Encriptación para Discord

Miguel Angel Lama Carrasco

***Abstract*—Este trabajo de investigación presenta un algoritmo para la aplicación Discord, con la intención de encriptar los mensajes enviados en esta plataforma.**

I. Introduction

En los últimos años han surgido una variedad de *exploits* en Discord que han llevado a muchos mensajes a ser expuestos al publico [1]. Debido a esto, hoy en día Discord no se puede considerar un espacio privado o seguro. Por lo que, es necesario desarrollar métodos para ocultar la información para así mantener la privacidad dentro de la plataforma. Con este fin este trabajo de investigación presenta un algoritmo que encripta los mensajes enviados atravez de la plataforma utilizando el método de encriptación asimétrica RSA.

II. Marco teórico

El método RSA genera una llave privada d y una llave publica e y n. Luego, encripta un mensaje m utilizando:

$$c = m^e mod(n)$$

donde c corresponde al mensaje cifrado que se desea enviar. Para desencriptar el mensaje utilizamos la llave privada d que tiene la propiedad de:

$$m = c^d mod(n).$$

Donde n es el producto de dos primos grandes, aleatorios y bien espaciados p y q. Para generar e primero calculamos un $\theta = (p-1)(q-1)$ y luego encontramos un e aleatorio que cumpla las siguientes propiedades:

$$1 < e < \theta$$

$$gcd(\theta, e) = 1$$

Por ultimo, para calcular d utilizamos $de = 1 mod \theta$.

III. Implementación

Primero, para autenticar que un usuario sea realmente quien dice ser, se utilizará Github y Gitlab como servicios externo para almacenar las llaves públicas. Para garantizar la identidad del usuario, realizaremos una verificación presencial antes de autorizar la publicación de su llave pública en la plataforma. De este modo, evitaremos casos de suplantación de identidad.

Luego, desarrollaremos un bot que cada usuario podrá utilizar para encriptar, enviar, recibir y desencriptar mensajes. Este bot funcionará en una aplicación externa a Discord instalada en la computadora del usuario y solo tendrá acceso a la llave privada de la persona que esté utilizando la instancia del bot. Esto facilitará el uso del sistema RSA.

Después, El programa generará una llave privada RSA para cada usuario, con una longitud de dos kilobytes. Esta llave se almacenará de manera local en la computadora del usuario, protegida en un archivo cifrado utilizando el método AES-CBC (Cipher Block Chaining). Para asegurar que solo el usuario pueda acceder a su llave privada, la clave simétrica utilizada para el cifrado AES será generada aleatoriamente y protegida con una contraseña que el usuario definirá. De este modo, se garantiza que las llaves privadas estén a salvo incluso en caso de que alguien acceda a la máquina local.

Para mantener la integridad de los mensajes que se envíen y reciban, se empleará el algoritmo SHA (Secure Hash Algorithm). Antes de encriptar el mensaje con RSA, se generará un hash utilizando SHA. Este hash acompañará al mensaje encriptado y será verificado por el receptor tras desencriptar el mensaje en su computadora, lo que asegurará que el contenido no ha sido alterado durante la transmisión.

Finalmente, todos los mensajes enviados a través de un canal de Discord estarán encriptados con RSA y serán encriptados y desencriptados localmente en la computadora de cada usuario, asegurando la privacidad y seguridad del contenido.

Todo el programa se desarollará en Javascript utilizando Nodejs Express y se publicará como codigo abierto en Github.

IV. Resultados

Se realizaron pruebas de funcionalidad, determinando que es posible: encriptar la comunicación, firmar los mensajes, detectar cambios en las llaves públicas y utilizar Discord como medio de mensajería seguro.

V. Conclusión

Para futuros trabajos, es necesario investigar posibles exploits en JavaScript, las bibliotecas utilizadas y el trabajo propuesto.

Appendix A
Repositorio del trabajo

https://github.com/Skinde/EncriptacionDiscord

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see http://www.michaelshell.org/contact.html).

Revisado por José Carlos Pazos para el curso de ciberseguridad en la Universidad de Ingenieria y Tecnologia.

Enviado Octubre, 2024; Revisado Diciembre, 2024.

References

[1] PrivacyHawk, "Discord's massive data breach: Over 4 billion messages leaked and sold in april 2024." [Online]. Available: https://privacyhawk.com/resources/discords-massive-data-breach-over-4-billion-messages-leaked-and-sold-in/