UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

Laboratorio 3

Criptografía

Integrantes: Williams Álvarez
Patricia Melo

Curso: Sistemas de Comunicación

Profesor(a): Carlos González

Ayudante: Fernanda Muñoz

25 de Septiembre de 2020

Tabla de contenidos

Índice de figuras

Índice de cuadros

1. Introducción

Hoy en día gran parte del mundo está conectado a internet a través de sus dispositivos móviles, televisores inteligentes, computadores personales, entre otros, así sea para disfrutar del contenido digital y/o navegar por diferentes páginas web. Pero para que los datos lleguen a los dispositivos conectados, estos deben viajar por la red y ser enviados como paquetes. Para que estos paquetes lleguen a destino existen diferentes protocolos que permiten la comunicación y conexión de computadores con la red. Estos protocolos especifican cómo los datos deben ser formateados, direccionados, transmitidos, enrutados y recibidos por el destinatario. Dichos protocolos se ubican en diferentes capas, pero en ninguna de ellas existe un protocolo de seguridad para proteger la información transmitida de los ciberataques, es por este motivo que en la capa de aplicación (del modelo TCP/IP) se debe implementar de manera manual esta seguridad, protegiendo la privacidad e integridad de los datos.

1.1. Objetivos

Aprender de manera práctica los conceptos de criptografía y criptoanálisis, desarrollando un sistema de encriptación simétrico de información para encubrir posibles ataques cibernéticos. Para esto se debe diseñar un cifrado simétrico, implementarlo (identificando diferentes conceptos vistos en cátedra), y realizando varias pruebas para luego analizar los resultados obtenidos

1.2. Estructura informe

El informe consta inicialmente de un marco teórico en donde se exponen conceptos importantes que serán tratados a lo largo del documento. A continuación, se tiene el desarrollo de la experiencia con todas las etapas realizadas y sus resultados obtenidos. Posteriormente, se presenta el análisis de los resultados, en donde se analizan e identifican si estos son los esperados, relacionándolos también con los contenidos descritos en el marco teórico. Finalmente, se entregan las conclusiones y referencias del trabajo realizado, en donde se sintetizan los principales resultados obtenidos.

2. Marco Teórico

En este capítulo se presentan algunos conceptos que son de utilidad para el entendimiento del informe.

2.1. Modelo TCP/IP

Según Ángel Robledano (2019) este modelo es un protocolo para comunicación de redes, permitiendo así la conexión de computadores con la red. A continuación se presentan los componentes de este modelo (ver Figura 1), los cuales reciben el nombre de capas.

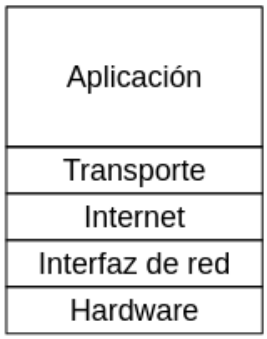

Figura 1: Capas del modelo TCP/IP

De acuerdo con la página de Oracle (S.F.), las capas tienen la siguiente definición:

- **Aplicación:** Esta capa define las aplicaciones de red y los servicios de Internet estándar que puede utilizar un usuario. Estos servicios utilizan la capa de transporte para enviar y recibir datos.
- **Transporte:** Aquí se garantiza que los paquetes lleguen en secuencia y sin errores, al intercambiar la confirmación de la recepción de los datos y retransmitir los paquetes perdidos. Este tipo de comunicación se conoce como transmisión de punto a punto.
- **Internet:** También conocida como capa de red o capa IP, acepta y transfiere paquetes para la red. Esta capa incluye el potente Protocolo de Internet (IP), el protocolo de resolución de direcciones (ARP) y el protocolo de mensajes de control de Internet (ICMP).

- **Interfaz de Red:** Esta capa identifica el tipo de protocolo de red del paquete, en este caso TCP/IP. La capa de vínculo de datos proporciona también control de errores y estructuras.

- **Hardware:** Esta capa especifica las características del hardware que se utilizará para la red. Describe los estándares de hardware como IEEE 802.3, la especificación del medio de red Ethernet, y RS-232, la especificación para los conectores estándar.

