



INSTITUTO POLITÉCNICO NACIONAL
Escuela Superior de Cómputo

ESCOM

Trabajo Terminal

**“Protocolo criptográfico para el almacenamiento
sin duplicados en la nube, resistente a ataques
por fuerza bruta.”**

2016-B045

Presentan

Eder Jonathan Aguirre Cruz
Diana Leslie González Olivier
Jhonatan Saulés Cortés

Directora

Dra. Sandra Díaz Santiago

INSTITUTO POLITÉCNICO NACIONAL



ESCOM

Mayo 2017

Índice

1. Introducción	1
1.1. Justificación	3
1.2. Objetivos	3
1.2.1. Objetivos Generales	3
1.2.2. Objetivos Específicos	3
2. Preliminares.	4
2.1. Definiciones.	4
2.1.1. Servicios criptográficos.	4
2.2. Ataques a servicios criptográficos.	5
2.3. Criptografía Simétrica.	5
2.4. Criptografía asimétrica.	6
2.5. Cifrado por bloques.	7
2.6. RSA.	8
2.7. Firmas a ciegas.	9
2.8. Funciones Hash.	9
2.9. Cómputo Nube.	10
3. Análisis y Diseño	14
3.1. Business Process Model and Notation (BPMN)	14
Bibliografía	15

Índice de Figuras

1.1. Diagrama de criptografía simétrica.	2
1.2. Diagrama de criptografía asimétrica.	2
2.1. Diagrama de Criptografía Simétrica.	6
2.2. Diagrama de Criptografía Asimétrica.	7
2.3. Diagrama de Cifradores por Bloques	8
3.1. BPMN Subir archivo.	14
3.2. BPMN Descargar archivo.	14

Índice de Tablas

Capítulo 1

Introducción

Hoy en día el manejo de información en la sociedad juega un papel importante en el desarrollo de las actividades que la conforman. Millones de personas en el mundo tienen la facilidad de acceder a un dispositivo electrónico que les permite manipular esta información o almacenarla para posteriormente darle un uso específico. La información que circula en dispositivos electrónicos es mayor a la memoria disponible que ofrecen estos, a medida que el volumen de información aumenta, también lo hace la demanda para los servicios de almacenamiento en línea. Un gran incremento en el uso de estos servicios implica tener más infraestructura y personal para que los sistemas de almacenamiento tengan más capacidad y puedan cubrir la demanda que se presenta en el mercado. Si bien el almacenamiento logró dar buenos resultados al cliente en sus primeras etapas, ahora la preocupación por el incremento de infraestructura para seguir dando esos resultados se ha incrementado considerablemente. [2] [1]

Una de las principales razones por la que está sucediendo es que muchos usuarios almacenan un mismo archivo un claro ejemplo es que n usuarios pueden subir la misma canción, por lo tanto esta se encuentra almacenada en las n cuentas de la nube, esto implica un gasto innecesario de almacenamiento. Según un estudio realizado por HP se estima que hay 1 Exabyte de datos almacenados en la nube, además de 2012 a 2017, las cargas de trabajo de los centros de datos crecerán 2,3 veces, mientras que en la nube aumentarán 3,7 veces, lo cual implica que el Exabyte que se estima se podría llegar a triplicar y las empresas que proporcionan estos servicios disminuyen su oferta en el mercado. [8]

Un problema agregado a la situación de servicios externos para almacenar datos, es la falta de protección estos, ya que al utilizar un servicio de almacenamiento, los usuarios están haciendo uso de estos servicios los cuales no están protegidos bajo ningún esquema de seguridad y por tanto los usuarios no cuentan con una garantía que le da la completa integridad a la información almacenada. Un claro ejemplo para comprobar esta problemática es que cualquier individuo que tenga acceso a los archivos almacenados, podrá visualizar el contenido de estos, que en ocasiones son utilizados para fines lucrativos perjudicando la integridad de la información del usuario.

Hoy en día existen soluciones para cada uno de los problemas, una de ellas consiste en encontrar y eliminar la duplicación dentro de los datos sin comprometer su integridad o la fidelidad. El objetivo es almacenar más datos en menos espacio mediante la segmentación de los archivos en pequeños trozos de tamaño variable (32 a 128 KB), la identificación de fragmentos duplicados, manteniendo una sola copia de cada trozo. Las copias redundantes del trozo se sustituyen por una referencia a la única copia. Los trozos se comprimen y luego son

organizados en contenedores especiales de archivos en la carpeta Información del volumen del sistema. Para garantizar la privacidad de los datos obtenidos después del proceso de eliminación de duplicación, es posible utilizar algoritmos criptográficos. [10]

Una posible solución para proteger a los datos es echar mano de la criptografía, ciencia que se encarga del estudio de técnicas para transformar la información a una forma que no pueda entenderse a simple vista; sin embargo, el objetivo de la Criptografía no es sólo mantener los datos secretos, sino también protegerlos contra modificación y comprobar la fuente de los mismos. [9] La criptografía esta dividida en dos grandes tipos que son:

- La criptografía simétrica: Utiliza una misma llave para realizar el proceso de cifrado y descifrado, ver figura 1.1.

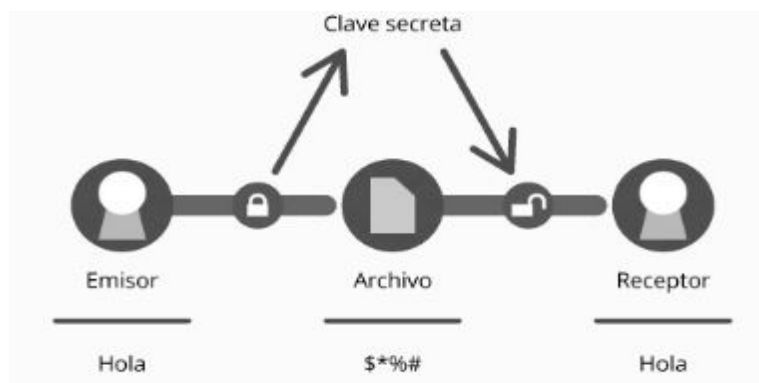


Figura 1.1: Diagrama de criptografía simétrica.

