Hybrid encryption prototype combining AES and RSA encryption methods

Prototipo de cifrado híbrido combinando los métodos de encriptación AES y RSA

Andrés Mauricio Clavijo[1], Jhon Alexander Chacón[2],

Abstract

This article is made with the purpose to present a hybrid encryption prototype implemented over a grid network, with the intention to demonstrate the security level that is possible to achieve with this way of encryption by means of a messaging application; which will be target to analysis with and without the libraries implementation generated in order to compare the results obtained once it is running, likewise there will perform an analysis process over the prototype proposed for verifying possible points of failures. The prototype was developed based on an architecture capable to share resources and functionalities in order to provide the ability to operate with any system that require the use of them, therefore, the messaging application itself is only a test object that give us information about the capacity of the cypher prototype, and aims to provide a confidence level for its implementation in other communication systems.

Keywords: RSA, AES, symmetric encryption, asymmetric encryption, hybrid cryptosystem.

---

[1] Ingeniero en Telemática, Universidad Distrital Francisco José de Caldas, Facultad Tecnológica, Bogotá Colombia. E-mail: amclavijom@correo.udistrital.edu.co ORCID: https://orcid.org/0000-0001-8927-8305

[2] Ingeniero en Telemática, Universidad Distrital Francisco José de Caldas, Facultad Tecnológica, Bogotá-Colombia. E-mail: jhachacont@correo.udistrital.edu.co ORCID: https://orcid.org/0000-0003-2957-3374

Resumen

Este trabajo se realiza con la finalidad de presentar un prototipo de cifrado híbrido, implementado en una red virtual en grid, con el propósito de aprovechar el nivel de seguridad que es posible alcanzar con esta forma de cifrado, por medio de una aplicación de mensajería, la cual será objeto de análisis con y sin la implementación del método de cifrado propuesto, verificando los resultados obtenidos y validando posibles puntos débiles en el proceso de cifrado y transmisión de la información. El prototipo fue desarrollado basado en una arquitectura que le permita compartir recursos y funcionalidades, con el fin de proveer la capacidad de operar en cualquier sistema que requiera el uso de ellos, por lo tanto, la aplicación de mensajería en si es solo un objeto de prueba, que brinda información de la funcionalidad del prototipo de cifrado y del nivel de seguridad alcanzado, para su posible implementación en otros sistemas de comunicación.

Palabras clave: RSA, AES, cifrado simétrico, cifrado asimétrico, criptosistema hibrido.

1. Introducción

Las técnicas que permiten la protección de la información en entornos de comunicación no seguros, como internet, han ido evolucionando en sus niveles de complejidad, generando niveles de seguridad crecientes para la transferencia de información, sin embargo, se han encontrado falencias, ya sea a nivel del algoritmo criptográfico, en el canal de comunicaciones o en la implementación de dicho algoritmo, lo que exige el avance o surgimiento de nuevos métodos de cifrado seguros.

El desarrollo del prototipo de cifrado, se fundamenta en el uso del cifrado simétrico y asimétrico, con el fin de aprovechar las ventajas y disminuir las desventajas y/o vulnerabilidades de estos en una misma solución; este proceso es conocido como cifrado híbrido.

Este método consiste en proteger la llave privada del método de cifrado AES [1], de aprovechar la ventaja del algoritmo AES en la velocidad de cifrado de grandes cadenas de datos, y las ventajas del cifrado asimétrico en la distribución de llaves. Es necesario tener en cuenta que se requiere más capacidad computacional y/o de tiempo de procesamiento para llevar a cabo las tareas de cifrado con el fin de aumentar la complejidad y la seguridad en la información.

Se busca disminuir las debilidades del cifrado simétrico, mediante la encapsulación por parte del cifrado asimétrico [2]; este método de encapsulamiento permite reducir las restricciones en cuanto a longitud de mensajes, sin afectar la seguridad del algoritmo, implementando sistemas que utilicen las ventajas del funcionamiento del cifrado hibrido, y haciendo posible la sustitución de las llaves usadas en la comunicación, por unas llaves nuevas para el cifrado de la información, cuando se realiza el proceso de transmisión de la información.