2.2. Seguridad informática

De acuerdo a Stallings (2010, p. 9) la seguridad informática se define como "la protección otorgada a un sistema de información automatizado para lograr los objetivos aplicables de preservar la integridad, disponibilidad y confidencialidad de los recursos del sistema de información (incluye hardware, software, firmware, información/datos y telecomunicaciones)".

Esta definición presenta tres objetivos clave que están en el corazón de la seguridad informática:

- **Confidencialidad:** es la cualidad de la información para no ser divulgada a personas o sistemas no autorizados.

- **Integridad:** es la cualidad de la información para ser correcta y no haber sido modificada, manteniendo sus datos exactamente tal cual fueron generados, sin manipulaciones ni alteraciones por parte de terceros.

- **Disponibilidad:** es la cualidad de garantizar el acceso y el uso oportuno y confiable de la información.

2.3. Ciberataque

Los ciberataque según Caser Seguros (S.F.) son un conjunto de acciones ofensivas contra sistemas de información (como bases de datos, redes computacionales, etc.). Estos

ataques están hechos para dañar, alterar o destruir instituciones, personas o empresas y utilizan la internet como principal medio. Los ciberataques se pueden clasificar en los siguientes grupos.

- **Cibercrimen:** Roban identidades de empresas o personas para realizar fraudes bancarios.
- **Hacktivismo:** Vulnerar páginas de empresas grandes, gobiernos u ordenadores privados para realizar protestas con fines ideológicos o sociales.
- **Ciberespionaje:** Es el robo de información sensible y valiosa que puede venderse a precios muy elevados en el mercado negro.
- **Ciberterrorismo:** Dirigido principalmente contra gobiernos o países, afectando servicios como salud, defensa, infraestructura, entre otros.

2.4. Criptografía

Según Guillermo Venturini (2020), la criptografía es una técnica utilizada para proteger documentos y datos, a través del cifrado de mensajes o codificación de estos. Lo anterior permite autentificar la identidad de los usuarios, también da confiabilidad e integridad de la información.

De acuerdo a Yran Marreno (2003), la criptografía se divide en dos grandes ramas, de llave privada o simétrica y de llave pública o asimétrica. La primera se refiere al conjunto de métodos que permiten una comunicación segura entre las partes siempre que, con anterioridad, se intercambie la llave correspondiente, que se denomina llave simétrica. La simetría se refiere a que las partes tienen la misma llave, tanto para cifrar como para descifrar. La idea general es aplicar diferentes funciones al mensaje que se desea cifrar de modo tal, que sólo conociendo la llave, pueda descifrarse, un ejemplo de esto es el cifrado AES.

La criptografía de llave asimétrica es por definición aquella que utiliza dos llaves diferentes para cada usuario, una para cifrar que se llama llave pública y otra para descifrar que es la llave privada. El nacimiento de la criptografía asimétrica ocurrió como resultado de la búsqueda de un modo más práctico de intercambiar las llaves simétricas.

3. Desarrollo y Resultados

Para poder realizar la experiencia a cabalidad es necesario tener instaladas ciertas herramientas en el computador, las principales son Python 3 (con su módulo PyCriptodome) y la IDE Pycharm.

El desarrollo de esta experiencia se divide en varios pasos, estos se detallan a continuación.

3.1. Diseñar cifrado simétrico

Para esto se realizó una réplica del algoritmo AES, el cual utiliza una sola llave (simétrico) y realiza el cifrado por bloques. Según Boxcrypto (S.F.) la longitud de las llaves puede ser de 128 bits, 192 bits o 256 bits, y hasta el día de hoy aun no existen ataques factibles frente a este cifrado, por lo que lo hace bastante seguro.

El algoritmo se basa en varias sustituciones, permutaciones y combinaciones lineales, cada una ejecutadas en bloques de 16 bytes. Estas operaciones se ejecutan reiteradas veces (rondas), y en cada una de ellas se ocupa una llave circular única a partir de la llave original, lo que hace que el cambio de una letra en el texto o un bit en la llave modifique por completo el bloque cifrado.