- La criptografía asimétrica: Utiliza una clave para el cifrado y otra para el descifrado, ver figura 1.2. [4]

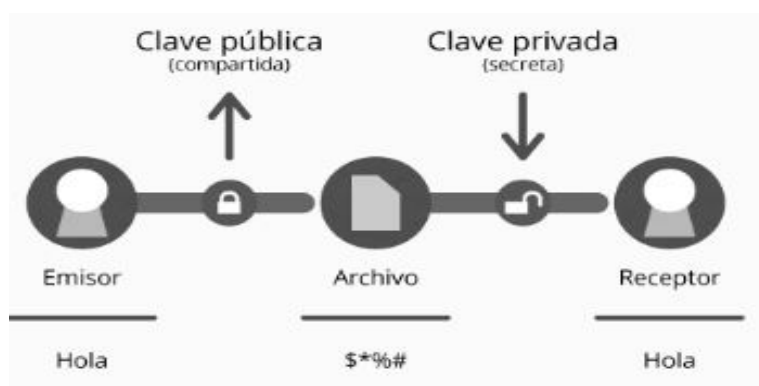


Figura 1.2: Diagrama de criptografía asimétrica.

Puesto que ambas cuestiones, la eliminación de duplicados y la privacidad de la información, son importantes, se ha comenzado a proponer mecanismos que solucionen ambos problemas de manera conjunta, que son: Dupless [2], ABS: the apportioned backup system. [4], Flud Backup [13], SIGOPS Oper. Syst. [5], TahoeFS [17].

1.1. Justificación

En la actualidad millones de personas usan los servicios de almacenamiento que ofrece la nube, ya sean gratuitos o privados, este número de personas ha ido en un incremento exponencial lo cual hace que el espacio de almacenamiento disminuya, entonces ¿Cómo podría mitigar el problema de almacenamiento y tener privacidad de los datos al mismo tiempo?

Usando la criptografía clásica para poder cifrar un archivo se utiliza una clave privada la cuál es distinta para cada usuario, cada vez que se cifra un archivo el resultado de este es diferente para cada intento. Por tanto no se puede evitar la duplicación de archivos utilizando este mecanismo de la criptografía y se deben implementar soluciones más robustas.

Una solución para tener privacidad y evitar duplicación la proporcionó John R. Douceur, la cual dice que teniendo a M que será el contenido de un archivo de aquí en adelante denominado el mensaje, el cliente primero calcula una clave $K \leftarrow H(M)$ mediante la aplicación de una función de hash criptográfica H al mensaje y luego calcula el texto cifrado $C \leftarrow E(K, M)$ a través de un esquema de cifrado simétrico determinista. El derivado del mensaje K se almacena por separado cifrándolo con una llave por cliente. Un segundo cliente B cifra el mismo archivo M que producirá el mismo C , evitando la duplicación. [6]

En el artículo publicado por Mihir Bellare, Sriram Keelveedhi, Thomas Ristenpart, nombrado “DupLESS: Server-Aided Encryption for Deduplicated Storage” [2], se observó que uno de los principales problemas al que nos enfrentamos es que el esquema de cifrado solo es seguro cuando el espacio de mensajes es demasiado grande, por lo tanto agentes externos pueden provocar agravios a la integridad de la información de los usuarios.

Si bien esta solución se ocupa de la duplicación de archivos deja muy vulnerable el aspecto de la privacidad, ya que ante un espacio de mensajes pequeño las amenazas del adversario son demasiadas. Si se tuvieran como ejemplo 1000 mensajes, para el adversario sería muy fácil intentar encontrar la clave, probando las 1000 claves posibles generadas con la función hash, hasta descifrar el archivo, por lo tanto se comprueba que un espacio de 1000 mensajes sigue siendo pequeño.

Es por ello que este trabajo terminal tiene como principal meta atacar esta problemática de privacidad, proponiendo una arquitectura del sistema que a través de un servidor de llaves se generaran llaves de acuerdo al contenido del archivo, para con esta se pueda cifrar y luego almacenar en la nube donde se eludirá la duplicación de archivos. Dicha arquitectura se explica con detalle en el siguiente apartado.

1.2. Objetivos

Diseñar un sistema que evite la duplicación de archivos, garantizando la privacidad de los usuarios contra adversarios cuando el espacio de mensajes es pequeño, utilizando algoritmos criptográficos para su implementación.

1.2.1. Objetivos Generales

1.2.2. Objetivos Específicos

Capítulo 2

Preliminares.

2.1. Definiciones.

Criptografía. La Criptografía es la ciencia que se encarga del estudio de técnicas para transformar la información a una forma que no pueda entenderse a simple vista; sin embargo, el objetivo de la Criptografía no es sólo mantener los datos secretos, sino también protegerlos contra modificación y comprobar la fuente de los mismos.

Criptoanálisis. Es la ciencia que se ocupa del análisis de un texto cifrado para obtener la información original sin conocimiento de la clave secreta, esto es, de forma ilícita rompiendo así los procedimientos de cifrado establecidos por la Criptografía, por lo que se dice que Criptoanálisis y Criptografía son ciencias complementarias pero contrarias. El criptoanálisis es el arte de descifrar comunicaciones cifradas sin conocer las llaves. [18]

2.1.1. Servicios criptográficos.

Los servicios de seguridad, son aquellos que garantizan en un sistema de información la adquisición, almacenamiento, procesamiento y transmisión de la información y para lograrlo se valen de uno o más mecanismos de seguridad.

Confidencialidad. Este servicio asegura que sólo las personas o procesos autorizados tengan acceso a la información. Con ello se busca que un agente no autorizado no pueda leer, copiar o modificar la información.

Autenticación. Este servicio verifica la identidad de un agente que pretende acceder a la información. En una conexión entre dos entidades, el servicio verifica que las entidades sean quienes dicen ser, además de asegurar que un tercer individuo no pueda hacerse pasar por alguna de las entidades autorizadas y realizar una transmisión o recepción de datos.

Integridad. Este servicio asegura que el contenido de los datos no ha sido modificado y que la secuencia de los mismos se ha mantenido a lo largo de toda la transmisión, con ello se evita una réplica o un reordenamiento del mensaje por parte de un atacante. Al igual que el servicio de confidencialidad, la integridad puede aplicarse a todos los datos transmitidos por un usuario durante una conexión, sólo a parte de ellos o sólo a algunos campos dentro del mensaje. [19]

No repudio. El no repudio sirve a los emisores o a los receptores para negar un mensaje transmitido. Por lo que cuando un mensaje es enviado, el receptor puede probar que el mensaje fue enviado por el presunto emisor. De manera similar, cuando un mensaje es recibido, el remitente puede probar que el mensaje fue recibido por el presunto receptor. [7]

2.2. Ataques a servicios criptográficos.

Un ataque es una violación a la seguridad de la información realizada por intrusos que tienen acceso físico al sistema sin ningún tipo de restricción, su objetivo es robar la información o hacer que ésta pierda valor relativo, o que disminuyan las posibilidades de su supervivencia a largo plazo.