Cifrado simétrico: Este tipo de cifrado tiene como propiedad fundamental el uso de una clave privada, que se genera para realizar el proceso de cifrar y descifrar la información o datos que se transmiten. Los algoritmos criptográficos simétricos tienen dos versiones, cifrador en bloque y cifrador en flujo. Los cifradores en bloque codifican datos en bloques pequeños de longitud fija de 64 bits de longitud; hay distintos tipos de cifradores en bloque, que se utilizan por los algoritmos de cifrado DES, 3-DES, RC2, RC5, RC6 y Rijndael (AES) [3]. Este tipo de cifrado es el utilizado en el desarrollo del prototipo.

Cifrado asimétrico: Este tipo de cifrado es mucho más complejo que el simétrico, ambos tienen la funcionalidad de tratar la información para que no pueda ser interceptada por terceros, pero a diferencia estos tipos de sistemas, utilizan dos claves, una privada y una pública. Ambas pueden ser usadas para cifrar y descifrar información; dichas claves están matemáticamente relacionadas entre sí, la clave pública está disponible para todos y la clave privada es conocida solo por el emisor. Existen varios algoritmos de cifrado asimétrico muy utilizados como Diffie-Hellman, RSA, DSA [4].

Cifrado hibrido: Es un algoritmo que combina dos o más algoritmos que resuelven el mismo problema, ya sea eligiendo uno (según los datos o información a cifrar) o cambiando entre ellos a lo largo de la implementación del algoritmo. Esto generalmente se hace para combinar las características deseadas de cada uno, de modo que el algoritmo general sea mejor que los componentes individuales [5].

Computación en grid: La computación en grid (en malla), ha surgido como un importante campo de la computación; contrario a otras arquitecturas, donde el enfoque es crear computadoras de gran rendimiento, medido en número de cálculos de punto flotante por unidad de tiempo, la importancia de la computación grid está definida en términos de cantidad de trabajo que puede despachar en una unidad de tiempo. Esta tecnología no es revolucionaria, se puede decir que ha evolucionado de otras tecnologías como los sistemas distibuidos, la virtualización, los servicios web, el internet y la criptografía [6].

Servicios REST: Los servicios REST (Representational State Transfer) es un estilo de arquitectura para desarrollar servicios; los servicios web basados en este estilo de arquitectura cumplen las siguientes premisas: Cliente/Servidor: Los servicios web cliente/servidor definen

una interfaz de comunicación entre ambos, separando completamente las responsabilidades entre ambas partes; Sin estado: Son servicios web que no mantienen estado asociado al cliente; cada petición que se realiza a ellos es completamente independiente de la siguiente, todas las llamadas al mismo servicio serán idénticas [6].

WebSockets: Los WebSockets fueron introducidos con la llegada de HTML5, y es una de las mejoras más utilizadas. Estos permiten trabajar de forma bidireccional entre el navegador y el servidor, para enviar y recibir mensajes de forma simultánea (Full Duplex), manteniendo siempre una conexión activa con el servidor mediante Sockets TCP [7].

Los autores Avinash, R., Potnis, A., Kumar, S., Dwivedy, P., y Soofi, S presentan un modelo de encriptación hibrido, iniciando con la encripción con RSA, seguido por el algoritmo AES, y finaliza con la operación XOR. Se realiza el proceso de los datos con los diferentes algoritmos; con el algoritmo AES se usa un tamaño de llave de 128 bits (10 rondas) [8]. Los resultados experimentales del modelo propuesto por los autores muestran que el cifrado tiene un rendimiento variable, esto debido a los resultados en la medición de parámetros relacionados con su nivel de complejidad. Así mismo, Anane, M., y Nadjia, A, proponen un modelo basado en componentes electrónicos para la encriptación basada en el algoritmo de cifrado AES, ya que según los autores, el proceso de cifrado necesita ser acelerado por hardware a fin de brindar mejoras en el desempeño. También se plantea el desarrollo de un software para realizar la encriptación de la llave del algoritmo AES sobre RSA, adicional a la generación de RoundKeys del algoritmo AES, la cual es realizada solo al inicio de las tareas de cifrado y descifrado [9]. Los resultados de la aplicación de este sistema, muestran que la optimización realizada a nivel de hardware permite mejorar la eficiencia, cuando se trata de ejecutar

procesos de cifrado, proporcionando tiempos de respuesta óptimos, que permiten un aumento en el rendimiento de la ejecución del algoritmo de cifrado AES.

