curvas, ya que el método necesita que el número de puntos racionales sea igual a el número de elementos del campo, o sea que la traza sea 1.

Aunque ha sido grande el esfuerzo para resolver el PLDE, la forma más eficaz que se ha encontrado es un método (de Pollard) que tiene una complejidad $\sqrt{n}$, donde n es el tamaño del campo, es decir, que en un campo Z_p donde p ~2160 tendríamos que efectuar $\sqrt{(2^{160})} = 2^{80}$ operaciones para poder calcular un logaritmo. Se puede conjeturar que el poder de cómputo necesario para efectuar esta cantidad de operaciones es de alrededor 1×10^{12} *mipsy* y este poder de cómputo podrá ser alcanzado no antes del año 2012. Actualmente en el mundo contamos con un poder de cómputo estimado de 1×10^{9} *mipsy*, lo que hace a los CCE de 160 bits seguros hasta al menos el año 2012 [9].

Una simple comparación con el PFE, es que con los mejores métodos de factorización hoy en día (la criba de campos numéricos general), con el mismo poder de cómputo (1×10^{12} *mipsy*) podríamos factorizar a un número entero producto de dos números primos de tamaño 1024 bits, lo que permite afirmar que los CCE proporcionan la misma seguridad que el sistema RSA con una considerable menor longitud del campo donde se trabaja. Dando así mejores condiciones de implementación donde se tengan reducidas capacidad de operación, poder de cómputo reducido, espacio en circuitos integrado limitado, etcétera [9].

A continuación se describirá un estándar que usa CCE para encriptar datos.

Se determina el campo F_q, la curva E y un punto $P \in E(F_q)$ de orden n.

Generación de llaves

1) Se elige un número aleatorio $d \in [1, n - 1]$
2) Se calcula el punto $Q = dP$
3) La llave pública es el punto Q
4) La llave privada es el entero d

Proceso de encriptamiento

El usuario **A** desea mandar el mensaje M a el usuario **B**
1) Se representa el mensaje como $M = (m_1, m_2) \in F_q 2$
2) Se selecciona un número aleatorio k
3) Se calcula el punto $(x_1, y_1) = kP$
4) Se calcula el punto $(x_2, y_2) = kQ$
5) Se combinan los elementos m_1, m_2, x_2 y y_2 obteniendo c_1, c_2
6) Se transmite a $c = (x_1, y_1, c_1, c_2)$

Proceso de desencriptamiento

1) El usuario **B** calcula a $d(x_1, y_1)$ que es (x_2, y_2)
2) Se recobra el mensaje M a partir de c_1, c_2, x_2 y y_2

Lo anterior se verá a continuación descrito con un pequeño ejemplo concreto.
Considerando a el campo $F_2{}^3$, y la **BNO** generada por $\beta = \alpha^3$ de 2, entonces los elementos de $F_2{}^3$ como potencias de b pueden representarse como:

β	$= \beta$	~(001)
β^2	$= \beta^2$	~ (010)
β^3	$= \beta^{2 \exp 2} + \beta$	~ (101)
β^4	$= \beta^{2 \exp 2}$	~ (100)
β^5	$= \beta^2 + \beta^{2 \exp 2}$	~ (110)
β^6	$= \beta + \beta^2$	~ (011)
β^7	$= \beta + \beta^2 + \beta^{2 \exp 2}$	~ (111)

La curva $E : y^2 + xy = x^3 + x^2 + 1$ definida sobre este campo tiene $2 \bullet 7$ puntos racionales y estos son los siguientes:

$P_0 = O$	$P_1 = (\beta,\beta)$	$P_2 = (\beta, 0)$	$P_3 = (\beta^2, \beta^2)$
$P_4 = (\beta^2, 0)$	$P_5 = (\beta^3,b)$	$P_6 = (\beta^3, \beta^4)$	$P_7 = (\beta^4, \beta^4)$
$P_8 = (\beta^4, 0)$	$P_9 = (\beta^5, \beta^2)$	$P_{10} = (\beta^5, \beta^4)$	$P_{11} = (\beta^6, \beta)$
$P_{12} = (\beta^6, \beta^2)$	$P_{13} = (0, \beta^7)$		

y el grupo es isomorfo a $Z^2 \times Z^7$, donde $2P_{13} = O$, y el subgrupo de orden 7 es
P_1, $2P_1 = P_7$, $3P_1 = P_4$, $4P_1 = P_3$, $5P_1 = P_8$, $6P_1 = P_2$, $7P_1 = O$.
La tabla de suma es:

+	*P*1	*P*2	*P*3	*P*4	*P*5	*P*6	*P*7	*P*8	*P*9	*P*10	*P*11	*P*12	*P*13
*P*1	*P*7	*P*0	*P*8	*P*3	*P*12	*P*10	*P*4	*P*2	*P*5	*P*9	*P*6	*P*13	*P*11
*P*2	*P*0	*P*8	*P*4	*P*7	*P*9	*P*11	*P*1	*P*3	*P*10	*P*6	*P*13	*P*5	*P*12
*P*3	*P*8	*P*4	*P*1	*P*0	*P*6	*P*12	*P*2	*P*7	*P*11	*P*13	*P*5	*P*10	*P*9
*P*4	*P*3	*P*7	*P*0	*P*2	*P*11	*P*5	*P*8	*P*1	*P*13	*P*12	*P*9	*P*6	*P*10
*P*5	*P*12	*P*9	*P*6	*P*11	*P*4	*P*0	*P*13	*P*10	*P*7	*P*1	*P*2	*P*3	*P*8
*P*6	*P*10	*P*11	*P*12	*P*5	*P*0	*P*3	*P*9	*P*13	*P*2	*P*8	*P*4	*P*1	*P*7
*P*7	*P*4	*P*1	*P*2	*P*8	*P*13	*P*9	*P*3	*P*0	*P*12	*P*5	*P*10	*P*11	*P*6
*P*8	*P*2	*P*3	*P*7	*P*1	*P*10	*P*13	*P*0	*P*4	*P*6	*P*11	*P*12	*P*9	*P*5
*P*9	*P*5	*P*10	*P*11	*P*13	*P*7	*P*2	*P*12	*P*6	*P*1	*P*0	*P*8	*P*4	*P*3
*P*10	*P*9	*P*6	*P*13	*P*12	*P*1	*P*8	*P*5	*P*11	*P*0	*P*2	*P*3	*P*7	*P*4
*P*11	*P*6	*P*13	*P*5	*P*9	*P*2	*P*4	*P*10	*P*12	*P*8	*P*3	*P*7	*P*0	*P*1
*P*12	*P*13	*P*5	*P*10	*P*6	*P*3	*P*1	*P*11	*P*9	*P*4	*P*7	*P*0	*P*8	*P*2
*P*13	*P*11	*P*12	*P*9	*P*10	*P*8	*P*7	*P*6	*P*5	*P*3	*P*4	*P*1	*P*2	*P*0