Ataque sólo con texto cifrado. Este caso es cuando el criptoanalista sólo conoce el criptograma y el algoritmo con que fue generado; con esta información pretende obtener el texto en claro.

Ataque con texto original conocido. En esta situación el criptoanalista conoce mensajes en claro seleccionados por él mismo y sus correspondientes criptogramas, así como el algoritmo con que éstos fueron generados; aquí el objetivo es conocer la clave secreta y poder descriptar libremente cualquier texto.

Ataque con texto cifrado escogido. El criptoanalista conoce el algoritmo de cifrado, así como un criptograma seleccionado por él mismo y su correspondiente texto en claro, su objetivo es obtener el mensaje en claro de todo criptograma que intercepte.

Ataque con texto escogido. En este caso el criptoanalista además de conocer el algoritmo de cifrado y el criptograma que quiere descriptar, también conoce el criptograma de un texto en claro que él elija y el mensaje en claro de un criptograma también elegido por él. [20]

2.3. Criptografía Simétrica.

La criptografía simétrica utiliza la misma clave para cifrar y descifrar el mensaje de datos, es decir se basa en un secreto compartido [14].

Características de la Criptografía simétrica:

- La clave es la misma para cifrar que para descifrar un mensaje, por lo que sólo el emisor y el receptor deben conocerla.
- Se basa en operaciones matemáticas sencillas, por ello son fácilmente implementados en hardware.
- Debido al uso de operaciones básicas como son XOR y permutaciones son capaces de cifrar grandes cantidades de datos en poco tiempo en comparación con el uso de aritmética modular.

[14]

Los algoritmos criptográficos simétricos tienen dos versiones: cifrador en bloque y cifrador de flujo. El beneficio del uso de un algoritmo simétrico radica en el procesamiento rápido para cifrar y descifrar un alto volumen de datos. El cifrado simétrico es una táctica eficaz de almacenamiento de información sensible en una base de datos, un registro o archivo. [14] El cifrado simétrico se muestra en la figura 2.1.

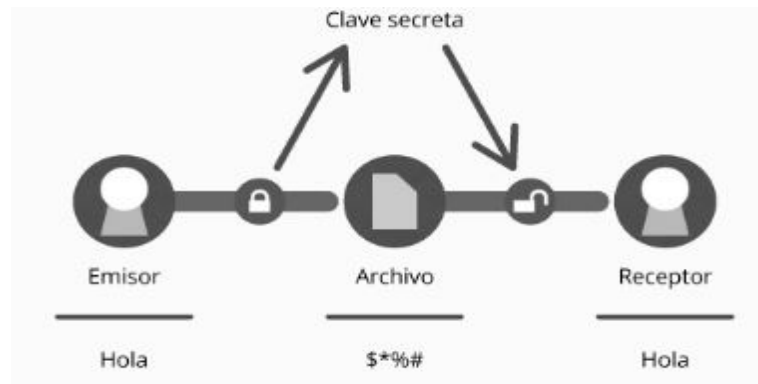


Figura 2.1: Diagrama de Criptografía Simétrica.

La sintaxis de un esquema de cifrado simétrico, esta dada por la siguiente definición.

Definición 2.1 *Un esquema de cifrado simétrico está conformado por una tripleta de algoritmos $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$, definidos como se describe a continuación:*

- *El algoritmo generador de claves **Gen** selecciona una llave K al azar del conjunto de llaves \mathcal{K} , esto se denotará como $K \xleftarrow{\$} \mathcal{K}$. Esta llave K será usada por los algoritmos **Enc** y **Dec**, esta llave la compartirán emisor y receptor.*
- *El algoritmo de cifrado **Enc**, toma como entrada un texto en claro $M \in \mathcal{M}$ y una llave K generada por **Gen** y regresa un texto cifrado $C \in \mathcal{C}$. Usualmente esto se denota como $C \leftarrow \text{Enc}_K(M)$.*
- *El algoritmo de descifrado **Dec**, toma como entrada un texto cifrado C y una llave K y regresa M . Esta operación se denota por $M \leftarrow \text{Dec}_K(C)$. Para que cualquier algoritmo de cifrado simétrico funcione correctamente, se debe garantizar que para todas las llaves posibles en \mathcal{K} y todos los posibles mensajes \mathcal{M} ,*

$$\text{Dec}_K(\text{Enc}_K(M)) = M.$$

2.4. Criptografía asimétrica.

La criptografía asimétrica es por definición aquella que utiliza dos claves diferentes para cada usuario, una para cifrar que se le llama llave pública y otra para descifrar que es la llave privada. Los algoritmos asimétricos son diferentes a los simétricos en un sentido muy importante [14]. Cuando se genera una llave simétrica, simplemente se escoge un número aleatorio de la longitud apropiada. Al generar llaves asimétricas el proceso es más complejo. [14].

Características de la Criptografía asimétrica:

- Se basa en operaciones matemáticas complejas.
- Se ejecuta de 100 a 1000 veces más lento que los algoritmos simétricos.

[14]

Los beneficios de la criptografía asimétrica son la solución a los problemas de la criptografía simétrica, pues las claves públicas pueden ser distribuidas con toda tranquilidad, no valen de nada sin las claves privadas. El cifrado asimétrico se emplea muy frecuentemente para pasar con seguridad una clave privada, que posteriormente, será la que se utilice para cifrar y/o descifrar otra información. El cifrado asimétrico puede ser representado como aparece en la figura 2.2.

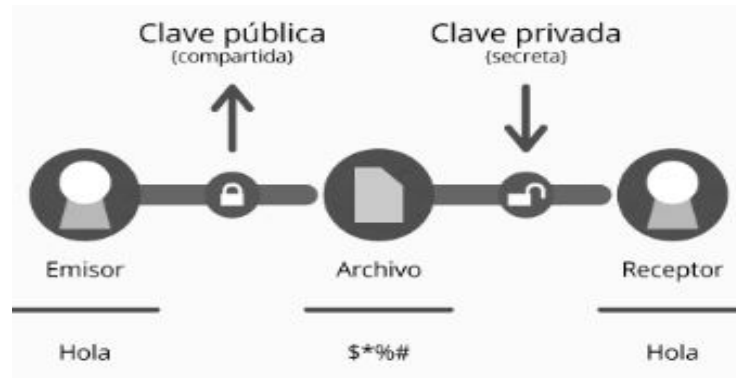


Figura 2.2: Diagrama de Criptografía Asimétrica.

