Criptografía

Andrei Noguera Gil[1]

Universidad Panamericana
Cuidad de México, México
[1]0187940@up.edu.mx

Abstract— **En el siguiente trabajo se describirá la implementación de un chat con implementaciones de seguridad como es el crifado con llave asimétrica y la implementación de un PKI para autenticidad e integridad del mensaje en un desarrollo en llava con un cliente-servidor .**

Keywords— **Seguridad, cifrado asimétrico y simétrico, llaves privadas y públicas, java, cliente-servidor, mensajes, funciones hash.**

I. INTRODUCCIÓN

La criptografía ha estado presente desde hace mucho tiempo por nuestros antepasados, desde un inicio se trataron de ocultar mensajes para evitar que alguien ajeno lo pudiera leer, por ello, en la actualidad es una actividad muy popular y que se ocupa a diario en los sistemas de la mayoría de las instituciones para proteger aquellos datos que son importantes como puede ser contenidos sensibles o críticos.

Existen muchas empresas que se dedican únicamente a proteger la información y desde que empezó el poder del cómputo han realizado varias aplicaciones para lograr ese objetivo, se basan en funciones matemáticas para lograr el mayor grado de complejidad y realmente pueda ser seguro.

Del griego criptos (oculto) y logos (tratado), la criptología se puede definir como la ciencia que estudia las bases teóricas y las implementaciones prácticas para garantizar la privacidad en el intercambio de información.

La criptología puede dividirse en dos partes fundamentales: la criptografía y el criptoanálisis. La criptografía se centra en las técnicas para cifrar la información, mientras que el criptoanálisis se basa en los mecanismos utilizados para descodificar dicha información, es decir, busca romper los procedimientos de cifrado y así conseguir el mensaje original.

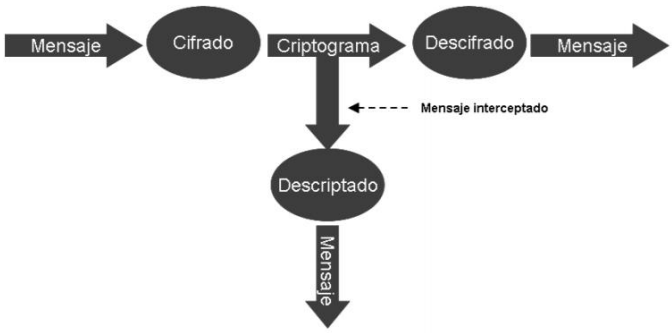

Fig 1. Esquema de concepto de un mensaje cifrado

La criptografía es una técnica muy antigua, y durante mucho tiempo se ha relacionado con los círculos militares, religiosos y comerciales. Actualmente, la necesidad de proteger la información ha hecho que la utilidad de la criptografía se haya extendido a actividades comunes. Otras aplicaciones aparte de la comunicación segura de información es la autenticación de información digital (firma digital).

Las técnicas criptográficas se remontan a la antigüedad, y ya en el año 400 a.C. aparecen las primeras prácticas. A continuación, se enumeran algunas de ellas así como su evolución a lo largo de la historia.

Cifrado César

Remontándose al 100 a.C. el “Cifrado César” nació con la necesidad de ocultar información escrita en latín por parte del ejército de Julio César.

La técnica utilizada para cifrar un mensaje en el “Cifrado Cesar” era sustituir cada una de las letras del mensaje por aquella que ocupaba tres posiciones más en el alfabeto.

Original	a	b	c	d	e	f	g	h	i	j	k	l	m
Cifrado	d	e	f	g	h	i	j	k	l	m	n	o	p

Original	n	o	p	q	r	s	t	u	v	w	x	y	z
Cifrado	q	r	s	t	u	v	w	x	y	z	a	b	c

Fig 3. Cifrado César

La máquina enigma

En 1918 el inventor alemán Arthur Scherbius y Richard Ritter querían sustituir los inadecuados sistemas de criptografía empleados en la Primera Guerra Mundial recurriendo a la tecnología en vez de utilizar lápiz y papel. De esta manera desarrollaron la máquina Enigma.

La máquina Enigma en sus inicios se componía de tres partes fundamentales: un teclado en el cual se escribía el texto que se quería cifrar, una unidad modificadora y un tablero en el que se mostraba el mensaje codificado.