2. Metodología

Este trabajo tiene como propósito el desarrollo de un prototipo de cifrado hibrido, mediante la combinación de los algoritmos de encriptación AES y RSA, pretendiendo aprovechar los niveles de seguridad que poseen cada uno de ellos, por medio de su implementación en una aplicación de mensajería, en una virtualización de un sistema de red en grid. Se plantea el uso de los algoritmos AES y RSA para generar un método de cifrado híbrido, capaz de proveer un nivel de seguridad adecuado en la transmisión de información a través de internet. La Figura 1, presenta la implementación del modelo de cifrado hibrido.

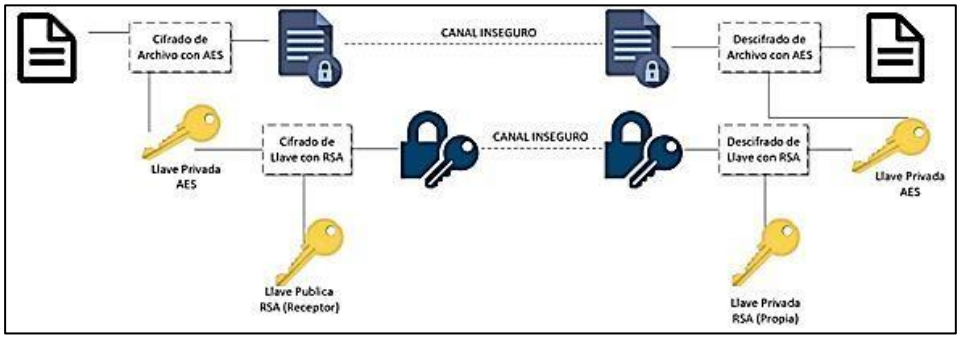

Figura 1. Modelo de cifrado hibrido. Fuente: propia.

Computacionalmente, la complejidad en estos métodos de cifrado híbrido se presenta en el proceso de generación de las llaves, debido a que requiere ejecutar métodos matemáticos, que cumplan las condiciones del cifrado simétrico y asimétrico, y que cuando se utilicen métodos de criptoanálisis como el método de fuerza bruta, los datos sean casi imposibles de descifrar, ya sea que se pretenda alterarlos o robar de información, cuando se encuentran las llaves de descifrado. Para este prototipo, se realiza la implementación de una aplicación de

escritorio y de un entorno virtual en grid. Con el fin de validar su funcionalidad, se plantea el uso de llaves de longitud adecuada para-AES y RSA. Para el caso del algoritmo AES, se establece una longitud de llave privada de 256 bits, 14 rondas, de acuerdo a lo planteado en [10]; esta longitud provee una adecuada protección, como se presenta en la tabla 1.

Nombre	Longitud de la clave	Longtiud del bloque	Nivel de seguridad
DES	56	64	56
triple DES de dos llaves	112	64	95-100
triple DES de tres llaves	168	64	112-116
DESX	120	64	120
IDEA	128	64	128
AES-128	128	128	128
AES-192	192	128	192
AES-256	256	128	256

Tabla 1. Niveles de seguridad para algoritmos comunes de cifrados en bloque. Fuente: https://infoscience.epfl.ch/record/164539/files/NPDF-32.pdf

Para el algoritmo RSA, se establece que una longitud de llaves de 2048 bits, y de acuerdo a la tabla 2, es su