Los pasos para crear el cifrado son, primero especificar el tamaño de la llave y de la sal (este último no es obligación). Luego definir el texto que será cifrado, junto con la llave a utilizar. Finalmente se realiza la encriptación y desencriptación.

Cabe destacar que la réplica del algoritmo fue realizada a partir de la página Sodocumentation (S.F.).

3.2. Implementar cifrado

El cifrado simétrico implementado se presentará a continuación. En primer lugar, se define el tamaño del vector de inicialización, el tamaño de la llave y el tamaño de la sal.

```
IV_SIZE = 16
KEY_SIZE = 32
```

```
SALT_SIZE = 16
```

Como se puede observar el tamaño del vector de inicialización es de 16 bytes (128 bits), por su parte el tamaño de la llave es de 32 bits (256 bits), el tamaño de la sal es de 16 bytes (128 bits) y como se utilizará AES, este tiene un tamaño de bloque de datos fijo de 16 bytes (128 bits).

Una vez definidos los tamaños, en la función de encriptación se genera en primer lugar una sal aleatoria de 16 bytes. Luego se obtiene la llave y el vector de inicialización con el algoritmo PDKDF2 de hashlib el cual generar un vector de inicialización de 128 bits y una llave de cifrado de 256 bits aleatorios a partir de la contraseña y tamaños establecidos con anterioridad. Con la llave y el vector de inicialización obtenidos se procede a encriptar el texto con AES, a través de un cifrado por bloques CFB (que utiliza XOR), y codificación 'utf-8' para el texto a cifrar y la contraseña, es en este punto del algoritmo en donde se debe producir el efecto avalancha. Una vez cifrado el texto se concatena la sal al texto cifrado, con el fin de que sea más difícil de descifrar al aplicar fuerza bruta o criptoanálisis. Finalmente, se retorna el texto cifrado obtenido.

```
def encryp(password, text):
    salt = os.urandom(SALT_SIZE)
    derived = hashlib.pbkdf2_hmac('sha256', password.encode('utf-8'),
        salt, 100000, dklen=IV_SIZE + KEY_SIZE)
    iv = derived[0:IV_SIZE]
    key = derived[IV_SIZE:]
    encrypted_text = salt + AES.new(key, AES.MODE_CFB,
        iv).encrypt(text.encode('utf-8'))
    return encrypted_text
```

Por su parte en la función de desencriptación, se obtiene la sal a partir de su tamaño y del texto cifrado. Luego, se obtiene la llave y el vector de inicialización con el algoritmo PDKDF2 de hashlib de la misma manera que fue realizado en el proceso de encriptación. Con AES se procede a desencriptar el texto, utilizando la llave y el vector de inicialización obtenidas.

```
def decrypt(password, encrypted_text):
    salt = encrypted_text[0:SALT_SIZE]
    derived = hashlib.pbkdf2_hmac('sha256', password.encode('utf-8'),
        salt, 100000, dklen=IV_SIZE + KEY_SIZE)
    iv = derived[0:IV_SIZE]
    key = derived[IV_SIZE:]
    decrypted_text = AES.new(key, AES.MODE_CFB,
        iv).decrypt(encrypted_text[SALT_SIZE:])
```

A continuación, se presenta un ejemplo de la ejecución del programa, en donde la parte verde del texto cifrado es la sal que se le agrega, y el texto en color negro corresponde al mensaje cifrado como tal.

Texto	Hola mundo
Texto cifrado	9249a84c6b0b17f25207223f43282427234b1f83cbfe394613e8
Texto descifrado	Hola mundo

Cuadro 1: Ejemplo ejecución

3.3. Pruebas del sistema cifrado

Para evaluar al cifrado que se implementó se realizan las pruebas de avalancha y de throughput (al encriptar y desencriptar). Los resultados obtenidos se presentan a continuación.

3.3.1. Avalancha

Se espera que si el texto a cifrar cambia levemente (aunque sea un bit), entonces al momento de encriptar se debería ver una gran diferencia en el texto cifrado. Dicho lo anterior, se exponen 3 ejemplos en la Tabla 2, donde se prueba el efecto avalancha del algoritmo construido.