En la Figura 5 se visualiza el proceso de cifrado de la máquina Enigma con un ejemplo en el que se han utilizado 7 letras. Cada letra del texto a cifrar se pulsaba en el teclado y la unidad modificadora lo transformaba en otra diferente antes de mostrarla en el tablero. Así, la letra E se intercambiaba por la letra B.

Arthur Scherbius la creó de manera que cada vez que se pulsara una letra la unidad modificadora girase un veintiseisavo de vuelta (para un alfabeto completo de 26 letras), con lo que la codificación de la siguiente letra sería diferente a la de la primera.

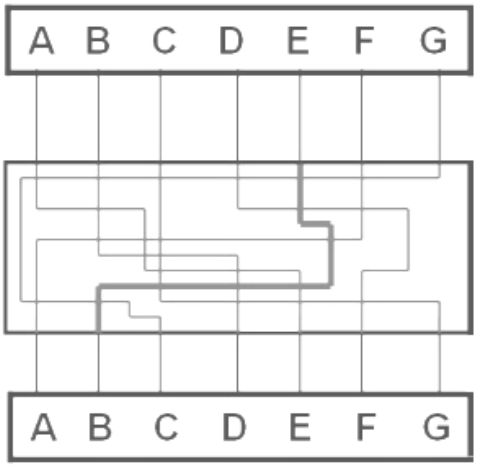

Fig 5. Cifrado de la máquina Enigma con un disco modificador (con 7 letras)

El principal problema que presentaba era que de esta forma se disponía de un cifrado de Vigenére de 26 letras el cual era fácil de descifrar. Para solventarlo Scherbius colocó una segunda unidad modificadora, de tal forma que cuando el primer disco daba una vuelta completa, el segundo giraba una posición.

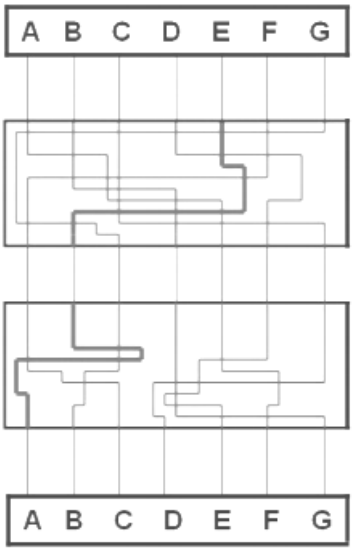

Fig 6. Cifrado de la máquina Enigma con dos discos modificador (con 7 letras)

Insertando este nuevo disco modificador se consigue pasar de una clave de 26 posiciones a una de 26*26, es decir, de 676. Mas tarde, aún se añadió otro disco más, que avanzaba una posición cuando el segundo había dado una vuelta completa consiguiendo así 26*26*26= 17.576 posiciones.

La máquina Enigma disponía de otro complemento más, el Reflector, mediante el cual, los receptores del mensaje podían descifrarlo. Cuando éste llegaba al destino los receptores estaban preparados con otra máquina Enigma con los rotores colocados en la misma posición que la máquina que enviaba el mensaje cifrado y a través del Reflector introduciendo el mensaje cifrado se reproducía el mensaje original.

Finalmente, se le añadieron dos características más:

- Los rotores eran intercambiables, de manera que como hay 6 modos de poner los tres rotores, el número de claves aumenta
- Se introdujo un clavijero con el que se podía intercambiar pares de letras en grupos de 6.

De esta manera la máquina Enigma ofrecía billones de posibilidades de creación de claves.

El ejército alemán aseguraba que era el sistema de cifrado más seguro del mundo para sus comunicaciones, y adquirieron una gran cantidad de máquinas Enigma durante la Segunda Guerra Mundial.

Los alemanes utilizaban una nueva clave cada día, para que los enemigos dispusieran de sólo 24 horas para poder descifrarla con lo que supuso el método criptográfico más seguro hasta el momento.

Fig 7. Máquina enigma

Cifrado simétrico

La criptografía simétrica o de llave secreta es aquella que utiliza algún método matemático llamado sistema de cifrado para cifrar y descifrar un mensaje utilizando únicamente una llave secreta. Se puede observar en la siguiente figura que la línea punteada es el eje de simetría: lo mismo que hay de un lado existe exactamente igual en el otro, esto ilustra el hecho del porqué se le da el nombre de criptografía simétrica.

