Guatemala 05 de octubre de 2020

Avances en el Proyecto #2

Cifrado de Información

Juan Fernando De León 17822
Jose Amado Garcia 181469
Edgar Andree Toledo 18439
Ricardo Antonio Valenzuela 18762
Sara Zavala 18893

Introducción

En la actualidad diversos países cuentan regulaciones y legislaciones en las cuales permiten a los proveedores de servicios de internet registrar el tráfico en línea de cada usuario. Esta es una de las razones por la cual la mensajería segura y encriptada se ha vuelto de suma importancia. Si se utiliza el sistema de mensajería SMS estándar y no cifrados estos están abiertos para que proveedores de servicios de telefonía, el gobierno y piratas informáticos vean la información que se maneja en estas aplicaciones.

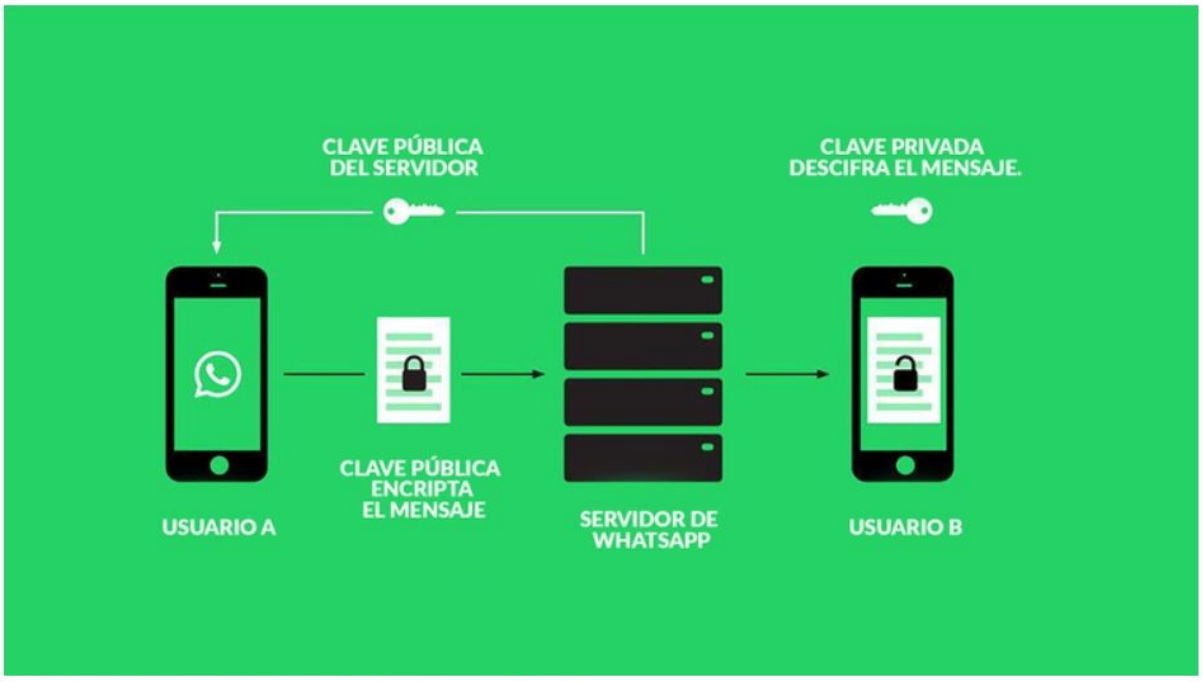
La mensajería cifrada brinda una encriptación *peer to peer,* esto quiere decir que tanto la persona que envía el mensaje y la persona que lo recibe cuentan con una comunicación segura. La mensajería encriptada evita que la información enviada y recibida permanezca secreta e integra. El cifrado de información es un proceso de codificación para evitar que cualquier persona que no sea el destinatario previsto vea el mensaje enviado. En el cifrado de información moderno se utilizan algoritmos donde la información cifrada es ilegible para cualquier persona que no tenga acceso a una clave que se utiliza para descifrar el texto. Dos métodos modernos de cifrado son la clave pública (asimétrica) y la clave privada (simétrica).

Hace poco más de dos años, a principios de abril de 2016, WhatsApp anunciaba la encriptación de extremo a extremo (end-to-end) en las comunicaciones. Esto significa que la información de las conversaciones, en vez de estar almacenadas en los servidores centralizados y gestionados por WhatsApp, son guardadas a través de claves de cifrado en el dispositivo del usuario.