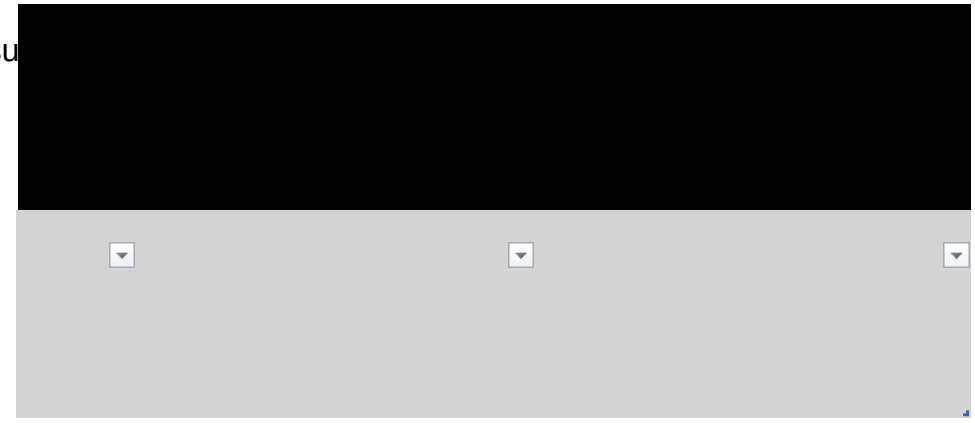

Tabla 2. Proyección de seguridad en el tiempo según la longitud de las llaves en bits. Fuente: https://infoscience.epfl.ch/record/164539/files/NPDF-32.pdf

Para los procesos de firmado y verificación de la información, se plantea el uso de las llaves generadas para el cifrado del algoritmo RSA, con el fin de aprovechar los recursos computacionales usados en el proceso de generación de llaves y cifrado, a pesar de no ser el algoritmo de firmas más rápido y sencillo [12]; también se implementa el algoritmo hash SHA-512, para alcanzar un mensaje suficientemente corto y fácil de procesar por el emisor y receptor, sin dejar a un lado la seguridad que este provee. Teniendo en cuenta las diferentes librerías utilizadas para el desarrollo del prototipo en lenguaje .Net, se elige la versión de núcleo 4.7.2. La librería AesCryptoServiceProvider, permite realizar el uso del algoritmo AES de manera correcta y la generación del Vector de Inicialización (IV) de forma rápida y segura, los parámetros utilizados para la implementación del algoritmo AES en el prototipo se observan en la tabla 3. Para la implementación del algoritmo RSA, se utiliza la librería RSACryptoServiceProvider, la cual, con una longitud de bits particular, genera un par de llaves, la llave pública y la llave privada, para su uso en los procesos de cifrado, descifrado y firmado.

Tamaño de la llave privada en bits	Tipo de Relleno	Modo de cifrado en bloque	Vector de inicialización
256	PKCS7	CBC	Aleatorio

Tabla 3. Parámetros establecidos para el proceso de cifrado con el algoritmo AES. Fuente: propia.

3. Arquitectura planteada

Para el desarrollo del servicio de cifrado y descifrado, se tienen en cuenta las acciones que los usuarios pueden realizar en el prototipo, para así poder establecer una comunicación segura. Es importante realizar la segmentación por diferentes servicios, dado que la información no puede emitirse desde el origen sin ser procesada. En este caso, el envío de un texto cifrado, inicia el procedimiento a servicios locales y remotos. La figura 2, muestra el proceso planteado y los pasos a realizar en el prototipo para cifrar, enviar y capturar (mostrar) un mensaje de manera segura y adecuada.