Ahora si se toma por ejemplo a P_1 de orden 7 como punto base, es ejemplo queda como:

Generación de llaves

1) El usuario **B** selecciona a $d = 6$
2) **B** calcula el punto $Q = 6P_1 = P_2 = (\beta, 0)$
3) La llave privada es 6
4) La llave pública es P_2

Proceso de Cifrado

1) **A** representa el mensaje M como $m_1 = \beta^5$ y $m_2 = \beta^2$
2) **A** selecciona el número aleatorio $k = 2$
3) **A** calcula el punto $2P_1 = P_7 = (\beta^4, \beta^4)$
4) **A** calcula el puno $2Q = 2P_2 = P_8 = (\beta^4, 0) = (x_2, y_2)$
5) **A** calcula el elemento $x_2^{\,3} = (\beta^4)^3 = \beta^5$
6) **A** forma a $x_4 = (100)$ concatenando los primeros dígitos binarios de x_2 y el último de $x_2^{\,3}$. De forma simétrica forma a $y_4 = (110)$, es decir, tomando los dos primeros bits de $x_2^{\,3}$ y el último de x_2
7) **A** calcula los puntos
$c_1 = (m_1 + y_2)x_4 = (\beta^5 + 0)\ \beta^4 = \beta^2$
$c_2 = (m_2 + x_2)y_4 = (\beta^2 + b4)\ \beta^5 = \beta^3$
8) **A** transmite el mensaje
$c = (\beta^4, \beta^4, \beta^2, \beta^3)$

Proceso de Descifrado

1) **B** calcula el punto $6(\beta^4, \beta^4) = 6P_7 = P_8$
2) **B** calcula a $x_4 = (100)$ y $y_4 = (110)$ de la misma forma que **A**
3) **A** calcula los elementos
$m_1 + y_2 = b_2 x_4^{-1} = \beta^5$
$m_2 + x_2 = b_3 y_4^{-1} = \beta^5$
4) Finalmente **B** recobra el mensaje M por medio de
$m_1 = \beta^5 + 0 = \beta^5$
$m_2 = \beta^5 + \beta^4 = \beta^2$

Otro esquema importante en criptografía es el de firma digital, con **CCE** puede ser implementado un sistema de firma digital análogo al **DSA** (**D**igital **S**ignature **A**lgorithm), que también basa su seguridad en la presunta imposibilidad de resolver el **PLDE** [9].

ANEXO 3 MANUAL DE USUARIO APLICACIÓN SEGURIDAD

La Aplicación principal **Seguridad**, consta de 2 aplicaciones básicas diseñadas tanto para cifrar/descifrar texto plano o archivos, según los requerimientos del usuario; lo que brinda la oportunidad de conocer de manera más profunda la manera como trabajan los algoritmos criptográficos y además se explica paso a paso como funciona una PKI (Infraestructura de Claves Públicas) híbrida, para garantizar un alto nivel de seguridad en el transporte de los documentos digitales a través de una red.

Al iniciar se desplegará en pantalla la siguiente vista:

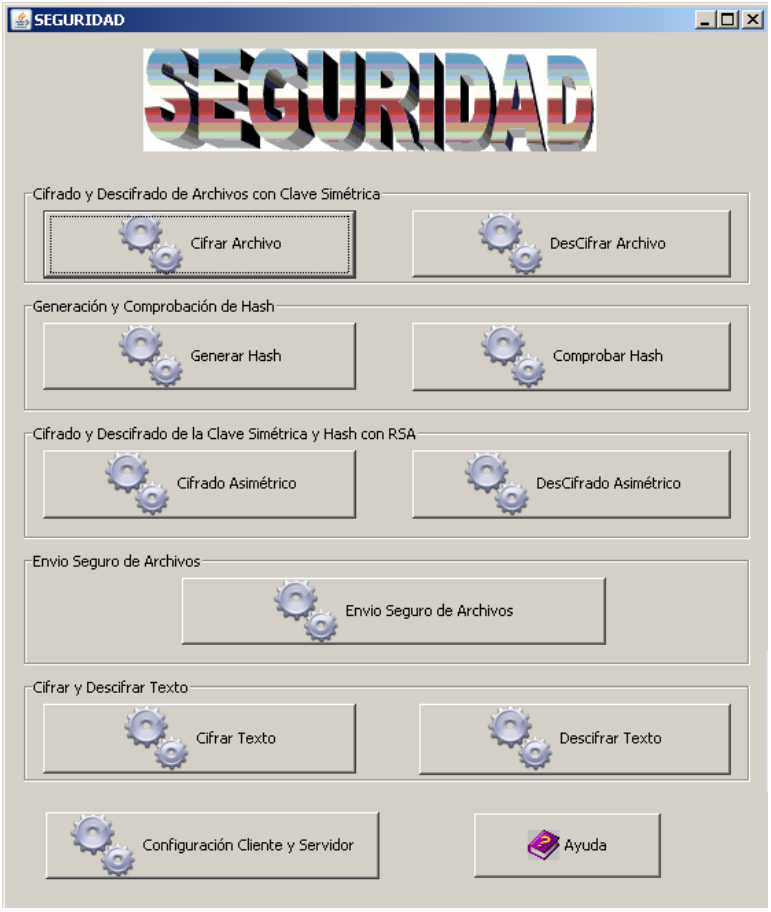

Figura 10: Interfaz Principal Aplicación Seguridad

Simultáneamente se desplegará un aviso de información que le indica al usuario en que puerto, está funcionando la clase Servidor que estará atenta a los archivos que pueda enviar la aplicación desde otro punto de red.
Este aviso se muestra a continuación:

Figura 11: Interfaz de información del Servidor Activo

En este caso se tiene por defecto el puerto 20000, pero se puede escoger el de la preferencia del usuario, mediante el botón "Configuración Cliente y Servidor" de la aplicación principal, la cual desplegará en pantalla:

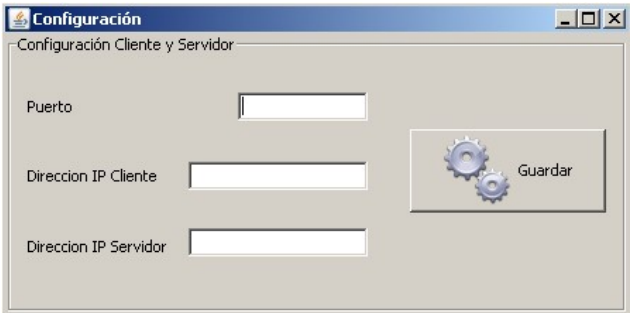

Figura 12: Interfaz de Configuración del Cliente y Servidor

Mediante estas aplicaciones se pueden observar a manera de información como se efectúan las operaciones de cifrado y descifrado de los diferentes tipos de algoritmos (simétricos, asimétricos y de hash), teniendo en cuenta que a la hora de utilizar el algoritmo RSA se debe cargar el certificado digital correspondiente para extraer las llaves privada y pública. Lo anterior queda claramente ilustrado a continuación:

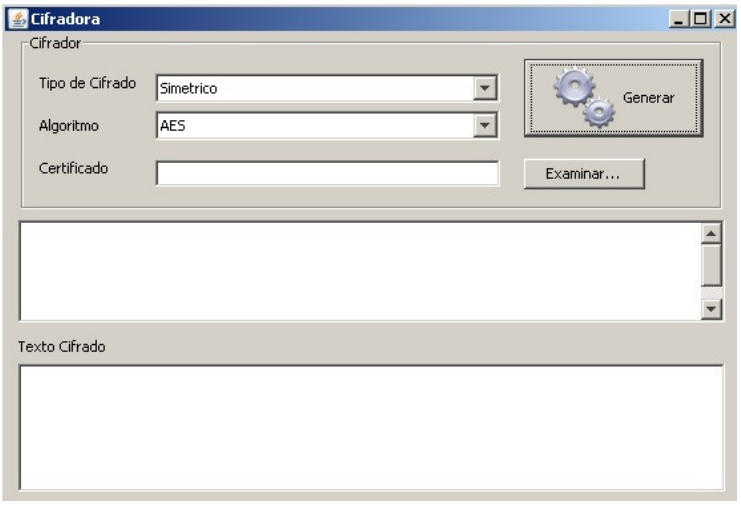

Figura 13: Interfaz de Cifrado de Texto

En la primer área de texto se escribe el texto plano a cifrar y en la segunda área de texto se muestran tanto los caracteres generados (texto cifrado) como su correspondiente representación en hexadecimal (del texto cifrado).

Para descifrar se debe presionar primero el botón "Cargar Entrada", para ver los datos concernientes al algoritmo empleado y se debe digitar el password del certificado digital en el caso en que el algoritmo sea el RSA, ya que este es el único que emplea certificados digitales, y se presiona el botón "DesCifrar".

A continuación se mostrará la interfaz generada.

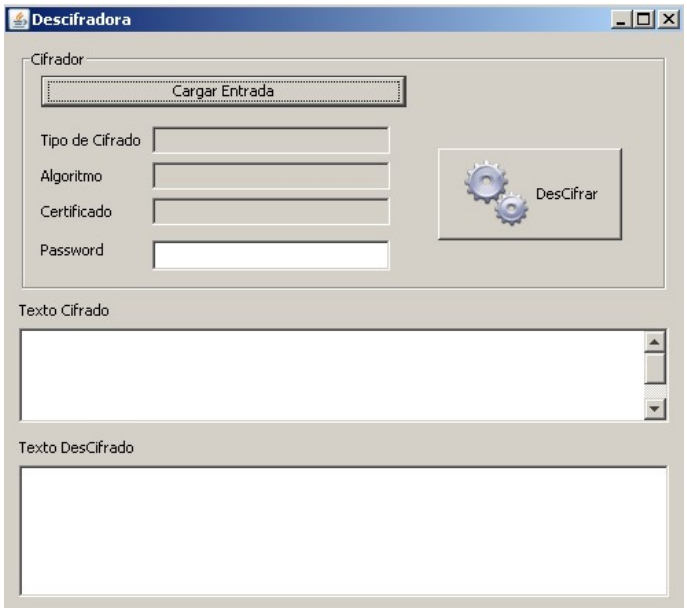

Figura 14: Interfaz Descifradora de Texto

En la primer área de texto se cargan tanto los caracteres como su equivalente en hexadecimal del texto cifrado y en la segunda área de texto se muestra el texto original o descifrado.

Para cifrar archivos, se deben seguir los siguientes pasos en orden consecutivo:

PASO 1. Presionar el botón "Cifrar Archivo", lo que generará la siguiente ventana:

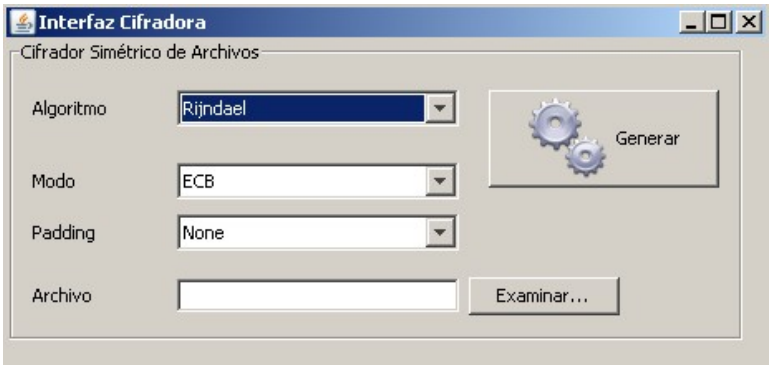

Figura 15: Interfaz Cifradora Simétrica de Archivos

A continuación se debe seleccionar el algoritmo simétrico que se desea utilizar. Seguidamente se selecciona el modo y el padding, que indican la manera como ha de rellenarse los archivos para completar las medidas estándar para el tamaño soportado por cada uno de los algoritmos, teniendo en cuenta que los algoritmos (BouncyCastle): AES(Rijndael), Camellia, CAST6, IDEA, RC6, RC4, Serpent, Twofish, TEA, SEED y Noekeon funcionan con los modos: ECB, CBC, OFB, CFB, CTS, GOFB (a excepción del AES), OpenPGPCFB, PGPCFB y con los siguientes paddings: PKCS7Padding, TCBPadding, X923Padding, ZeroBytePadding.

Mientras que los algoritmos (Cryptix): DES, Blowfish, SKIPJACK, Square, MARS y TripleDES (DESede), no funcionan con los modos: CTS, GOFB y PGPCFB; en cuanto a los paddings funcionan con: None, NoPadding y PKCS#5.

Lo anterior debido a que se toman de dos API's distintas que funcionan de manera diferente, los de la primera API (BouncyCastle) son más "actuales", manejan más algoritmos, modos y paddings que los de la segunda API (Cryptix).

Todos los modos y paddings funcionan según su proveedor, aunque a veces se pueden alterar los archivos de manera imperceptible (solo se puede comprobar mediante las funciones de hash) en cuanto a lo que es imágen y video, para los documentos WORD no hay ningún problema (siempre y cuando solo se necesite del cifrado simétrico), aunque se aconseja siempre usar el modo ECB por restricciones de cifrado asimétrico ya que para este tipo de cifrado no se pueden cifrar archivos de más de 157 bytes y los demás modos generan archivos de tamaño ligeramente superior, y en cuanto al padding los que garantizan la total integridad de la información son PKCS7Padding para los algoritmos de BouncyCastle y PKCS#5 para los de Cryptix.