"Este protocolo de cifrado extremo a extremo está diseñado para evitar que terceras partes y WhatsApp tengan acceso al texto plano de las llamadas o los mensajes. Y lo que es más, incluso si las claves de cifrado de un dispositivo de usuario se ven comprometidas

físicamente, no podrán ser usadas para volver atrás en el tiempo y descifrar mensajes transmitidos con anterioridad."

Figura# 1. Descripción del proceso de cifrado de Whatsapp



El funcionamiento del cifrado de extremo a extremo.

Así, en este proceso, existen claves (keys) de cifrado públicas y privadas. Pero, ¿es WhatsApp un sistema 100% seguro?

"Es casi perfecto ya que su seguridad ha sido comprometida en lo que va del año por una investigación realizada por el Perito Informático Marcelo Romero. Encontró un falla en las claves para vulnerar este protocolo de cifrado de extremo a extremo, en la versión web y móvil", señala Emiliano Zárate, investigador digital e instructor en Informática Forense y Seguridad de la Información. De todos modos, en general, los expertos en ciberseguridad consideran que WhatsApp no es fácil de hackear sin acceso al teléfono.

Materiales y métodos

Para la realización de este chat se utilizarán ciertas herramientas de programación relacionadas con el lenguaje JavaScript y Python.

- Visual Studio Code: es un editor de código fuente desarrollado por Microsoft para Windows , Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código

  En este editor se realizará todo el código que construye la estructura de nuestro proyecto
- Django: es un framework de desarrollo web de código abierto. Este se utilizará para el manejo de base de datos y el mantenimiento de un API con el cual se controlará las entradas y salidas con un servidor

  Metodología:

  El proyecto se llevará a cabo por medio de react, javascript y Django. Se creará un proyecto de react donde se manejan los componentes principales para nuestro chat web. Ya teniendo creados cada uno de los componentes, de los cuales están
  - página principal
  - Interfaz de primer usuario
  - interfaz de segundo usuario
  - Manejo de permisos

Teniendo implementados los componentes, pasaremos a la creación de un API propio y/o implementación de uno ya existente. Con esto podremos hacer todas las llamadas necesarias, extracción, y almacenamiento de datos. Teniendo el chat ya estructurado y con el manejo de información estable, podremos comenzar la implementación del cifrado y descifrado de los mensajes y datos enviados por medio de nuestro chat. Para esto se utilizara el lenguaje Python

Todo lo anterior es la descripción de la manera por la cual se va armar la estructura de nuestro chat. Ahora se proseguirá con el detalle de un aspecto crucial, el cual es el cifrado y descifrado de cada uno de los mensajes que se enviará con nuestra estructura.

Para este proyecto se utilizará un método de cifrado de **llave pública** por medio de un sistema **AES** en conjunto con el sistema **RSA.** 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, del resto 0, con lo que ya tendremos un divisor del número.

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 es ser una función computacionalmente segura, ya que si bien realizar la exponenciación modular es fácil, su operación inversa, la extracción de raíces de módulo Ø 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.

Advanced Encryption Standard (AES) es uno de los algoritmos de cifrado más utilizados y seguros actualmente disponibles. Es de acceso público, y es el cifrado que la NSA utiliza para asegurar documentos con la clasificación "top secret". A diferencia de los sistemas tradicionales de cifrado simétrico, RSA trabaja con dos claves diferentes: una pública y una privada. Ambos trabajan complementarios entre sí, lo que significa que un mensaje cifrado con uno de ellos sólo puede ser descifrado por su contraparte. Dado que la clave privada no puede calcularse a partir de la clave pública, ésta está generalmente disponible para el público.

Básicamente utilizaremos el esquema de cifrado y descifrado que utiliza Whatsapp, como se puede ver en la Imagen #1 combinándolo con las herramientas previamente descritas. La única variante que se aplicará a ésa será que toda la información quedará y se manejara en un servidor, no en el dispositivo del usuario.

Resultados y Discusión

El objetivo de este proyecto consiste en crear una aplicación de chats en el lenguaje JavaScript para poder reflejar y dar a conocer cada uno de los conocimientos y estrategias que se tomaron y aprendieron durante el semestre. Al mismo tiempo poder mostrar la importancia de la seguridad informática en algo tan cotidiano como lo es enviar un mensaje de texto a un amigo. Para lograr esto se tomaron algunos sistemas base, como el cifrado y descifrado de mensajes de texto que actualmente la empresa WhatsApp Messenger usa.