Texto	Texto cifrado
Hola mundo	9249a84c6b0b17f25207223f43282427234b1f83cbfe394613e8
Holamundo	31f5acb9254923073c09393536ff0a6326d380ab84ffdb7597
Hola mudo	1cab037b2cf1d2a8b1edb0338433ab32e8bc29170d6a879036

Cuadro 2: Resultados efecto avalancha

Cabe destacar que en cada texto se utilizó una sal diferente, la cual se origina de manera aleatoria, es por ese motivo que todos parten diferentes. Pero eso no es la única diferencia, sino que también al ver el resto de los textos cifrados (omitiendo la sal) son muy diferentes entre ellos, por lo que no se puede encontrar relación alguna.

3.3.2. Throughput

Para esta experiencia se debe ocupar la definición de throughput dada en (1), donde el numerador corresponde al tamaño del bloque en kilobytes y el denominador al tiempo de ejecución en segundos, que puede ser durante la encriptación o desencriptación. Esto es utilizado para calcular el rendimiento según Fernando Piñal-Moctezuma (2009).

$$Throughput = \frac{Tamano_{bloque}}{Tiempo} \tag{1}$$

Como AES tiene por defecto el tamaño de bloque de 16 kilobytes, solo se procede a calcular el tiempo de ejecución durante la encriptación y el tiempo de ejecución para la desencriptación, utilizando como texto *"Hola mundo"* en ambos casos.

A continuación se presentan los calculos de throughput en las ecuaciones (2) y (3).

$$Throughput_{cifrado} = \frac{0{,}016\ kB}{0{,}3426\ s} = 0{,}0467\ kB/s \tag{2}$$

$$Throughput_{descifrado} = \frac{0{,}016\ kB}{0{,}2857\ s} = 0{,}0560\ kB/s \tag{3}$$

También, se solicitó graficar throughput v/s tamaño de bloque para los dos casos (cifrado y descifrado), pero esto no fue posible debido a que AES tiene un tamaño de bloque fijo, imposibilitando su cambio para realizar dichos gráficos.

4. Análisis de Resultados

En este capítulo se realizará el análisis de los resultados obtenidos de la experiencia descritos anteriormente.

4.1. Diseño e implementación cifrado simétrico

Una vez diseñado e implementado el cifrado simétrico AES, se puede determinar que cumple su objetivo, ya que realizada varias pruebas se logra encriptar y desencriptar con éxito todas los mensajes, otorgando seguridad a la información (confidencialidad y disponibilidad de los datos). Cabe destacar que esto es posible siempre y cuando la clave usada para encriptar sea la misma que para desencritar, ya que si cambia esta última el mensaje no podrá ser desencriptado o se obtendrá un mensaje erróneo.

Por otro lado, se puede destacar que el uso de sal ayuda a que el mensaje cambie su tamaño y se vuelve menos vulnerable a los ataques de fuerza bruta. Si bien aún el mensaje puede ser descifrado, la tarea se vuelve más difícil de realizar. Además, el cifrado AES resulta ser bastante bueno ya que el mensaje se vuelve ilegible una vez cifrado, protegiéndolo de algunos ciberataques.

4.2. Pruebas del sistema cifrado

Las pruebas realizadas fueron 3, una de avalancha y dos de throughput.

Para el primer caso, observando los resultados de la Tabla 2, se puede notar que pese a haber realizado cambios ínfimos en los textos de pruebas, luego de ser cifrados estas pequeñas diferencias se vuelven significativas, dejando irreconocible a los textos originales y sin poder relacionarlos entre ellos, aunque sean todos muy similares al inicio. De esta forma se demuestra que el algoritmo empleado se comporta como efecto avalancha, lo cual es una ventaja frente a otros cifradores.

Otra ventaja de este cifrado según Lindsay Mason (S.F.), es que resulta muy fácil utilizar este cifrado, es rápido y ocupa menos recursos informáticos que otros cifrados. Por otro lado, existen varias desventajas, la principal es la necesidad de compartir la clave, esto debe hacerse con mucho cuidado para que no sea revelada.