Mediante el botón "Examinar..." se busca la ruta del archivo a cifrar simétricamente.

Por último se presiona el botón "Generar" para cifrar el archivo con las anteriores características.

La llave simétrica generada y utilizada se almacena en la carpeta info y el archivo cifrado con extensión "encrypted" se almacena en la carpeta enviar.

PASO 2: Presionar el botón "Generar Hash", se generará la siguiente interfaz:

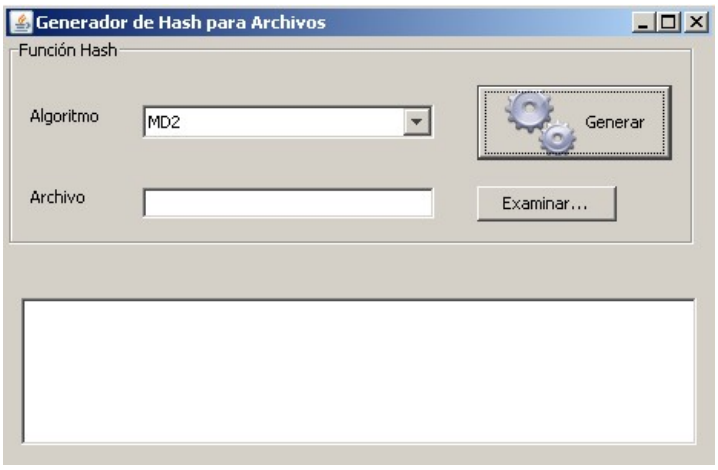

Figura 16: Interfaz Generadora de Hash de Archivos

Se escoge el algoritmo de hash a emplear, se escoge el archivo mediante la ventana de apertura mostrando su ruta y se presiona el botón "Generar" para extraer la reseña o digesto del archivo para comprobar su integridad posteriormente.

En el Área de Texto se desplegará a manera de ilustración la función hash generada y ésta se almacenara en un archivo de extensión igual al nombre del algoritmo de hash empleado, esto se hará en la carpeta info.

Los algoritmos de hash incorporados a la aplicación son: MD2, MD4, MD5, RIPEMD128, RIPEMD160, RipeMD320, SHA, SHA0, SHA1, SHA-512, Tiger, Whirlpool y GOST3411.

PASO 3: Pesionar el botón "Cifrado Asimétrico", lo que generará la siguiente interfaz:

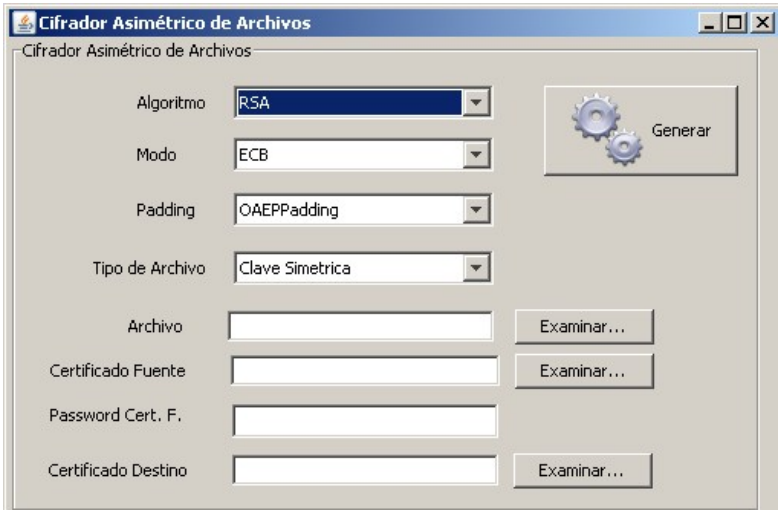

Figura 17: Interfaz Cifradora Asimétrica de Archivos

El algoritmo, modo y padding vienen por defecto en los valores predeterminados y por ahora no hay más opciones debido a que solo los proveedores anteriores pudieron ser empleados, librerías como la gnu-crypto no son compatibles con la versión jdk1.6.0, esta librería contiene algunos algoritmos que no se utilizan en esta aplicación y además contiene otros modos y paddings para el RSA, en cuanto a los otros algoritmos asimétricos como DSA, ElGamal y Rabin, no se utilizan debido a que no se consiguieron aplicaciones que generaran certificados digitales compatibles con las librerías utilizadas, además de encontrar poca documentación de los modos y paddings empleados.

A continuación, se escoge el tipo de archivo que se va a cifrar asimétricamente, ya sea la clave simétrica empleada en el paso 1 o el hash generado en el paso 2, por motivos de espacio se tuvo la necesidad de comprimir en una sola la interfaz de cifrado asimétrico, al escoger el archivo se debe tener en cuenta que corresponda con el tipo de archivo seleccionado previamente, ya que ambos tipos se encuentran en la carpeta info, se debe llenar el resto de campos, lo campos certificado Fuente y Password Cert. F. hacen referencia al certificado del emisor de los archivos y el certificado destino hace referencia a la persona a quien van dirigidos los archivos, ambos se pueden cargar del repositorio de certificados que en nuestro caso es la carpeta certificados, es de anotar que dichos certificados (con llaves RSA) fueron creados con una aplicación de la IBM llamada: ktl20sta, la cual está desarrollada en Java (archivo jar ejecutable), lo anterior se hizo para ilustrar la utilización de certificados, aunque en la vida real se debe tener una Autoridad Certificadora (AC) que expida los certificados cada vez que se requieran y que a la vez certifique que la persona que los utiliza es quien dice ser, lo anterior se hace mediante un hash cifrado (con la llave privada de la AC) del certificado que manda la AC junto con el certificado y que debe comprobarlo quien lo recibe.

Por razones de simplicidad, se ha establecido la regla, a la hora de crear la clave privada, de conservar el mismo password epleado para la creación del certificado

además de establecer el login o identificador de la clave privada como el nombre del certificado seguido de la palabra “rsa”.

Para cifrar la clave simétrica se debe tener en cuenta que la aplicación lo que hace es: tomar el certificado de destino, extraer la llave pública del destinatario y con esta cifrar la clave simétrica para que de esta manera el destinatario solamente pueda descifrar con su clave privada.

Para cifrar el hash lo que la aplicación hace es: obtener acceso a la clave privada del emisor mediante el password (autorización) utilizando su certificado y cifra con esta clave privada el hash, para que de esta manera en el destino se pueda comprobar mediante la llave pública del emisor que ha sido esta persona quien ha mandado el hash cifrado (validación), además de comprobar la integridad del archivo mediante el hash descifrado (integridad de la información).