La perspectiva gráfica de nuestro chat consiste de un displayer bastante simple donde se muestran los mensajes, en dicho displayer se pueden observar mensajes que se han enviado con anterioridad y aquí mismo aparecerán los mensajes enviados por el usuario, si el mensaje es una imagen se visualiza un preview de esta misma. Así mismo contamos un un input donde se escribirá el mensaje y un botón de enviar para realizar el post al servidor, también se puede enviar el mensaje tras presionar la tecla *Enter*. Como se puede evidenciar en la *Figura 2*, desde el punto de vista gráfico no se puede evidenciar el cumplimiento de nuestro objetivo, ya que en este se muestran los mensajes ya descifrados por el sistema, esto con el fin de ser amigables con el usuario.

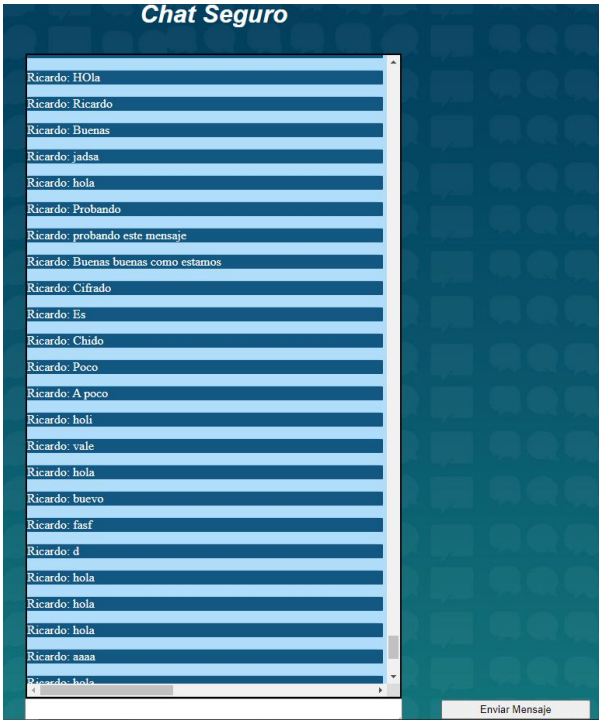


Figura 2. Visualización del Chat

El proceso en donde se envía el mensaje y se guarda en nuestro sistema por medio del API es donde utilizamos los conocimientos obtenidos sobre la utilización de cifrado de clave pública, en nuestro caso estaremos utilizando la implementación RSA en conjunto con el sistema de cifrado AES. El usuario Alice genera las llaves privada y pública por medio del

sistema RSA y encripta el mensaje por medio de AES, la llave pública está conformada por la tupla N y e, siendo N la multiplicación de dos números primos grandes p y q, mientras que e es un número inverso modular con un número d, mientras que la llave privada es conformada por la tupla de N y d. Este mensaje cifrado es mandado al servidor en conjunto con la clave pública generada.

Al momento en que Bob realiza la llamada al servidor lo que obtiene es el mensaje cifrado y la llave pública de Alice, entonces Bob procede a obtener el mensaje descifrado de Alice por medio del uso de su llave privada. De esta manera se completa el ciclo debido a que creamos una conexión segura entre Alice y Bob por medio del uso de llaves privadas y públicas.

Este cifrado se realizó por medio de dos métodos. Primero se utilizaron algunas librerías que JavaScript ya incluye, como podemos ver en la Figura 3. Podemos notar que pedimos como parámetros tanto el mensaje como la llave que se utilizará para el proceso previamente descrito.

```
function encrypt(message = '', key = ''){
    var message = CryptoJS.AES.encrypt(message, key);
    return message.toString();
}
function decrypt(message = '', key = ''){
    var code = CryptoJS.AES.decrypt(message, key);
    var decryptedMessage = code.toString(CryptoJS.enc.Utf8);

    return decryptedMessage;
}
```

Figura 3. Uso de librerías AES

La segunda manera fue una implementación propia. A pesar de que no es recomendable utilizar una manera propia en proyectos más grandes, conseguimos crear métodos y algoritmos con los cuales ciframos y desciframos cada uno de los mensajes de texto. Nos percatamos que a pesar que nuestra implementación se encuentra bastante desarrollada y bien basada, siempre es más eficiente utilizar librerías internas y/o externas ya que estas ya se encuentran verificadas y nos dan mayor seguridad en el ambiente de trabajo.

Literatura Citada

BoxCryptor (2020), Cifrado AES y RSA, Extraido: https://www.boxcryptor.com/es/encryption/

Herramientas Web para la enseñanza , (2020), RSA, Extraido de: https://neo.lcc.uma.es/evirtual/cdd/tutorial/presentacion/rsa.html

Gobierno de España, (2020), Cómo funciona el cifrado en las aplicaciones de mensajería instantánea,Extraído de:https://www.osi.es/es/campanas/copias-cifrado-informacion/cifrado-apps-mensajeria