2.5. Cifrado por bloques.

Los algoritmos de cifrado por bloques toman bloques de tamaño fijo del texto en claro y producen un bloque de tamaño fijo de texto cifrado, generalmente del mismo tamaño que la entrada. El tamaño del bloque debe ser lo suficientemente grande como para evitar ataques de texto cifrado. La asignación de bloques de entrada a bloques de salida debe ser uno a uno para hacer el proceso reversible y parecer aleatoria.

Para la asignación de bloques los algoritmos de cifrado simétrico realizan sustituciones y permutaciones en el texto en claro hasta obtener el texto cifrado.

La sustitución es el reemplazo de un valor de entrada por otro de los posibles valores de salida, en general, si usamos un tamaño de bloque k , el bloque de entrada puede ser sustituido por cualquiera de los bloques posibles. La permutación es un tipo especial de sustitución en el que los bits de un bloque de entrada son reordenados para producir el bloque cifrado, de este modo se preservan las estadísticas del bloque de entrada (el número de unos y ceros).

Los algoritmos de cifrado por bloques iterativos funcionan aplicando en sucesivas rondas una transformación a un bloque de texto en claro. La misma función es aplicada a los datos usando una subclave obtenida de la clave secreta proporcionada por el usuario. El número de rondas en un algoritmo de cifrado por bloques iterativo depende del nivel de seguridad deseado.

La sustitución es el reemplazo de un bloque de n bits por otro bloque de n bits en un espacio de 2^k [3]. Los cifradores por bloques mas usados son AES (Advanced Encryption Standard, por sus siglas en inglés) y DES (Data Encryption Standard, por sus siglas en inglés). [15]

Los cifradores por bloques pueden ser representados como se ve en la figura 2.3.

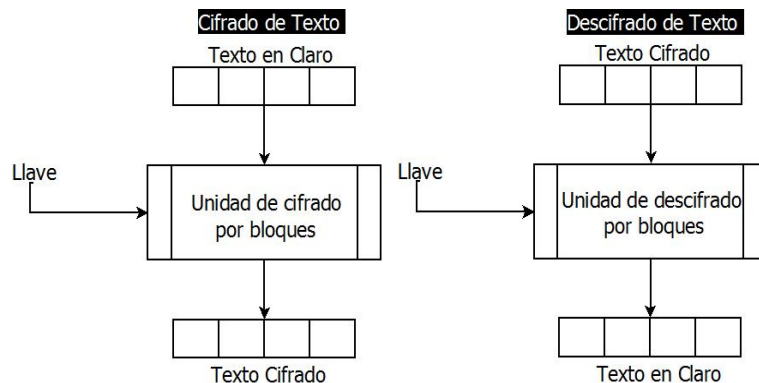


Figura 2.3: Diagrama de Cifradores por Bloques

2.6. RSA.

El algoritmo de clave pública RSA fue creado en 1978 por Rivest, Shamir y Adleman, y es el sistema criptográfico asimétrico más conocido y usado. Estos señores se basaron en el artículo de Diffie-Hellman sobre sistemas de llave pública, crearon su algoritmo y fundaron la empresa RSA Data Security Inc., que es actualmente una de las más prestigiosas en el entorno de la protección de datos. El sistema RSA se basa en el hecho matemático de la dificultad de factorizar números muy grandes. Para factorizar un número el sistema más lógico consiste en empezar a dividir sucesivamente éste entre 2, entre 3, entre 4,..., y así sucesivamente, buscando que el resultado de la división sea exacto, es decir, de resto 0, con lo que ya tendremos un divisor del número.

Ahora bien, si el número considerado es un número primo (el que sólo es divisible por 1 y por él mismo), tendremos que para factorizarlo habría que empezar por 1, 2, 3,... hasta llegar a él mismo, ya que por ser primo ninguno de los números anteriores es divisor suyo. Y si el número primo es lo suficientemente grande, el proceso de factorización es complicado y lleva mucho tiempo.

Basado en la exponenciación modular de exponente y módulo fijos, el sistema RSA crea sus claves de la siguiente forma:

- Se buscan dos números primos lo suficientemente grandes: p y q (de entre 100 y 300 dígitos).
- Se obtienen los números $n = p * q$ y $\phi = (p-1) * (q-1)$.
- Se busca un número e tal que no tenga múltiplos comunes con ϕ .
- Se calcula $d = e^{-1} \text{ mod } \phi$, con mod = resto de la división de números enteros. Y ya con estos números obtenidos, n es la clave pública y d es la clave privada. Los números p , q y ϕ se destruyen. También se hace público el número e , necesario para alimentar el algoritmo.

El cálculo de estas claves se realiza en secreto en la máquina en la que se va a guardar la clave privada, y una vez generada ésta conviene protegerla mediante un algoritmo criptográfico simétrico.

En cuanto a las longitudes de claves, el sistema RSA permite longitudes variables, siendo aconsejable actualmente el uso de claves de no menos de 1024 bits (se han roto claves de hasta 512 bits, aunque se necesitaron más de 5 meses y casi 300 ordenadores trabajando juntos para hacerlo).

RSA basa su seguridad en ser una función computacionalmente segura, ya que si bien realizar la exponenciación modular es fácil, su operación inversa, la extracción de raíces de módulo \mathcal{O} no es factible a menos que se conozca la factorización de e , clave privada del sistema.

RSA es el más conocido y usado de los sistemas de clave pública, y también el más rápido de ellos. Presenta todas las ventajas de los sistemas asimétricos, incluyendo la firma digital, aunque resulta más útil a la hora de implementar la confidencialidad el uso de sistemas simétricos, por ser más rápidos. Se suele usar también en los sistemas mixtos para encriptar y enviar la clave simétrica que se usará posteriormente en la comunicación cifrada. [12]

2.7. Firmas a ciegas.

Las firmas a ciegas son un tipo especial de firmas digitales en las que se firma algo que no se conoce. Para hacer firmas a ciegas se utilizan factores de opacidad, para ocultar el mensaje original que se requiere que esté firmado, y así la autoridad no pueda conocer lo que está firmando. Por lo tanto, el propósito de una firma a ciegas es evitar que el firmante B conozca el mensaje que firma; y así posteriormente, sea incapaz de asociar el mensaje que firmó con el remitente A. Entonces, las firmas a ciegas tienen aplicación en varias situaciones. A continuación se mencionan dos de ellas:

- Cuando se utiliza dinero electrónico. En este caso, m representa un valor monetario que A (el cliente) tiene derecho a gastar. Y así, cuando m y $s(m)$ se presentan a B (el banco) para efectuar el pago, B es incapaz de identificar al cliente que originalmente le dio ese dinero electrónico a firmar, pues le fue enviado de manera oculta. Lo anterior permite que la identidad de A permanezca anónima, y sus movimientos financieros no puedan ser monitoreados.
- En las elecciones electrónicas también pueden utilizarse las firmas a ciegas, ya que se requiere que B (una autoridad electoral) no conozca la identidad de A (el votante) debido a que el voto debe efectuarse de manera anónima. Sin embargo, es necesario que A demuestre que su voto m es válido. Lo cual se logra cuando A presenta ante B la firma $s(m)$. Y se sabe de antemano que B no puede asociar $s(m)$ a A, debido a que el votante previamente le envió a B su voto m pero de forma oculta para que se lo firmara. [11]

2.8. Funciones Hash.

A continuación se describirán las características de las *funciones hash*, también conocidas como *funciones de resumen*. Las funciones hash basan su definición en funciones de un solo

sentido (*one-way functions*, en inglés). Una función de un sólo sentido es aquella que para un valor x , es muy fácil calcular $f(x)$, pero es muy difícil hallar $f^{-1}(x)$. Es complicado en general, hallar funciones de éste tipo y probar que lo son.

Definición 2.2 *Una función hash, es una función de un sólo sentido cuya entrada m es un mensaje de longitud arbitraria y la salida es una cadena binaria de longitud fija. Al resumen o hash de un mensaje m , se le denotará como $h(m)$. Una función hash debe tener las siguientes propiedades:*

- *Para cualquier mensaje m , debe ser posible calcular $h(m)$ eficientemente.*
- *Dado $h(m)$, debe ser computacionalmente difícil, hallar un mensaje m' , tal que $h(m) = h(m')$.*
- *Debe ser computacionalmente difícil, hallar dos mensajes m y m' tales que $h(m) = h(m')$.*

Entre las funciones hash que se usan para criptografía están: MD2, MD4, MD5, donde MD significa *Message Digest*, y el algoritmo estándar al momento de escribir éstas notas es el *Secure Hash Algorithm* por sus siglas en inglés SHA. La MD5 fue diseñada por Ron Rivest, toma como entrada un mensaje de longitud arbitraria y proporciona como salida una cadena binaria de 128 bits. El mensaje de entrada se procesa por bloques de 512 bits. La SHA 256 fue diseñada por en NIST (National Institute of Standards and Technology) y se estableció como estándar en 1993. Recibe como entrada un mensaje con longitud menor a 2^{64} bits y como salida se obtiene una cadena binaria de 160 bits. Al igual que el MD5, se procesa en bloques de 512 bits [16].

2.9. Cómputo Nube.

El cómputo nube definido así por el NIST, es un modelo para permitir un acceso a la red ubicuo, es decir, que se encuentra presente en todas partes al mismo tiempo y conveniente a un conjunto de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que se puede aprovisionar y liberar rápidamente con un esfuerzo mínimo de gestión o una interacción entre el proveedor de servicios. Este modelo de cómputo nube se compone de 5 características esenciales, 3 modelos de servicio y 4 modelos de despliegue.

Características:

- **Auto-servicio bajo demanda.**

Un consumidor puede proporcionar unilateralmente capacidades del tiempo del servidor y el almacenamiento en red, según se necesite automáticamente sin interacción con cada proveedor de servicios.

- **Amplio acceso a la red.**

Las capacidades están disponibles a través de la red y se accede a través de mecanismos que promueven el uso por plataformas de cliente heterogéneas finas o gruesas (por ejemplo, teléfonos móviles, tablets, computadoras portátiles y estaciones de trabajo)

- **Agrupación de recursos.**

Los recursos informáticos del proveedor se agrupan para servir a múltiples consumidores utilizando un modelo de multi-usuario, con diferentes recursos físicos y virtuales asignados dinámicamente y reasignados de acuerdo con la demanda del consumidor. Hay una sensación de independencia de ubicación en que el cliente generalmente no tiene control o conocimiento sobre la ubicación exacta de los recursos proporcionados, pero puede especificar la ubicación en un nivel superior de abstracción (por ejemplo, país, estado o centro de datos). Ejemplos de recursos incluyen almacenamiento, procesamiento, memoria y ancho de banda de la red.

- **Elasticidad rápida.**

Las capacidades pueden ser suministradas elásticamente y liberadas, en algunos casos de forma automática, para escalar rápidamente hacia fuera y hacia adentro proporcional a la demanda. Para el consumidor, las capacidades disponibles para la provisión a menudo parecen ser ilimitadas y pueden ser apropiadas en cualquier cantidad en cualquier momento.

- **Servicio medido.**

Los sistemas de cómputo nube controlan y optimizan automáticamente el uso de recursos aprovechando una capacidad de medición en algún nivel de abstracción apropiado al tipo de servicio (por ejemplo, almacenamiento, procesamiento, ancho de banda y cuentas de usuario activas). El uso de recursos puede ser monitoreado, controlado y reportado, proporcionando transparencia tanto para el proveedor como para el consumidor del servicio utilizado.

Modelos de servicio.

- **Software como Servicio (SaaS).**

La capacidad proporcionada al consumidor es utilizar las aplicaciones del proveedor que se ejecutan en una infraestructura en la nube. Las aplicaciones son accesibles desde varios dispositivos cliente a través de una interfaz de cliente ligero, como un navegador web (por ejemplo, correo electrónico basado en web) o una interfaz de programa. El consumidor no gestiona ni controla la infraestructura oculta de la nube, incluyendo la red, los servidores, los sistemas operativos, el almacenamiento o incluso las capacidades de las aplicaciones individuales, con la posible excepción de las limitadas configuraciones específicas de la configuración de la aplicación.

- **Plataforma como Servicio (PaaS).**

La capacidad proporcionada al consumidor es desplegar en la infraestructura de la nube aplicaciones creadas por el consumidor, utilizando lenguajes de programación, bibliotecas, servicios y herramientas soportadas por el proveedor. El consumidor no gestiona ni controla la infraestructura oculta de la nube, incluyendo la red, los servidores, sistemas operativos o de almacenamiento, pero tiene control sobre las aplicaciones desplegadas y, posiblemente, configuración de configuración para el entorno de alojamiento de aplicaciones.