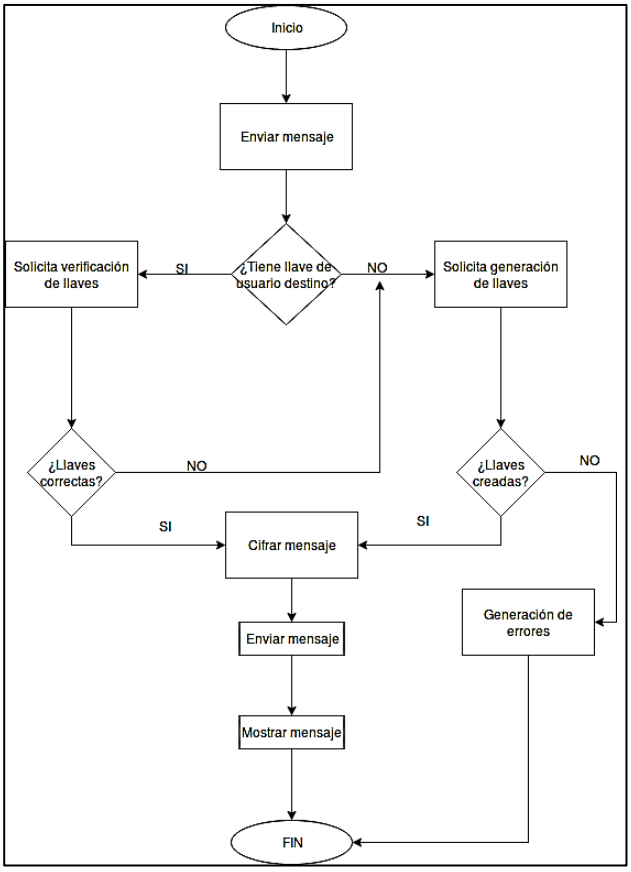

Figura 2. Diagrama de flujo del mensaje

Después de comprender el funcionamiento lógico del prototipo, se plantea un escenario de comunicación, representado por el diagrama de secuencia en la figura 3, en el cual existen dos usuarios, el emisor y el receptor, quienes iniciaran una comunicación, y un servidor el cual se encarga de realizar los respectivos procesos de direccionamiento y transmisión de la información. El emisor es quien realiza la petición al servidor para iniciar una conversación con el receptor, y cuando ocurre la primera llamada, el servidor comienza a realizar las peticiones entre ambos usuarios, para realizar los procesos de generación e intercambio de llaves, permitiendo que los usuarios puedan enviar mensajes de un punto al otro de manera cifrada y segura.

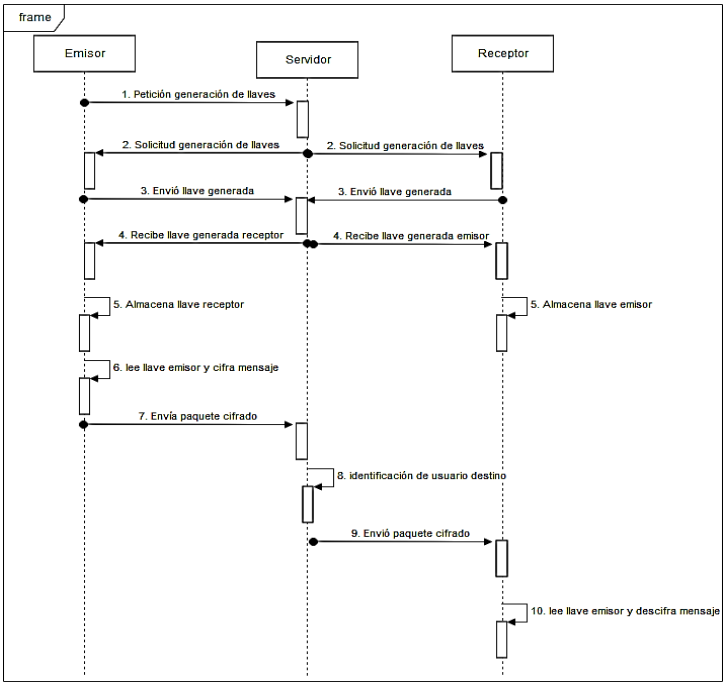

Figura 3. Diagrama de secuencia del prototipo. Fuente: propia.

Desarrollo del Prototipo: Los diferentes componentes para el desarrollo del entorno y aplicación, se realizan sobre un ambiente virtual con dos hosts, con el fin de simular la intercomunicación real y la transmisión de información.

Entorno virtual: Para la implementación del ambiente virtual se utiliza el software de virtualización Vmware Player Workstation, donde se realiza la creación de tres servidores virtuales (dos servidores de aplicaciones y un servidor BOINC), con el fin de simular el ambiente en grid entre dos terminales, que harán las veces de host, creando una máquina virtual (VM) del servidor de aplicaciones en cada uno de ellos, como se observa en la figura 4.