Fig 8. Cifrado con llave simétrica

Este tipo de criptografía sólo utiliza una llave para cifrar y descifrar, esto es: si yo cifro un mensaje m con una llave secreta k entonces el mensaje cifrado resultante m' únicamente lo voy a poder descifrar con la misma llave k. Este tipo de llave conocida como secreta se debe de compartir entre las personas que se desea que vean los mensajes.

Con este tipo de criptografía podemos garantizar la confidencialidad porque únicamente quien posea la llave secreta será capaz de ver el mensaje. El problema con la criptografía simétrica es que si yo quisiera compartir secretos con m personas, para cada persona tendría que generar una nueva llave secreta y la administración personal de todas m llaves sería un caos. Otro problema asociado con este tipo de criptografía es cómo comparto con otra persona de una forma confidencial e integra la llave secreta.

Estos problemas se resuelven de cierta manera con criptografía asimétrica

Cifrado Asimétrico

Si se observa la siguiente figura, que ilustra la idea de criptografía de llave pública, se puede ver claramente que no existe simetría en ella, ya que de un lado de la figura se cifra o descifra con una llave pública y en el otro lado con una privada. De este hecho es de donde la criptografía asimétrica debe su nombre.

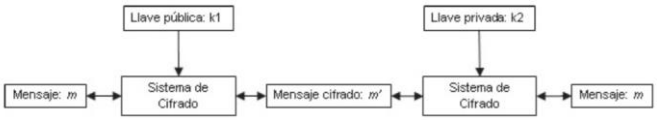

Fig 9. Cifrado con llave asimétrica

Es importante destacar que para este tipo de criptografía lo que se cifra con una llave se puede descifrar con la otra llave. Es decir, yo puedo cifrar con la llave pública y descifrar con la privada y viceversa. Esto es de gran ayuda ya que el número de llaves que debo de poseer se reduce considerablemente. Si alguien quisiera enviar un mensaje cifrado a n personas, necesitaría saber n llaves públicas una de cada persona, pero si n personas le quiere enviar un mensaje cifrado sólo es necesario que los demás conozcan su llave pública. Así, sólo tengo que preocuparme de que la llave pública sea de la persona que dice ser. Este es el problema de la criptografía asimétrica, la autenticidad de las llaves públicas.

Algunos ejemplos de este tipo de criptografía son RSA, El Gamal y Curvas Elípticas.

Solución al problema de intercambio de llaves secretas usando criptografía asimétrica: se supone que alguien va a enviar la llave secreta k a una persona para que puedan cifrar entre ellos mensajes. Lo que se hace es que se toma la llave

pública de la persona a la que se le va a enviar el mensaje y se cifra con un sistema asimétrico la llave secreta, esto implica que sólo la persona poseedora de la llave privada pueda descifrar lo que se está enviando y con ello tener la llave secreta.

Ejemplos de tipos de cifrado:

DES:

El algoritmo DES (Data Encryption Standard) es un algoritmo de cifrado desarrollado por la NSA a petición del gobierno de EEUU bajo la presión de las empresas por la necesidad de un método para proteger sus comunicaciones. DES fue escogido como FIPS (Federal Information Processing Standard) en el año 1976 y su uso se extendió por todo el mundo. Hoy en día DES es considerado inseguro dada su clave de 56 bits, insuficiente frente al poder computacional actual. En su variante Triple DES el algoritmo se cree seguro.

DES es un algoritmo de cifrado por bloques. Se toma un bloque de una longitud fija de bits y lo transforma mediante una serie de operaciones básicas en otro bloque cifrado de la misma longitud. En el caso de DES el tamaño del bloque es de 64 bits. La clave también tiene 64 bits pero 8 de estos bits se emplean para comprobar la paridad, haciendo que la longitud efectiva de la clave sea de 56 bits.

DES se compone de 16 fases o rondas idénticas. Al comienzo y al final se realiza una permutación. Estas permutaciones no son significativas a nivel criptográfico, pues se incluyeron para facilitar la carga y descarga del bloque en el hardware de los años 70. Antes de cada ronda el bloque se divide en dos mitades de 32 bits y se procesan alternativamente. Este proceso es conocido como esquema Feistel. El esquema Feistel nos proporciona un proceso de cifrado y descifrado casi iguales. La única diferencia es que las subclaves se aplican de forma inversa al descifrar.