El hash del archivo cifrado asimétricamente con la clave privada es lo que se conoce como firma digital.

Los archivos anteriormente explicados se crean y se almacenan en la carpeta enviar bajo la extensión: "encrypted".

PASO 4: Presionar el botón: "Envío Seguro de Archivos", al oprimir este botón la aplicación busca los archivos generados recientemente en la carpeta enviar y los manda a través de sockets por la red, no se necesita cargar a la red con seguridad en el canal de comunicación (SSL, entre otros) debido a que la información va cifrada, además se mandan 4 archivos más que contienen información concerniente los algoritmos empleados pero que de ninguna manera exponen información delicada.

Es aconsejable utilizar números de puerto altos ya que tienen poca probabilidad de estar siendo utilizados, los archivos enviados mediante la aplicación Seguridad son:

1. entArchHash, archivo que contiene lo concerniente al algoritmo de hash utilizado

2. CifASHash, archivo que contiene información del algoritmo asimétrico utilizado en este caso RSA y el nombre del archivo de hash cifrado asimétricamente.

3. entradaArch, archivo que guarda lo referente al cifrado simétrico del archivo o documento.

4. CifAsAr, archivo que guarda lo referente al algoritmo asimétrico utilizado para cifrar la clave simétrica, en este caso el RSA y guarda también el nombre de esta clave simétrica.

5. Archivo (que puede ser un documento) cifrado simétricamente.

6. Clave Simétrica (Usada para cifrar el anterior archivo) cifrada Asimétricamente con RSA mediante el certificado de destino para obtener la llave pública.

7. Hash cifrado Asimétricamente con RSA mediante el certificado de origen utilizando la llave privada, a la cual se accesa mediante un password.

Los cuatro primeros archivos guardan información que en caso de ser interceptada necesitaría todos los detalles de los algoritmos, saber sus números de identificación

dentro del programa y más importante aún, el uso de los certificados que le dan un valor agregado de seguridad.

Para Descifrar Archivos los pasos a seguir son:

PASO 1: comprobar en la carpeta recibido, que hayan llegado los tres archivos cifrados y presionar el botón: "Descifrado Asimétrico", lo que generará la siguiente interfaz:

Figura 18: Interfaz Descifradora Asimétrica de Archivos

En la anterior ilustración aparece por defecto en el tipo de archivo la opción Clave Simétrica, a continuación se debe presionar el botón: "Cargar entrada" para ver el algoritmo, modo y padding asimétrico empleado (por ahora solo se tiene RSA/ECB/OAEPPadding), el nombre del archivo que contiene la clave simétrica empleada y el certificado del origen (Certificado E.) para que la aplicación pueda reconocerlo y extraer la llave pública del emisor.

En el campo Certificado P.(Certificado Personal) el destinatario carga su certificado y en el campo Password C. P. digita su contraseña para que la aplicación obtenga su clave privada.

Para descifrar el hash se cambia el tipo de archivo a hash y se vuelve a presionar "Cargar Entrada" para ver la información concerniente al hash.

Una vez descifrados el hash y la clave simétrica estos se almacenan en la carpeta archivos.

PASO 2: Presionar el botón "DesCifrar Archivo" para que la aplicación utilice la llave simétrica de la carpeta archivos y la aplique al archivo cifrado simétricamente de la carpeta recibido, lo que se obtiene es una copia del archivo original en la carpeta archivos. Esto se muestra a continuación:

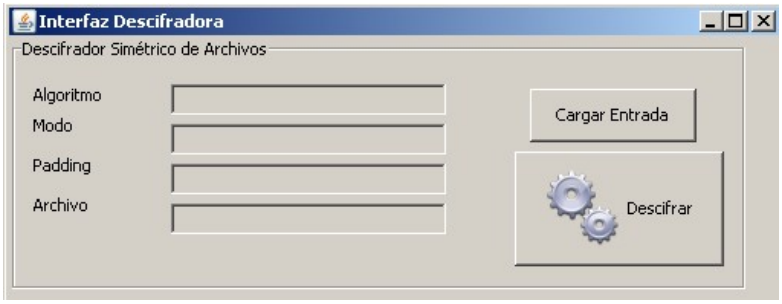

Figura 19: Interfaz Descifradora Simétrica de Archivos

Al presionar el botón "Cargar Entrada", en los campos Algoritmo, modo y padding se muestra la información pertinente al algoritmo simétrico usado y en el campo archivo se muestra el nombre del archivo que se va a generar.

Al presionar el botón "Descifrar" realiza el descifrado del archivo y lo almacena en la carpeta archivos.

PASO 3: Presionar el botón "Comprobar Hash" se despliega la siguiente interfaz:

Figura 20: Interfaz de Comprobación de Hash

Al oprimir el botón "Cargar Entrada" se muestra la información del algoritmo de hash utilizado y el nombre del archivo al que se le aplicó el hash y al oprimir el botón "Comprobar Hash" la aplicación carga el archivo de hash descifrado y almacenado en archivos, extrae la función hash para compararla con el hash generado a partir del archivo descifrado en la carpeta archivos si ambos coinciden el archivo ha mantenido su integridad y se mostrarán los resultados en el área de texto indicando que los hash son idénticos, caso contrario, se indicará que no son iguales, por lo que se asumirá que ha habido cambios en la estructura del archivo ya sea debido a daños o alteraciones.

ANEXO 4 DISPOSITIVOS DE SEGURIDAD

La escítala

Ya en siglo V antes de J.C. los lacedemonios, un antiguo pueblo griego, usaban el método de la escítala para cifrar sus mensajes. El sistema consistía en una cinta que se enrollaba en un bastón y sobre el cual se escribía el mensaje en forma longitudinal como se muestra en la Figura 27. [14]

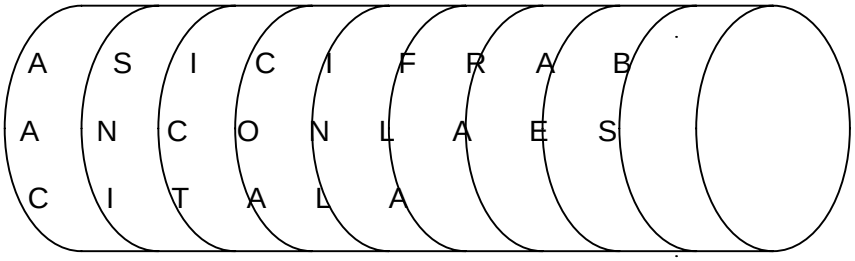

Figura 21: Cifrado mediante Sistema de Escítala.