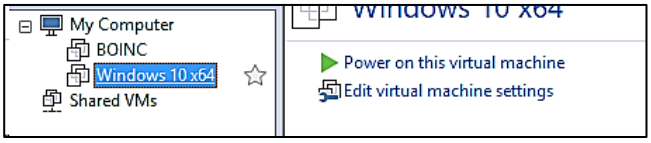

Figura 4. Máquinas virtuales creadas en un host. Fuente: propia.

Implementación de la red en malla: Para la creación del prototipo de red en malla, se hace uso del software con licencia de código abierto BOINC, el cual permite la creación y unión a entornos de red en malla, con el fin de hacer uso de los recursos de cada máquina unida al proyecto, y la ejecución de determinadas tareas, como se observa en la figura 5.

Figura 5. Proyecto Cifrado Híbrido en BOINC. Fuente: propia.

Servicio de cifrado híbrido: Los servicios de cifrado y descifrado se desarrollan en el lenguaje .Net core y Java con servicios Rest. El servicio de cifrado se diseña para trabajar de forma local en cada uno de los usuarios, y el servicio de descifrado para uso remoto (servidor); estos usuarios se comunican a través del servidor para realizar la distribución de llaves y establecer la comunicación entre ellos, como se presenta en la figura 6.

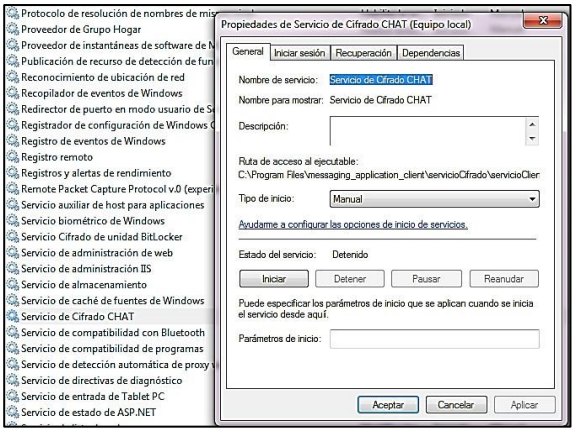

Figura 6. Propiedades del Servicio de Cifrado CHAT. Fuente: propia.

Aplicación de mensajería: El componente de mensajería es desarrollado para utilizar los servicios creados del módulo de cifrado. El desarrollo se realiza con ReactJs, el cual provee un rendimiento adecuado y permite generar aplicaciones intuitivas; luego se hace uso de Electrón para generar una aplicación de escritorio, que permita instalarse y utilizarse en el ambiente de desarrollo propuesto, como se observa en la figura 7.

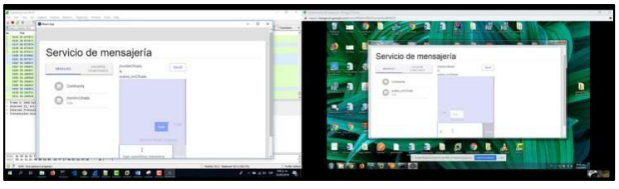

Figura 7. Servicio de mensajería entre dos clientes. Fuente: propia.

4. Pruebas del prototipo

Inicialmente, el servicio de mensajería se utiliza sin el prototipo de cifrado sobre el ambiente en grid, realizando la comunicación de los clientes; pero una vez se realiza la interceptación de los paquetes, mediante la herramienta de monitoreo (sniffer) Wireshark, se evidencia que la información es fácil de interpretar, debido a que esta viaja en texto claro, lo cual es considerado potencialmente una vulneración para la seguridad de la información, cómo se evidencia en la Figura 8.

Figura 8. Captura de mensajes sin prototipo de cifrado. Fuente: propia.

Ahora, cuando se implementa el prototipo de cifrado, se validan los paquetes capturados y los mensajes cifrados en las figuras 9 y 10, por medio del escaneo de paquetes sobre el servicio de mensajería. Se evidencia que no es posible interpretar fácilmente el mensaje que se está transmitiendo, debido al método de cifrado híbrido implementado.