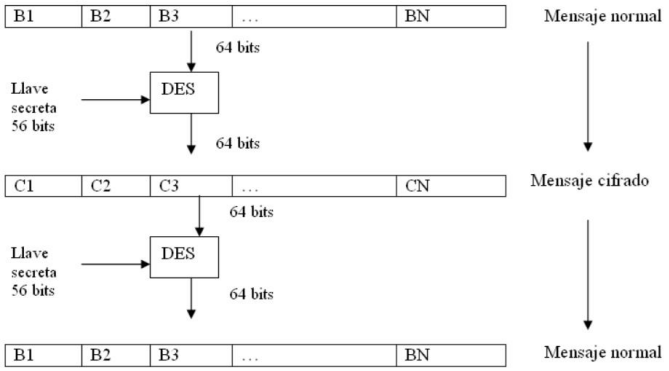

Fig 9. DES

AES:

El algoritmo AES (Advanced Encryption Standard) tambien conocido como Rijndael fue el ganador del concurso convocado en el año 1997 por el NIST (Intituto Nacional de Normas y Tecnología) con objetivo de escoger un nuevo algoritmo de cifrado. En 2001 fue tomado como FIPS y en 2002 se transformó en un estándar efectivo. Desde el año 2006 es el algoritmo más popular empleado en criptografía simétrica.

AES opera sobre una matriz de 4x4 bytes. Mediante un algoritmo se reordenan los distintos bytes de la matriz. El cifrado es de clave simétrica, por lo que la misma clave aplicada en el cifrado se aplica para el descifrado.

Basado en El algoritmo Rijndael, Al contrario que su predecesor DES, Rijndael es una red de sustituciónpermutación, no una red de Feistel. AES es rápido tanto en software como en hardware, es relativamente fácil de implementar, y requiere poca memoria.

El algoritmo AES funciona mediante una serie de bucles que se repiten. 10 ciclos para claves de 128 bits, 12 para 192 y 14 para 256.

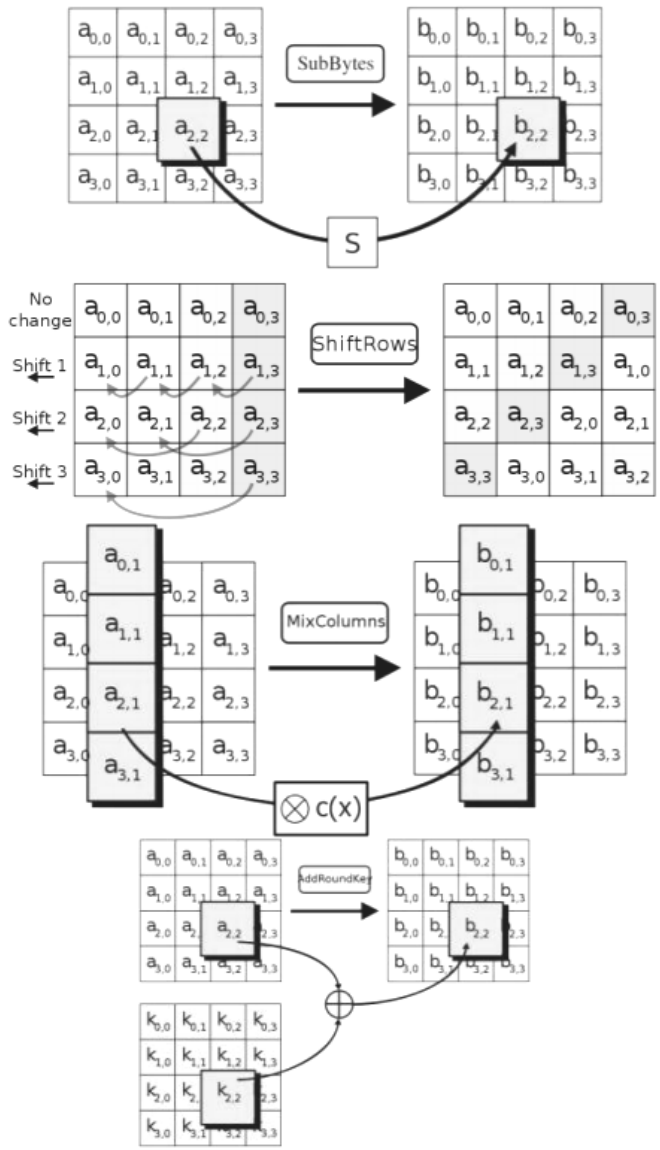

Fig 10. AES fases

Documento digital

En criptografía existen deferentes documentos digitales que se usan para garantizar las propiedades de confidencialidad e integridad, estos documentos son la integración de los dos tipos de criptografía: la simétrica y la asimétrica. Al hacer esta integración se compensan las desventajas de los tipos de cifrado y se utilizan las mejores características de cada uno, combinando rapidez del cifrado simétrico con la facilidad de la administración de llaves del cifrado asimétrico.