Una vez escrito el mensaje, la cinta se desenrollaba y era entregada al mensajero; si éste era interceptado por cualquier enemigo, lo único que se conseguía era un conjunto de caracteres o letras distribuidas al parecer de forma aleatoria en dicha cinta. Incluso si el enemigo intentaba enrollar la cinta en un bastón con diámetro diferente, el resultado obtenido era un conjunto de letras escritas una a continuación de otra sin sentido alguno. Por ejemplo, en el caso de la figura 27, la cinta llevará el mensaje M = ASI CIFRABAN CON LA ESCITALA si bien en ella sólo podrá leerse el criptograma C = AACSNIICTCOAINLFLARAAEBS. La clave del sistema residía precisamente en el diámetro de aquel bastón, de forma que solamente el receptor autorizado tenía una copia exacta del mismo bastón en el que enrollaba el mensaje recibido y, por tanto, podía leer el texto en claro. En este sistema no existe modificación alguna del mensaje, se lograba el objetivo de la confidencialidad, en tanto que la integridad estaba en entredicho y dependía de lo aguerrido y fiel que fuese nuestro mensajero. Si la cinta era robada y se cambiaban los caracteres, podría llegar al receptor un mensaje sin sentido y, lo que es peor, con un duplicado del bastón original podía enviarse un mensaje con sentido completamente distinto al encomendado al mensajero. [14]

El cifrador de Polybios

A mediados del siglo II antes de J.C., se creó el cifrador por sustitución de caracteres más antiguo que se conoce. Atribuido al historiador griego Polybios, el sistema de cifra consistía en hacer corresponder a cada letra del alfabeto un par de letras que indicaban la fila y la columna en la cual aquella se encontraba, en un recuadro de 5 x 5 = 25 caracteres, transmitiéndose por tanto en este caso el mensaje como un criptograma. En la Figura 28 se muestra una tabla de cifrar de Polybios adaptada al inglés, con un alfabeto de cifrado consistente en el conjunto de letras A, B, C, D y E aunque algunos autores representan el alfabeto de cifrado como los números 1, 2, 3, 4 y 5.

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

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	IJ	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Figura 22: Tablas de Cifrar de Polybios.

Acorde con este método, la letra A se cifrará como AA, la H como BC, etc. Esto significa que aplicamos una sustitución al alfabeto {A, B, C, .., X, Y, Z} de 26 letras convirtiéndolo en un alfabeto de cifrado {AA, AB, AC, ..., EC, ED, EE} de 25 caracteres, si bien sólo existen 5 símbolos diferentes {A, B, C, D, E}. Este tipo de tabla o matriz de cifrado será muy parecida a la que en el siglo XIX se utilizará en el criptosistema conocido como cifrador de Playfair y que será tratado más adelante en el apartado de cifradores poligrámicos, salvo que en este último la operación de cifra no se realiza por monogramas como en el de Polybios sino por digramas, conjunto de dos caracteres del texto en claro. [14]

El cifrador del César

Unos cincuenta años después del cifrador de Polybios, en el siglo I antes de J.C., aparece un cifrador básico conocido con el nombre genérico de cifrador del César en honor al emperador Julio César y en el que ya se aplica una transformación al texto en claro de tipo monoalfabética. El cifrador del César aplica un desplazamiento constante de tres caracteres al texto en claro, de forma que el alfabeto de cifrado es el mismo que el alfabeto del texto en claro pero desplazado 3 espacios hacia la derecha módulo n, con n el número de letras del mismo. [14]

El cifrador de Alberti

En el siglo XVI Leon Battista Alberti presenta un manuscrito en el que describe un disco cifrador con el que es posible cifrar textos sin que exista una correspondencia única entre el alfabeto del mensaje y el alfabeto de cifrado como en los casos analizados anteriormente. Con este sistema, cada letra del texto en claro podía ser cifrada con un carácter distinto dependiendo esto de una clave secreta. Se dice entonces que tales cifradores usan más de un alfabeto por lo que se denominan cifradores polialfabéticos, a diferencia de los anteriores denominados monoalfabéticos.

Como se aprecia en la Figura 29, el disco de Alberti presenta en su círculo exterior los 20 caracteres del latín, esto es, los mismos del alfabeto castellano excepto las letras H, J, Ñ, K, U, W e Y, y se incluyen los números 1, 2, 3 y 4 para códigos especiales. Por su parte, en el disco interior aparecen todos los caracteres del latín además del signo & y las letras H, K e Y. Al ser 24 los caracteres representados en cada disco, es posible definir hasta 24 sustituciones diferentes; es decir, dependiendo de la posición del disco interior la cantidad máxima de alfabetos de cifrado es igual a 24. Luego, para cifrar un mensaje, una vez establecida la correspondencia entre caracteres de ambos discos o, lo que es lo mismo, el alfabeto de cifrado, se repasa letra a letra el texto en claro del disco exterior y se sustituye cada una de ellas por la letra correspondiente del disco interior. [14]

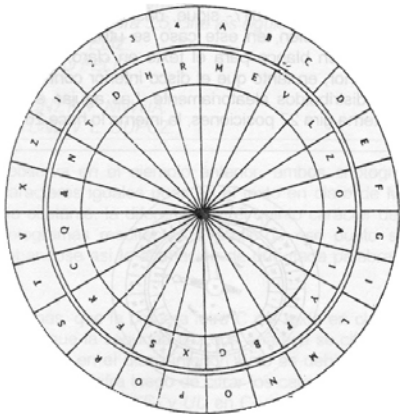

Figura 23: Disco Cifrador de Alberti.

La innovación que supone este sistema consiste en que el alfabeto de sustitución puede ser cambiado durante el proceso de cifrado, por ejemplo cada k caracteres, simplemente girando el disco interior y por tanto utilizando otro alfabeto de sustitución. [14]

Cifradores Del Siglo XIX

El cifrador de Wheatstone

El criptógrafo de Wheatstone mostrado en la Figura 30 según un invento de Decius Wadsworth desarrollado en 1817- sigue, básicamente, el mismo algoritmo de cifra que el de Alberti. Ahora bien, en este caso se utiliza el alfabeto inglés de 26 caracteres más el espacio en blanco para el texto en claro, representado de forma ordenada en el disco exterior, en tanto que el disco interior contiene solamente los 26 caracteres del lenguaje distribuidos aleatoriamente. Las agujas están engranadas de forma que cuando la externa gira 27 posiciones, la interna lo hace 26. [14]

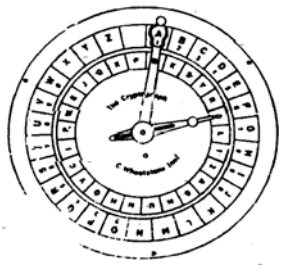

Figura 24: Máquina de Cifrar de Wheatstone.