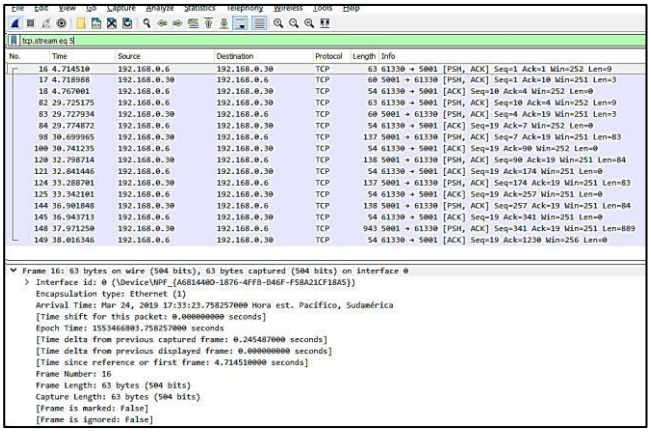

Figura 9. Escaneo de paquetes. Fuente: propia.

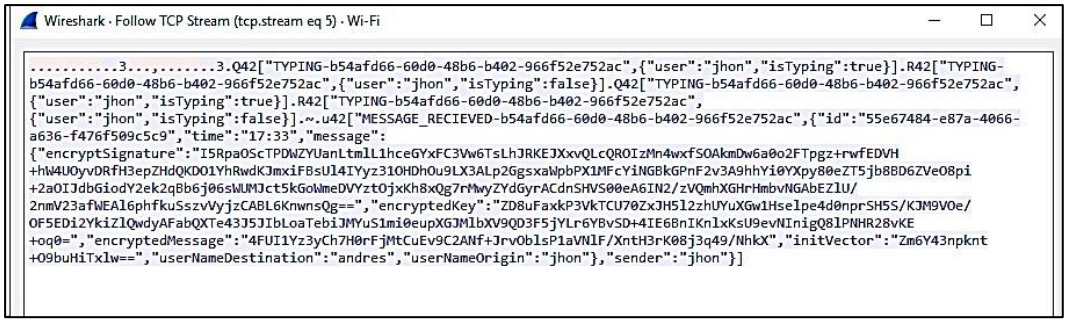

Figura 10. Captura de mensajes con implementación del prototipo de cifrado. Fuente: propia.

5. Resultados y Discusión

Una posible vulneración sobre el algoritmo RSA mostraría en claro la llave del proceso AES, permitiendo descifrar la información enviada; pero que este proceso de vulneración sea exitoso una vez, no garantiza que las próximas transmisiones de mensajes puedan ser descifrados con la misma llave encontrada, debido a que estas llaves se generan nuevamente cada vez que se envía la información. Si en efecto, la información fue descifrada, y el atacante altera los datos e intenta engañar al receptor con paquetes corruptos, el sistema doble de verificación de integridad, proporcionado por las firmas digitales del algoritmo RSA y el Vector de

Inicialización del algoritmo AES, permite identificar que la información fue alterada por un tercero, y por lo tanto no se presenta en la interfaz de usuario (UI), registrándose internamente como un origen de información no válido. Se genera un prototipo de cifrado que puede ser integrado a un servicio de comunicación que requiera seguridad en el envío de los datos, por medio de servicios de mensajería, sin la necesidad de generar certificados de seguridad. Los análisis a los sistemas criptográficos usados para la implementación del modelo híbrido, aún permiten determinar un nivel adecuado de eficiencia y seguridad, según estudios realizados.

6. Conclusiones

El prototipo planteado presenta una buena funcionalidad, pero puede verse afectado al realizar análisis de vulnerabilidad más profundos y extensos, realizados con métodos y herramientas de criptoanálisis potentes.

Se pueden implementar distintos métodos y alternativas para la protección de información, ya sea de forma personal o corporativa. Aunque existen herramientas de software que permiten la implementación de los algoritmos de cifrado clásico, es importante proponer y probar distintos métodos e ideas, para que las nuevas tecnologías sean fáciles de integrar en soluciones de comunicaciones ágiles, actuales y seguras.