- **Infraestructura como Servicio (IaaS).**

La capacidad proporcionada al consumidor es proveer procesamiento, almacenamiento, redes y otros recursos de computación fundamentales donde el consumidor es capaz

de desplegar y ejecutar software arbitrario, que puede incluir sistemas operativos y aplicaciones. El consumidor no gestiona ni controla la infraestructura subyacente de la nube, sino que tiene control sobre los sistemas operativos, el almacenamiento y las aplicaciones implementadas; Y posiblemente un control limitado de componentes de red selectos (por ejemplo, firewalls de host).

Modelos de despliegue.

- **Nube privada.**

La infraestructura de la nube está preparada para el uso exclusivo de una sola organización que comprende varios consumidores (por ejemplo, unidades de negocio). Puede ser propiedad, administrado y operado por el órgano.

- **Nube de la comunidad.**

La infraestructura de la nube está preparada para uso exclusivo por una comunidad específica de consumidores de organizaciones que tienen preocupaciones compartidas (por ejemplo, misión, requisitos de seguridad, política y consideraciones de cumplimiento). Puede ser propiedad, administrado y operado por una o más de las organizaciones de la comunidad, un tercero, o una combinación de ellos, y puede existir dentro o fuera de las instalaciones.

- **Nube pública.**

La infraestructura de la nube está preparada para el uso abierto por el público en general. Puede ser propiedad, administrado y operado por una organización comercial, académica u gubernamental, o alguna combinación de ellos. Existe en las instalaciones del proveedor de la nube.

- **Nube híbrida.**

La infraestructura de la nube es una composición de dos o más infraestructuras de nube distintas (privadas, comunitarias o públicas) que siguen siendo entidades únicas, pero están unidas por una tecnología estandarizada o propietaria que permite la portabilidad de datos y aplicaciones (por ejemplo, burbujas de nube para equilibrar la carga entre Nubes).

Problemas en Cómputo Nube.

- **Interfaces y API poco seguros.**

Generalmente los proveedores de servicios en la nube ofrecen una serie de interfaces y API (del inglés, Application Programming Interface) para controlar e interactuar con los recursos. De este modo, toda la organización, el control, la provisión y la monitorización de los servicios cloud se realiza a través de estos API o interfaces. Dado que todo (autenticación, acceso, cifrado de datos, etc.) se realiza a través de estas herramientas, se hace necesario que los interfaces estén diseñados de forma segura, evitando así los problemas de seguridad, tanto los que son intencionados como los que se producen de forma accidental.

- **Pérdida o fuga de información.**

Existen muchas formas en las que los datos se pueden ver comprometidos. Por ejemplo, el borrado o modificación de datos sin tener una copia de seguridad de los originales,

supone una pérdida de datos. En la nube, aumenta el riesgo de que los datos se vean comprometidos ya que el número de interacciones entre ellos se multiplica debido a la propia arquitectura de la misma. Esto deriva en pérdida de imagen de la compañía, daños económicos y, si se trata de fugas, problemas legales, infracciones de normas, etc.

- **Secuestro de sesión o servicio.**

En un entorno en la nube, si un atacante obtiene las credenciales de un usuario del entorno puede acceder a actividades y transacciones, manipular datos, devolver información falsificada o redirigir a los clientes a sitios maliciosos.

Capítulo 3

Requerimientos del sistema

3.1. Requerimientos Funcionales.

Servidor de Llaves	
ID	Descripción
RF – SLL1	El sistema permitirá gestionar estados para la administración de llaves de usuario a través de una clave secreta (K_s) propia del servidor de llaves.
RF – SLL2	El sistema debe gestionar 3 estados en el servidor de llaves: Generación, Cambios y Eliminación de llaves de usuarios para manipular un archivo.
RF – SLL3	El sistema comenzará la creación de una llave (K) cuando el usuario solicite la carga de un archivo a almacenar.
RF – SLL4	El sistema deberá enviar al cliente la llave generada en el servidor de llaves (K) del archivo que el usuario solicitó carga al servicio de almacenamiento.
RF – SLL5	El sistema almacenará en la base de datos del servidor de llaves, la llave (K) generada por cada nuevo archivo que se solicite cargar al servicio de almacenamiento.
RF – SLL6	El sistema actualizará la base de datos del servidor de llaves cada vez que una nueva llave sea generada dentro de éste.
RF – SLL7	El sistema detectará mediante la función hash del archivo $[H(F)]$ si se le ha generado anteriormente una llave (K), si ésta existe en la base de datos se procederá a enviarla al usuario que la solicita.
RF – SLL8	El sistema modificará la llave (K) para un archivo cuando este sea modificado por el propietario del archivo (F).
RF – SLL9	El sistema eliminará la llave (K) para un archivo cuando este sea eliminado del servicio de almacenamiento por el usuario.

Tabla 3.1: Requerimientos funcionales del servidor de llaves

Cliente	
ID	Descripción
RF – CL1	El sistema permitirá al usuario gestionar archivos: Subir, Descargar y Eliminar
RF – CL2	El sistema permitirá al usuario calcular a su archivo (F) una función hash $[H(F)]$ cuando dicho usuario solicite una nueva carga de éste.
RF – CL3	El sistema permitirá al usuario iniciar el proceso de carga de un archivo (F) al enviar al servidor de llaves la función hash $[H(F)]$ calculada de éste.
RF – CL4	El sistema entregará al cliente la llave (K) generada en el servidor de llaves del archivo (F) que el usuario solicitó cargar al servicio de almacenamiento.
RF – CL5	El sistema mandará a cifrar la llave (K) obtenida del servidor de llaves junto con el archivo (F) que el usuario desea cargar al servicio de almacenamiento.
RF – CL6	El sistema obtendrá el cifrado C1 correspondiente al archivo original (F) y lo mandará al cliente, cifrado que el usuario cargará al servicio de almacenamiento.
RF – CL7	El sistema calculará al cifrado C1 una función hash $[H(C1)]$
RF – CL8	El sistema mandará a cifrar la llave secreta (Ka) del usuario junto con la llave (K) obtenida del servidor de llaves del archivo (F) que éste usuario desea cargar al servicio de almacenamiento.
RF – CL9	El sistema obtendrá el cifrado C2 correspondiente a la llave secreta (Ka) del usuario y lo mandará al cliente, cifrado que el usuario cargará al servicio de almacenamiento.
RF – CL10	El sistema enviará $[H(C1)]$ hacia el servicio de almacenamiento para su evaluación.
RF – CL11	El sistema permitirá al usuario cargar los cifrados C1 y C2 servicio de almacenamiento.
RF – CL12	El sistema notificará al usuario el estatus final de la gestión de sus archivos en el servicio de almacenamiento.
RF – CL13	El sistema permitirá al usuario iniciar el proceso de descarga de un archivo (F) cuando el usuario elige de su lista de archivos el nombre de este.
RF – CL14	El sistema entregará al cliente el cifrado C1 correspondiente al archivo (F) que el usuario eligió y el cifrado C2 que corresponde a la llave (K) del mismo archivo.
RF – CL15	El sistema enviará desde el cliente al módulo cifrador la llave secreta (Ka) del usuario junto con el cifrado C1 y C2.
RF – CL16	El sistema obtendrá el archivo original (F) que solicitó el usuario para la descarga.
RF – CL17	El sistema permitirá al usuario eliminar un archivo (F) cuando el usuario elige alguno de su lista de archivos cargados en el servicio de almacenamiento.
RF – CL18	El sistema enviará al servidor de almacenamiento el nombre del archivo (F) que el usuario solicite eliminar.