Firma Digital

Una firma digital es un documento que permite garantizar la integridad de un documento y se puede relacionar de manera única al firmante con su firma, ya que realiza ésta con la llave privada y únicamente el firmante posee esa llave, ésto se traduce en que se verifica la autenticidad del firmante.

Antes de entrar más en detalle de cómo se realizan las firmas digitales, es importante hablar de una función denominada “Hash” o resumen del documento. Esta función lo que hace es que a partir de un documento de tamaño N bits entrega una cadena de M bits. No hay límite para el tamaño de N, pero M siempre es de tamaño constante de acuerdo con el algoritmo usado, normalmente es de 128 o 256 bits.

Una de las características de este tipo de funciones es que son unidireccionales, es decir, que debe de ser imposible a partir del resumen del documento encontrar el mensaje original. También deben cumplir la propiedad de dispersión, lo que significa que si se cambia al menos un bit del documento, su resumen debe de cambiar la mitad de sus bits aproximadamente.

La firma de un documento d se realiza tomando un documento digital, se extrae el resumen del documento H(d) y este resumen se cifra asimétricamente con la llave privada del firmante Ck1(H(d)), esto es lo que vendría siendo la firma digital, ahora hay que ponérsela al documento, para eso se concatenan el documento y su resumen cifrado.

Ahora hay que verificar la firma, para eso se separan el documento d del resumen cirado. Se descifra asimétricamente con la llave pública k2 del firmante el resumen cifrado Dk2(Ck1(H(d))) obteniéndose el resumen del documento original H(d). Se obtiene el resumen del documento enviado H(d)’ se comparan las dos digestiones H(d)= H(d)’ y si estos son iguales, se dice que la firma es válida, de lo contrario es inválida. Si la firma es inválida puede deberse a dos causas: una es que se está usando una llave pública que no corresponde con la privada del firmante (problema de autenticación) o la otra es que el documento que se envió fue alterado (problema de integridad). La siguiente figura ilustra el proceso descrito de firmar y validar la firma digital

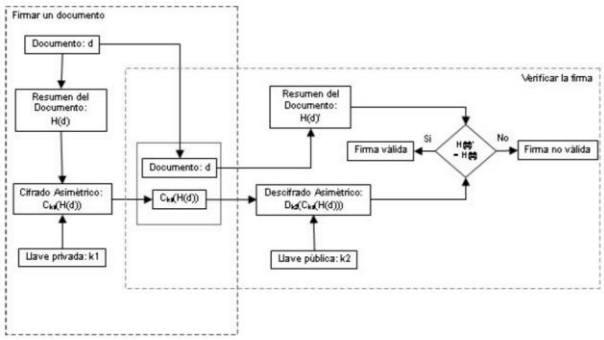

Fig 11. Firma Digital

Sobre Digital

Con un sobre digital se pueden garantizar las propiedades de confidencialidad de un documento. El sobre digital usa criptografía simétrica y asimétrica. Un sobre digital se genera a partir de un documento d y una llave secreta k que se genera de forma aleatoria, se cifra simétricamente Ck(d) el documento d con la llave secreta k, luego la llave secreta k se cifra asimétricamente con la llave pública k2 de la persona a la que le vamos a enviar el sobre Ck2(k) y finalmente se concatenan el cifrado del documento Ck(d) con el cifrado de la llave secreta Ck2(k) dando origen al sobre digital.

Para abrir el sobre digital se toma el cifrado de la llave secreta Ck2(k) y se descifra Dk1(Ck2(k)) con la llave privada k1 de la persona a la que va dirigida el sobre, obteniendo la llave secreta k. Con la llave k se descifra el cifrado del documento Dk(Ck(d)) obteniendo así el documento d original. Esto se puede ver gráficamente en la siguiente figura.