El método de cifrado consiste en hacer girar la aguja externa en el sentido de las manecillas del reloj hasta hacer coincidir cada letra del texto en claro con la letra del disco externo y apuntar el carácter correspondiente que aparece en el círculo interior, incluso para el espacio en blanco. Por la relación de giro de las agujas, éstas se van separando una posición o letra por cada vuelta, de forma que el alfabeto de cifrado será diferente cuando se cumpla cualquiera de estas tres condiciones:

- Que se termine una palabra del texto en claro y por tanto demos un giro completo de la aguja mayor al buscar el espacio en blanco.

- Que aparezcan letras repetidas y tengamos que dar toda una vuelta completa al buscar la segunda. No obstante, según los autores, en este caso es posible

también omitir cifrar la letra repetida o bien cifrar ambas como una única letra poco usual, por ejemplo la letra Q.

- Que las letras de una palabra no vengan en orden alfabético. Es decir, si ciframos la palabra CELOS no alcanzamos a dar la vuelta completa al disco exterior, en tanto que la palabra MUJER implica dos vueltas y HOMBRE significa tres. No trate de encontrar ningún mensaje subliminal en estas tres palabras y sus vueltas.

La importancia de este cifrador está en que cada una de las palabras del mensaje influye en la forma en que se cifran las siguientes, una propiedad muy interesante y que precisamente utilizan los cifradores modernos, sencillamente definiendo el concepto de palabra como bloque de bits para la cifra y aplicando lo que se denomina cifrado con encadenamiento. [14]

El cifrador de Bazeries

El cifrador de Étienne Bazeries, criptólogo francés nacido a finales del siglo XIX, está basado en el cifrador de ruedas de Jefferson, inventado unos 100 años antes por Thomas Jefferson reconocido como el padre de la criptografía americana. El criptógrafo mostrado en la Figura 1.6 consta de 20 discos, cada uno de ellos con 25 letras en su circunferencia, de forma que la clave se establece sobre la generatriz del cilindro, determinándose 25 alfabetos diferentes. Su funcionamiento es el siguiente: para cifrar el mensaje, primero se divide éste en bloques de 20 letras, procediendo luego a su colocación en forma longitudinal en la línea del visor. El criptograma que se envía puede ser cualquiera de las 25 líneas, también llamadas generatrices del cilindro. Por ejemplo, si se elige la generatriz de distancia +2 en la Figura 1.6, el mensaje M = JE SUIS INDECHIFFRABLE del visor se cifraría como C = LOVS PQUU TPUKEJHHCFDA.

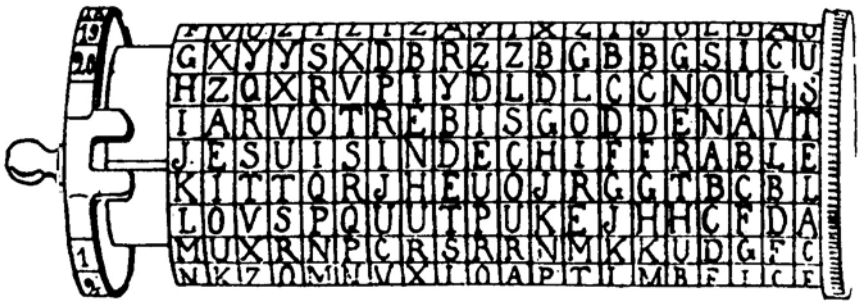

Figura 25: Máquina de Cifrar de Bazeries.

Se puede elegir la misma distancia a la generatriz en la cual se lee el criptograma para todo el bloque o bien cambiar ésta en cada bloque o elemento del bloque, de forma que el número de combinaciones o alfabetos distintos en vez de ser solamente 25 podría crecer hasta el factorial de 25, un valor verdaderamente alto. Uno de estos posibles alfabetos podría ser elegir una secuencia de distancias, una vez introducido el mensaje en el visor, igual a -1,-2,-2,-1,1,2,2,1,-1,-2,-2,-1,1,2,2,1,-1,-2,-2,-1. Es decir, una vez se tenga el mensaje en claro en el visor, se envía como primer carácter del criptograma el que, en la misma columna, está desplazado una posición hacia atrás en el anillo; como segundo el que está desplazado dos posiciones atrás, el tercero también dos posiciones atrás, el cuarto una posición atrás, el quinto una posición hacia delante, el sexto dos adelante, etc., de manera que el criptograma forma una especie de zig-zag en torno al texto en claro, sin transmitir ningún carácter de éste puesto que

la posición 0 no se encuentra en la secuencia indicada. Como es fácil observar, dicha secuencia sería la clave del sistema y, en este caso, su valor máximo sería igual todas las posible permutaciones es decir 25! = 1,55x1025, un valor muy grande aunque el sistema de cifra sería engorroso y poco práctico. [14]

La operación de descifrado consiste en poner los caracteres del criptograma en el visor y buscar en alguna de las líneas el mensaje en claro o seguir el proceso inverso al comentado anteriormente. Como los bloques de criptograma tienen longitud de veinte caracteres, es prácticamente imposible que exista más de una solución con sentido. [14]

Máquinas de cifrar en el siglo XX

La Máquina de Rotor: ENIGMA

En el año 1923, un ingeniero alemán llamado Arthur Scherbius patentó una máquina específicamente diseñada para facilitar las comunicaciones seguras. Se trataba de un instrumento de apariencia simple, parecido a una máquina de escribir. Quien deseara codificar un mensaje sólo tenía que teclearlo y las letras correspondientes al mensaje cifrado se irían iluminando en un panel. El destinatario copiaba dichas letras en su propia máquina y el mensaje original aparecía de nuevo. [13]

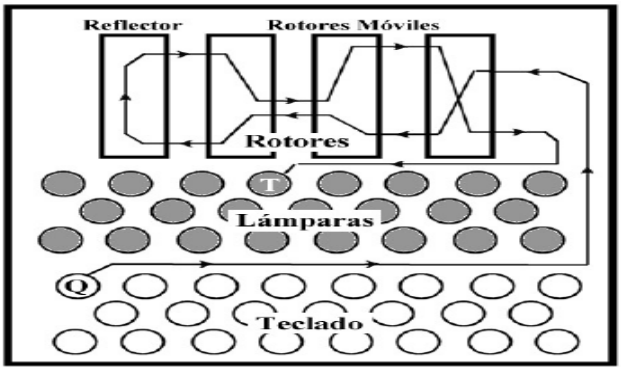

Figura 26: Esquema de la Máquina ENIGMA

La clave la constituían las posiciones iniciales de tres tambores o rotores que el ingenio poseía en su parte frontal. [13]