Tabla 3.2: Requerimientos funcionales del cliente

Servicio de almacenamiento (Nube)	
ID	Descripción
RF – SA1	El sistema permitirá al servicio de almacenamiento gestionar archivos: Almacenar, Descargar y Eliminar.
RF – SA2	El sistema creará una base de datos con los cifrados que cada usuario tenga registrados dentro del servicio de almacenamiento.
RF – SA3	El sistema permitirá al servicio de almacenamiento guardar cualquier cifrado que el usuario solicite cargar.
RF – SA4	El sistema permitirá al servicio de almacenamiento descargar cualquier cifrado que el usuario tenga en su lista de archivos.
RF – SA5	El sistema permitirá al servicio de almacenamiento eliminar cualquier cifrado que el usuario tenga en su lista de archivos.
RF – SA6	El sistema actualizará dentro de la base de datos del servicio de almacenamiento, la lista de archivos (F) que el usuario a gestionado dentro de éste.
RF – SA7	El sistema entregará $[H(C1)]$ generado por el cliente al servicio de almacenamiento
RF – SA8	El sistema almacenará en una base de datos dentro del servicio de almacenamiento a $[H(C1)]$ asociando al cliente que lo envió.
RF – SA9	El sistema corroborará la existencia $[H(C1)]$ dentro de la base de datos, si existe procederá al RF – SA6, si no procederá al RF – SA3
RF – SA10	El sistema entregará al servidor de almacenamiento el nombre del archivo (F) que el usuario solicite eliminar.
RF – SA11	El sistema eliminará el nombre del archivo (F) de la lista de archivos que el usuario tiene registrados en la base de datos del servidor de almacenamiento.
RF – SA12	El sistema enviará una notificación al cliente con el estatus de cada gestión hecha por el usuario sobre sus archivos.
RF – SA13	El sistema recibirá el nombre del archivo (F) que el usuario solicita descargar.
RF – SA14	El sistema buscará el nombre del archivo (F) en la base de datos de los cifrados que tiene el usuario registrados en el servicio de almacenamiento.
RF – SA15	El sistema enviará al cliente el archivo C1 correspondiente al archivo (F) que el usuario eligió y el cifrado C2 que corresponde a la llave (K) del mismo archivo.

Tabla 3.3: Requerimientos funcionales del Servicio de almacenamiento (Nube)

Módulo Cifrador	
ID	Descripción
RF – MC1	El sistema recibirá en el modulo cifrador la llave (K) obtenida del servidor de llaves junto con el archivo (F) que el usuario desea cargar.
RF – MC2	El sistema cifrará la llave (K) junto con el archivo (F) que el usuario a solicitado, y generara el cifrado C1.
RF – MC3	El sistema enviará al cliente el archivo C1 obtenido en el módulo cifrador.
RF – MC4	El sistema recibirá en el módulo cifrador la llave secreta (Ka) del usuario junto con la llave (K) obtenida del servidor de llaves del archivo (F) que éste usuario desea cargar
RF – MC5	El sistema cifrará la llave secreta (Ka) junto con la llave (K) que el usuario a solicitado, y generará el cifrado C2
RF – MC6	El sistema enviará al cliente el archivo C2 obtenido en el módulo cifrador.
RF – MC7	El sistema recibirá en el modulo cifrador la llave secreta (Ka) junto con el cifrado C1 y el cifrado C2 que el usuario solicite descargar.
RF – MC8	El sistema descifrá el C1 junto con la llave secreta (Ka) que el usuario a solicitado descargar y generará la llave (K) del archivo (F).
RF – MC9	El sistema descifrá el C2 junto con la llave (K) que el usuario a solicitado descargar y generará el archivo original (F).

Tabla 3.4: Requerimientos funcionales del Módulo Cifrador

Capítulo 4

Análisis y Diseño

4.1. Business Process Model and Notation (BPMN)

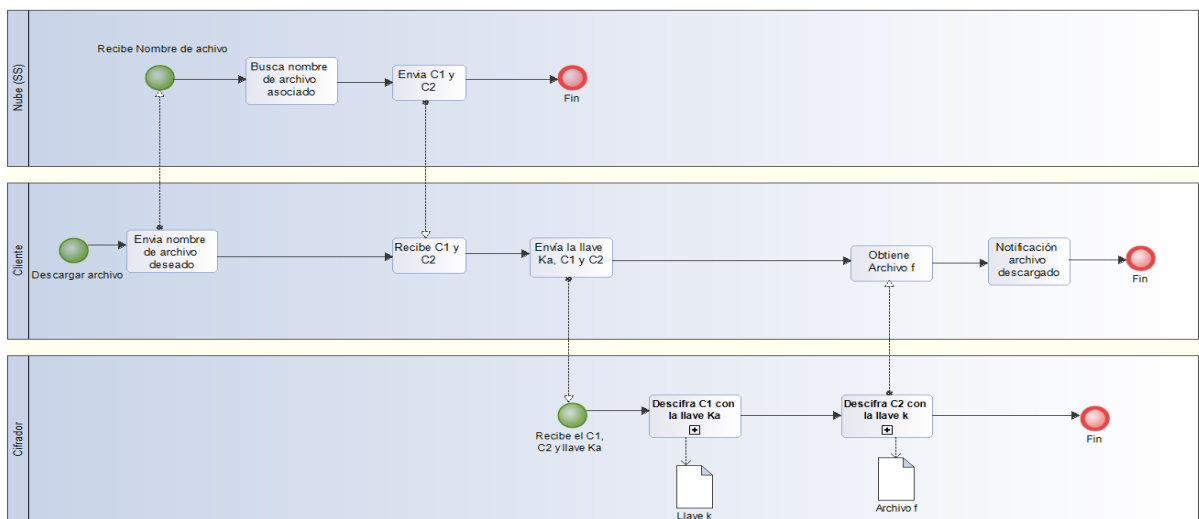


Figura 4.1: BPMN Subir archivo.