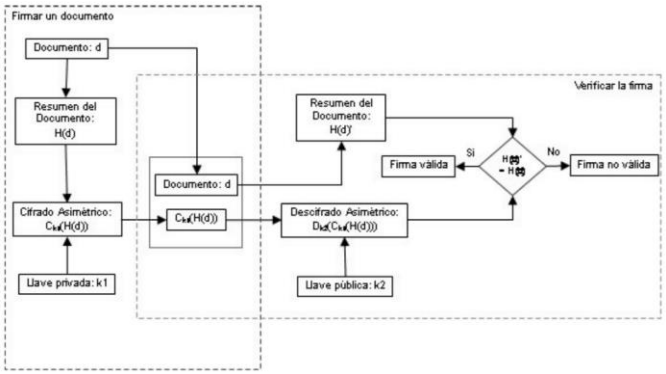

Fig 12. Sobre digital

Ahora bien, se pueden combinar los sobres digitales con las firmas digitales dando lugar a un sobre firmado y así se garantizan las propiedades de integridad, confidencialidad y autenticación.

Certificados digitales:

Un certificado digital básicamente es un documento digital expedido por una autoridad de confianza que contiene los datos que identifican al dueño del certificado, su llave pública, fecha de expedición, fecha de caducidad, los datos de la autoridad de confianza y finalmente todo esto está firmado por la misma autoridad.

Los certificados sirven para establecer lazos de confianza entre sistemas o personas, ya que si se confía en la autoridad de confianza entonces se puede confiar en la llave pública del dueño del certificado. Tratando así de resolver el problema de relacionar las identidades con las llaves públicas.

Como podemos observar, la criptografía no es la panacea, pero bien usada puede ser de gran ayuda para mantener la seguridad informática.

Función Hash

El término hash proviene, aparentemente, de la analogía con el significado estándar (en inglés) de dicha palabra en el mundo real: picar y mezclar. Donald Knuth cree que H. P. Luhn, empleado de IBM, fue el primero en utilizar el concepto en un memorándum fechado en enero de 1953. Su utilización masiva no fue hasta después de 10 años.

A las funciones resumen también se les llama funciones hash o funciones digest. Una función hash H es una función computable mediante un algoritmo tal que:

$H : U \Rightarrow M$

$X \Rightarrow h(x)$

Tiene como entrada un conjunto de elementos, que suelen ser cadenas, y los convierte en un rango de salida finito, normalmente cadenas de longitud fija. Es decir, la función actúa como una proyección del conjunto U sobre el conjunto M.

Observar que M puede ser un conjunto definido de enteros. En este caso podemos considerar que la longitud es fija si el conjunto es un rango de números de enteros ya que podemos considerar que la longitud fija es la del número con mayor número de cifras. Todos los números se pueden convertir al número especificado de cifras simplemente anteponiendo ceros.

Normalmente el conjunto U tiene un número elevado de elementos y M es un conjunto de cadenas con un número más o menos pequeño de símbolos. La idea básica de un valor hash es que sirva como una representación compacta de la cadena de entrada.

La definición formal dada, a veces se generaliza para poder aprovechar las funciones hash en otros ámbitos. Para ello a la función resumen se le añaden nuevos parámetros de forma que el valor hash no es solo función del contenido en sí, sino además de otros nuevos factores.

Para hallar valores resumen de ficheros a veces se usan, además del contenido en sí, diversos parámetros como el nombre del archivo, su longitud, hora de creación, etc.

Otras veces se añaden parámetros que permiten configurar el comportamiento de la función. Por ejemplo, la función resumen puede recibir como parámetro una función de generación de valores pseudoaleatorios que es usada dentro del algoritmo de la función hash.

Otros ejemplos de parámetros son el uso de valores sal, el uso de claves secretas, el uso de parámetros que especifican el rango de la función (funciones hash de rango variable), el uso de parámetros que especifican el nivel de seguridad que se quiere en el valor resumen de salida (funciones hash dinámicas), etc.