Con respecto al throughput de codificación y decodificación, al no poder ser graficados con respecto a la variación del tamaño de bloque, no se puede realizar las comparaciones y análisis esperados, pero lo que sí se puede analizar es la diferencia entre las ecuaciones (2) y (3).

Como se puede observar, a partir de los resultados dados en el capítulo anterior, el throughput en la encriptación es menor que el throughput en la desencriptación, lo que quiere decir que éste tiene peor rendimiento que el proceso de desencriptar. En otras palabras, se demora más encriptar que desencriptar, esto según la implementación del algoritmo que se realizó.

5. Conclusiones

En esta experiencia se logró comprender con mayor profundidad el funcionamiento del cifrado simétrico, observando cómo se logra encriptar y desencriptar varios mensajes con la misma clave aplicando un cifrado AES.

Fue interesante percibir como el mensaje se vuelve ilegible una vez realizado el cifrado, y cómo es posible reconstruir el mensaje al conocer la clave y el algoritmo utilizado para encriptar, junto con el tamaño de la llave, el tamaño del vector de inicialización, el tamaño de la sal y la posición de este último. Asimismo, fue interesante descubrir que el efecto avalancha ocurre satisfactoriamente, por lo menos en las pruebas utilizadas, lo que nos indica que el cifrado de bloques utilizado es de buena calidad.

Por otro lado, dado la implementación del algoritmo, se observó que en la mayoría de las ejecuciones se produce que el throughput de encriptación es menor que el de desencriptación, lo que permitió inferir que se tiene un mejor rendimiento al momento de descifrar el mensaje que al cifrar.

Con respecto a los objetivos, estos se cumplieron en su mayoría, ya que se logró aprender de manera práctica los conceptos de criptografía y criptoanálisis, en particular a través de un sistema de cifrado simétrico, solo que no se pudo realizar la comparación y análisis de los gráficos con respecto al throughput v/s tamaño de bloque que se solicitó.

Bibliografía

Boxcrypto (S.F.). Cifrado aes y rsa. [Online] https://www.boxcryptor.com/es/encryption/.

Caser Seguros (S.F.). ¿qué es un ciberataque y qué tipos hay? [Online] https://www.caser.es/seguros-empresas/articulos/que-es-un-ciberataque-y-tipos.

Marreno, Y. (2003). La criptografía como elemento de la seguridad informática. [Online] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352003000600012#cargo.

Mason, L. (S.F.). Ventajas y desventajas de la criptografía de clave simétrica. [Online] https://techlandia.com/ventajas-desventajas-criptografia-clave-simetrica-info_276624/.

Piñal-Moctezuma, F. (2009). Implementación en hardware del estándar de encriptación avanzado (aes), en una plataforma fpga, empleando el microcontrolador picoblaze. [Online] https://www.researchgate.net/publication/285598341_IMPLEMENTACION_EN_HARDWARE_DEL_ESTANDAR_DE_ENCRIPTACION_AVANZADO_AES_EN_UNA_PLATAFORMA_FPGA_EMPLEANDO_EL_MICROCONTROLADOR_PICOBLAZE.

Sodocumentation (S.F.). Python language seguridad y criptografía. [Online] https://sodocumentation.net/es/python/topic/2598/seguridad-y-criptografia.

Stallings, W. (2010). *Cryptography and Network Security: Principles and Practice*. Prentice Hall, 5 edition.

Universidad Santiago de Chile, Facultad de Ingeniería (2020). Laboratorio 3: Criptografía. [Online] https://docs.google.com/document/d/1HsK3KQTF2E9080icyqDv_8QkKcNFgdkxDT7MlfsLObE/edit.

Venturini, G. (2020). ¿qué es la criptografía? [Online] https://www.tecnologia-informatica.com/que-es-la-criptografia/.

Ángel Robledano (2019). ¿qué es tcp/ip? [Online] https://openwebinars.net/blog/que-es-tcpip/.