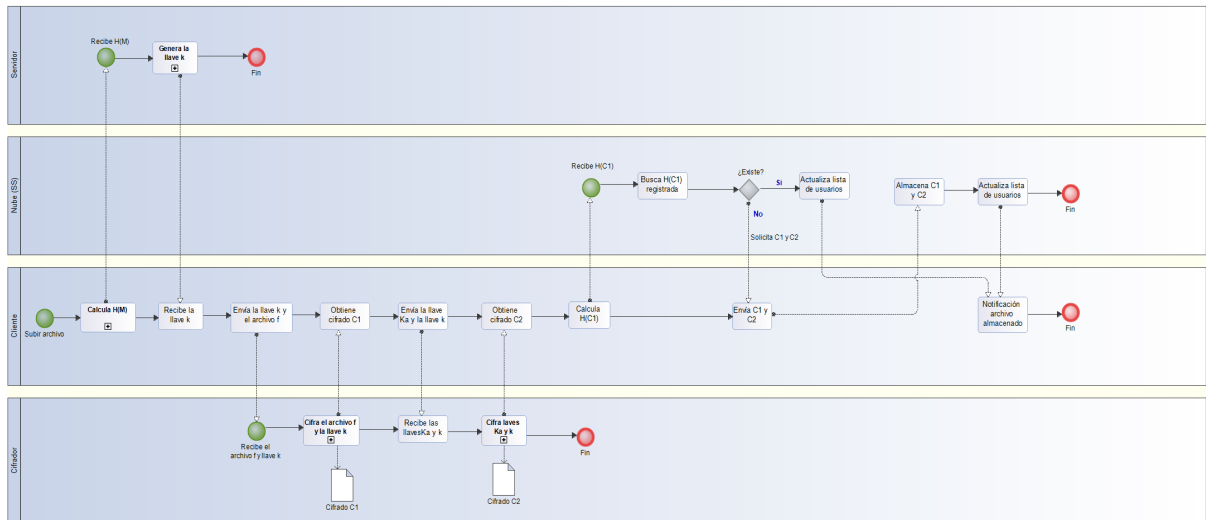


Figura 4.2: BPMN Descargar archivo.

Referencias

- [1] R. Bellare, Keelveedhi. *Message-locked encryption and secure deduplication.*, volume 7881. EUROCRYPT, 2013.
- [2] R. Bellare, Keelveedhi. Dupless: Server-aided encryption for deduplicated storage., 2013:429.
- [3] D. Chakraborty and F. Rodríguez-Henríquez. Block cipher modes of operation from a hardware implementation perspective. In Ç. K. Koç, editor, *Cryptographic Engineering*, pages 321–363. Springer, 2009.
- [4] T. C. y. P. A. Cooley J. Abs: the apportioned backup system. MIT Laboratory for Computer, 2004.
- [5] M. C. y. N. B. Cox L. *SIGOPS Oper. Syst.* Pastiche: making backup cheap and easy.
- [6] B. W. J. S. D. y. T. M. Donceur J. R., Adya A. *Reclaiming space from duplicate files in a serverless distributed file system.* ICDCS, 2002.
- [7] L. L. Erika. Fundamentos de seguridad informática. Universidad Nacional Autónoma de México, 2012. <http://redyseguridad.fi-p.unam.mx/proyectos/seguridad/ServNoRepudio.php>.
- [8] P. HP. Estadísticas que todos deberían conocer sobre cloud computing, 2016. <http://www.popa.hn/index.php/es/soluciones/96-otras-noticias/152-20-estadisticas-que-todos-los-cios-deberian-conocer-sobre-cloud-computing>.
- [9] L. B. Jaquelina. Fundamentos de criptografía. Universidad Nacional Autónoma de México, 2012. <http://redyseguridad.fi-p.unam.mx/proyectos/criptografia/criptografia/index.php/1-panorama-general/11-concepto-decriptografia>.
- [10] Microsoft. Data deduplication overview. Biblioteca TechNet, 2015. [https://technet.microsoft.com/enus/library/hh831602\(v=ws.11\).aspx](https://technet.microsoft.com/enus/library/hh831602(v=ws.11).aspx).
- [11] G. Z. C. Patricia. Diseño y desarrollo de un sistema para elecciones electrónicas seguras (seles) . Master’s thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2005.
- [12] s/a. Rsa. Herramientas WEB para la enseñanza de Protocolos de Comunicación. <http://neo.lcc.uma.es/evirtual/cdd/tutorial/presentacion/rsa.html>.
- [13] s/a. Flud backup, 2011. http://flud.org/wiki/Flud_Backup.

- [14] s/a. Cifrado simetrico. Guía de Gnu Privacy Guard, 2015. <https://www.gnupg.org/gph/es/manual/c190.html#AEN201>.
- [15] T. O. Sergio. Introducción a la criptología. *InfoCentreUV*, 2003.
- [16] W. Stallings. *Cryptography and Network Security: Principles and Practice*. Pearson Education, 5a edition, 2002.
- [17] H. D. y. W. N. Wilcox-O’Hearn Z. *Tahoe: The least-authority*. In Proceedings of the 4th ACM, 2008.
- [18] E. A. M. y M.C. Ma. Jaquelina López Barrientos. Fundamentos de criptografía. Universidad Nacional Autónoma de México, 2012. <http://redyseguridad.fi-p.unam.mx/proyectos/criptografia/criptografia/index.php/1-panorama-general/11-concepto-de-criptografia>.
- [19] E. A. M. y M.C. Ma. Jaquelina López Barrientos. Fundamentos de criptografía. Universidad Nacional Autónoma de México, 2012. <http://redyseguridad.fi-p.unam.mx/proyectos/criptografia/criptografia/index.php/1-panorama-general/13-servicios-y-mecanismos-de-seguridad/131-servicios-de-seguridad>.
- [20] E. A. M. y M.C. Ma. Jaquelina López Barrientos. Fundamentos de seguridad informática. Universidad Nacional Autónoma de México, 2012. <http://redyseguridad.fi-p.unam.mx/proyectos/criptografia/criptografia/index.php/1-panorama-general/14-ataques/142-ataques-a-los-metodos-de-cifrado>.