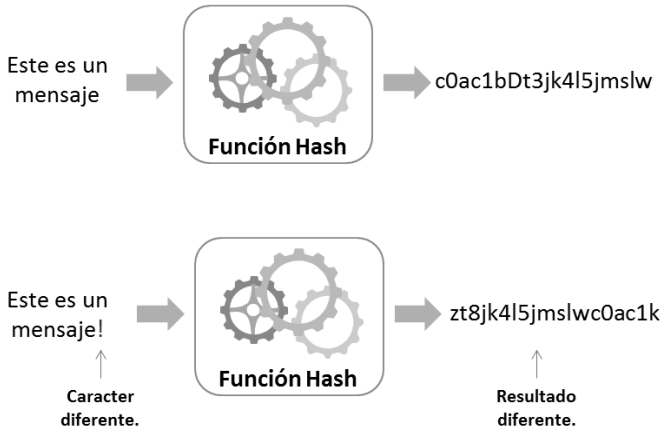

Fig 13. Función hash

II. Implementación

El proyecto consiste en la creación de un programa de mensajería con implementaciones de seguridad y criptografía, el usuario debe de ser capaz de agregar contactos, mandar mensajes con la opción de elegir el nivel de seguridad que desea aplicar (texto plano, simétrico, asimétrico, firma digital o documento ensobretado) e iniciar sesión por medio de un protocolo y factores de autenticación. El proyecto se desarrolló en Java y todas las funciones se crearon desde cero sin utilizar otros tipos de librerías. Se compone de un programa para el cliente para la mensajería, un servidor encargado de administrar los mensajes entre los clientes y un servidor de PKI como administrador de certificados que contienen las llaves públicas de cada usuario.

Los requisitos del proyecto son los siguientes:

- Utilizar distintos métodos de cifrado para el envío de mensajes
- Realizar una función hash propia
- Generar llaves simétricas o asimétricas basados en el algoritmo de césar
- Un servidor centralizado
- Bases de datos embebidas
- ARC y AR para generar un PKI
- Utilizar dos factores de autenticación
- Implementar un protocolo de autenticación

Aplicaciones:

Cliente o chat

El cliente está constituido por sockets e hilos para conectarse al servidor central, un gestor de mensajes que tiene como función mandar mensajes y leer los mensajes. Para cada mensaje se utilizaron objetos y cada objeto de mensajes tiene los siguientes componentes:

- Content Code: Describe el tipo de servicio a solicitar, 0 es para servicio entre clientes, 1 es para servicios a servidores centrales y 2 es para servicos de PKI
- Type Sender: Este campo es para especificar el tipo de usuario quien lo mandó o tipo de interfaz, el 0 es para el servidor, 1 es para el cliente y el 2 es del PKI
- Origin: Es el número de la intefaz de donde se envía el mensaje, puede ser un número o el nombre de la interfaz
- Destination: Es el número de la intefaz a donde debe de llegar el mensaje, puede ser un número o nombre de la interfaz
- Operación: Define el tipo de solicitud que solicita, para cada tipo de solicitud es diferente la implementación del método

- Value: Es el contenido del mensaje, puede ser un objeto como un sobre digital o firma digital o un mensaje en texto plano
- Event ID: Es el nombre del evento o es un ID único al evento que se genera por el mensaje

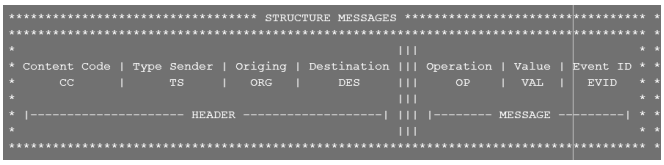

Fig 14. Contenido del mensaje

Las características que tiene la aplicación del cliente:

- Agregar contactos
- Introducir las llaves públicas y privadas
- Mandar mensajes
- Leer mensajes
- Obtener los mensajes de la base de datos
- Corroborar la firma digital
- Cambiar el tipo de cifrado
- Iniciar Sesión
- Generar un usuario

Para el registro de usuarios en la plataforma se realizó un programa secundario que su función principal es realizar una conexión al servidor central y al PKI para agregarlo a la base de datos para poder utilizar sus servicios posteriormente.

Los datos que se necesitan son:

- Nombre
- Contraseña
- Número de registro
- Frase de seguridad

Con los datos anteriores se genera el registro de usuario y contraseña en el servidor, la llave pública y privada para la generación del certificado que se registra en el PKI además de encriptar la llave con una frase privada para garantizar su seguridad. Todos los usuarios tienen como identificador único el número requerido y en caso de poner uno duplicado los servidores les generan un mensaje para que cambie el número.