La máquina Enigma consiste en un banco de rotores montados sobre un eje, en cuyos perímetros había 26 contactos eléctricos, uno por cada letra del alfabeto inglés. En realidad el precursor de este tipo de máquinas con rotores fue Edward Hugh Hebern que algunos años antes inventa y comercializa los denominados cifradores de códigos eléctricos. Los rotores se desplazan como un odómetro; es decir, al cifrar un carácter el primer rotor avanza una posición y sólo cuando éste ha realizado una rotación completa, el segundo se desplaza un carácter, y así sucesivamente. Estos volverán a su posición inicial, tras un período igual a nt. Por ejemplo, en un sistema con 4 rotores, se utilizan de 264 = 456.976 alfabetos. Si aumentamos los rotores a 5, esta cantidad asciende a 11.881.376. La operación de cifra para estas máquinas sigue la siguiente congruencia:

$$Ei(M) = (fi(M - pi) \bmod 26 + pi) \bmod 26$$

En la ecuación anterior, pi es la posición en la que se encuentra el rotor iésimo y fi la correspondencia de los caracteres de la cara anterior y posterior de este rotor. Por lo tanto, el carácter iésimo Mi del mensaje M = m1m2m3... se cifrará como:

$$Eki(Mi) = Ft * ..... * F1(M) \quad [14]$$

El reflector no existía en los primeros modelos, se introdujo posteriormente para permitir que la misma máquina sirviera tanto para cifrar como para descifrar.

ENIGMA pronto llamó la atención del ejército alemán, que la utilizó de forma intensiva a lo largo de la II Guerra Mundial y se le aplicaron varias mejoras. Aunque ENIGMA parecía virtualmente imposible de romper, presentaba una serie de debilidades, tanto en su diseño como en los mecanismos empleados para utilizarla, que fueron aprovechadas por el ejército aliado.

El protocolo empleado por el ejército alemán para colocar los rotores al principio de cada mensaje consistía en escoger una posición de un libro de claves, y enviar tres letras cualesquiera dos veces, para evitar posibles errores. En realidad se estaba introduciendo una redundancia tal en el mensaje que permitía obtener sin demasiados problemas la clave empleada. Se construyó un aparato que permitía descifrar los mensajes, y se le bautizó como Ciclómetro.

En 1938 Alemania cambió el protocolo, lo cual obligó a los matemáticos polacos a refinar su sistema, aunque básicamente se seguían enviando tres letras repetidas.

La fuerza de la máquina ENIGMA radicaba en que tras codificar cada letra se giran los rotores, lo cual hace que la permutación que se aplica a cada letra sea diferente, y que esa permutación además no se repita hasta que los rotores recuperen su posición inicial. Tengamos en cuenta que hay 17576 posiciones iniciales de los rotores, y 60 combinaciones de tres rotores a partir de los cinco de entre los que se puede elegir. La potencia del método de criptoanálisis empleado radica en que bastaba con rastrear dentro del espacio de posibles configuraciones para encontrar aquella que llevara a cabo la transformación esperada. No disponer de dicho emparejamiento hubiera complicado enormemente el criptoanálisis, tal vez hasta el punto de hacerlo fracasar.

Tarjetas de Seguridad [2].

Las tarjetas son un elemento fundamental en el control de acceso a los edificios y permiten la identificación a través de ella, bien con códigos PIN (Personal Identification Number) o/y bien a través de fotografías escaneadas en su superficie. La tarjeta (genérica) es un dispositivo de plástico de dimensiones determinadas, capaz de almacenar y, en algunos casos, procesar información de manera segura. Fue inventado por Roland Moreno (periodista francés) a finales de los 70 y actualmente su complejidad es tal que utilizan tecnología VLSI con microprocesador y sistemas de ficheros cifrado.

Las utilidades que puede tener son almacenamiento y procesamiento de datos confidenciales, estado de las cuentas de crédito, historiales médicos (identificación de pacientes), números de identificación personal, claves privadas (control de acceso), dinero electrónico (micropagos), ...

Las propiedades que tienen las tarjetas son respecto a:

•seguridad, en el almacenamiento de la información (sistemas de fichero) y en las operaciones que realizan (clave simétrica y clave asimétrica)
•portabilidad de información, como historiales, claves, ...
•coste acsequible
•utilizan protocolos propietarios para su programación, como para la comunicación lector-tarjeta, incluso la comunicación entre lectores entre sí, aunque en todos los protocolos existe una función básica para el procesado de revocación que consiste en la gestión de dos listas, lista blanca o de autorizados y lista negra o de personas excluidas (caducados, extraviados, revocados...)

El tipo de tarjetas existentes en el mercado hoy en día son:
Magnéticas (ISO 7811): son muy extendidas, pero son de fácil fraude, con poco espacio de memoria 180 bytes. Son utilizada en accesos e identificación

Con memoria (ISO 7816): con chip integrado que almacena información de forma "segura". Son utilizadas como monederos, ejemplo tarjetas para teléfonos públicos. Se las conoce como Chipcard

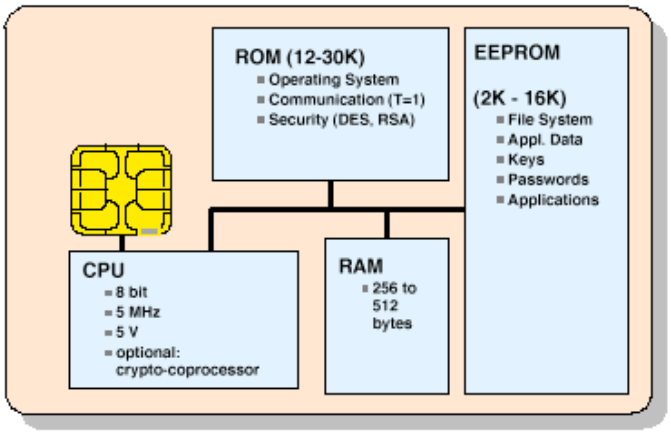

Figura 27: Ejemplo de Tarjeta Inteligente con Microprocesador (dónde la CPU está en la parte de acceso al bus interno de la tarjeta donde conecta la ROM, RAM y EEPROM)

Con microprocesador (ISO 7816/ISO 14443): son tarjetas seguras y pueden ser programadas, con memoria de entre 2-4kbytes. Se consideran multiaplicación, pero tienen el inconvenientes que sólo aceptan aplicaciones desarrolladas por el propio fabricante. Son utilizadas en módulos SIM, comercio electrónico, donde el usuario de identifica con PIN (personal identification number). Son conocidas como SmartCards.

JAVA cards: formada por capas, donde las aplicaciones (applets) y el S.O. Son independientes, utilizan un subconjunto de lenguaje JAVA, Java Card Api 2.1, permiten aplicaciones independiente del hardware escritas en lenguaje de alto nivel, son 10 a 200 veces mas lentas que las anteriores. Este tipo de tarjetas incorpora una máquina virtual JAVA y te permite la programación de Applets.