La utilización de distintas herramientas que permitan que la información pueda cifrarse, con el fin mantener la seguridad y confidencialidad de los datos, juegan un papel importante y relevante, ya que la disponibilidad y variedad de sistemas de comunicación cada día son mayores, por lo que la implementación de criptosistemas, pueden prevenir alteraciones, suplantaciones y robo de la información de distinta índole.

El avance en materia de seguridad de la información es grande, pero también aumentan las formas de ataque a los sistemas de cifrado de la información, por lo que el estudio y análisis constante de nuevos métodos de cifrado, que permitan la mejora de seguridad en la información, es vital para el desarrollo social y tecnológico.

Referencias

[1] A. Al Hasib, «A Comparative Study of the Performance and Security Issues of AES and RSA Cryptography,» de Convergence Information Technology, International Conference, Finlandia, 2008.

[2] Shamir R.L. Rivest and L. Adleman, (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Magazine Communications of the ACM, 1978.Volumen 21 págs. 120–126. https://doi.org/10.1145/359340.359342

[3] Castro Lechtaler, A., Cipriano, M., García, E., Liporace, J., Maiorano, A., Malvacio, E. and Tapia, N., (2021). Estudio de técnicas de criptoanálisis.XXI Workshop de Investigadores en Ciencias de la Computación. [online] Sedici.unlp.edu.ar. Available at: http://sedici.unlp.edu.ar/handle/10915/77269

[4] J. C. Mendoza T, «Universidad Politecnica Salesiana de Ecuador,» [En línea]. Available: https://dspace.ups.edu.ec/bitstream/123456789/8185/1/Demostraci%C3%B3n%20de%20cifrado%20sim%C3%A9trico%20y%20asim%C3%A9trico.pdf.

[5]A. W. Dent, «Hybrid Cryptography,» 3 Junio 2009. [En línea]. Available: https://eprint.iacr.org/2004/210.ps

[6] Escobar Molero Gabriel. (2011). Clúster de alto rendimiento en un cloud: ejemplo de aplicación en criptoanálisis de funciones hash. Universidad de Almería. pg 60. http://repositorio.ual.es/bitstream/handle/10835/1202/PFC.pdf?sequence=1

[7] A. Pousa, «Universidad Nacional de la Plata,» Diciembre 2011. [En línea]. Available: https://postgrado.info.unlp.edu.ar/wp-content/uploads/2014/07/Pousa_Adrian.pdf.

[8] A. Lenstra, «Key Lengths,» [En línea]. Available: https://infoscience.epfl.ch/record/164539/files/NPDF-32.pdf.

[9] R. Avinash, A. Potnis, S. Kumar, P. Dwivedy y S. Soofi, «Internation Journal Of Engineering Research and Applications,» Agosto 2017. [En línea]. Available: http://www.ijera.com/papers/Vol7_issue8/Part-1/O0708019094.pdf

[10] A. Faget, «What are Cryptographic Signatures? | Introduction to the Most Common Schemes,» 14 Noviembre 2018. [En línea]. Available: https://coindoo.com/what-are-cryptographic-signatures-introduction-to-the-most-common-schemes/.

[11] Goldreich, O. (2000). Modern Cryptography, Probabilistic Proofs and Pseudorandomness (Second Edition - author's copy). Springer.pag 1-2, consultado en http://www.wisdom.weizmann.ac.il/~oded/PDF/mcppp-v2.pdf

[12] Muñoz, R., Muñoz, R., & completo, V. (2021). Algoritmo RSA en aplicación web. Retrieved 12 July 2021, from http://criptografiaverm1.blogspot.com/2013/07/tarea-5-algoritmo-rsa-en-aplicacion-web.html

[13] Eslava Blanco, H. J., Rocha, J. F., & Morales, J. I. (2011). Estudio de tráfico sobre una plataforma de virtualización. Visión electrónica, 5(2), 78-94. https://doi.org/10.14483/22484728.3572