Los certificados contienen:

- El número de certificado o identificador
- El nombre del usuario
- La llave pública
- Fecha de creación
- Fecha de vigencia

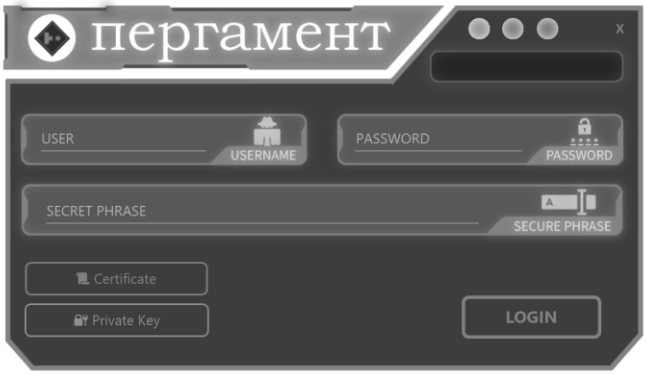

Fig 15. Inicio de sesión y registro

Fig 16. Aplicación del chat

Servidor:

El servidor es el encargado de la gestión de los mensajes, guarda las conexiones de los clientes conectados, las conexiones se realizan mediante sockets, para el intercambio de mensajes. Guarda todos los mensajes mandados entre los clientes en una base de datos. El servidor contiene el código 0 como identificador y un certificado público junto con una llave privada.

Las actividades que realiza es las siguientes:

- Recibir mensajes
- Mandar los mensajes únicamente al receptor
- Guarda los mensajes a la base de datos

PKI:

El PKI es el administrador de llaves públicas y de certificados, es el encargado de guardar ambos componentes y atender solicitudes.

Las actividades que realiza son las siguientes:

- Ingresar nuevas llaves
- Actualizar llaves
- Ingresar certificados
- Actualizar certificados
- Peticiones de llaves públicas
- Peticiones de certificados

Implementaciones de cifrados y protocolos de autenticación

Existen 5 tipos de mensajes de los cuales 4 implementan algún tipo cifrado:

- Texto plano: Mensaje sin ningún tipo de implementación de seguridad
- Simétrico: Utiliza la llave pública del emisor para cifrar el mensaje y el receptor realiza una petición al PKI para descifrar el mensaje con la llave pública del emisor
- Asimétrico: Utiliza la llave privada del emisor para cifrar el mensaje y el receptor realiza una petición al PKI para descifrar el mensaje con la llave pública del emisor
- Firma digital: El mensaje se pasa por una función Hash y después se cifra con la llave privada del emisor junto con el mensaje original, para descifrarlo se debe de realizar el proceso inverso con la llave pública del emisor
- Sobre digital: Se crea una firma digital y se cifra con la llave aleatoria creada en ese momento junto con la llave de aleatoria cifrada con la llave pública del emisor

Protocolo de autenticación PFS:

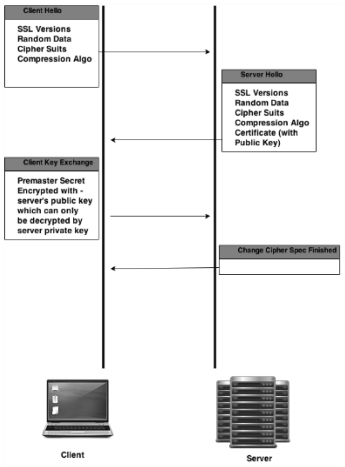

Fig 16. Aplicación de PFS

III. Conclusiones

Realizar un sistema seguro en un chat requiere de muchos componentes para poderlo desarrollar, se necesita un amplio conocimiento en java, para ello también requiere que tengas un amplio conocimiento en los conceptos de criptografía y la administración de las llaves.

References

[1] https://profecd.webnode.es/_files/200000079-90fc291f71/Introduccion%20a%20la%20criptografia.pdf. [online] Título: Introducción a al criptografía. Autores: Gibrán Granados Paredes

[2] http://www3.uah.es/libretics/concurso2014/files2014/Trabajos/Criptografia%20y%20Metodos%20de%20Cifrado.pdf [online] Título: Criptografía y Métodos de Cifrado. Autores: Héctor Corrales Sánchez, Carlos Cilleruelo Rodríguez, Alejandro Cuevas Notario

[3] https://www.researchgate.net/publication/228173118_La_Criptografia_y_la_Proteccion_a_la_Informacion_Digital [online] Título: la criptografía y la protección a la información digital. Autores: Jhonny Antonio Pabón

[4] https://www.matem.unam.mx/~rajsbaum/cursos/web/presentacion_seguridad_1.pdf [online] Título: Criptografía.

[5] http://www.egov.ufsc.br/portal/sites/default/files/la_criptografia_desde_la_antigua_grecia_hasta_la_maquina_enigma1.pdf [online] Título: LA CRIPTOGRAFÍA DESDE LA ANTIGUA GRECIA HASTA LA MÁQUINA ENIGMA. Autores: Instituto Nacional de Tecnologías de la Comunicación