Enfoque UTE, V.7-Sup.1, Feb.2017, pp.160 - 171
http://ingenieria.ute.edu.ec/enfoqueute/
e-ISSN: 1390-6542 / p-ISSN: 1390-9363
Recibido (Received): 2017/01/09
Aceptado (Accepted): 2017/02/24
CC BY-NC-ND 3.0

Seguridad de la información en el intercambio de datos entre dispositivos móviles con sistema Android utilizando el método de encriptación RSA

(Information security in data exchange between mobile devices with Android system using RSA encryption)

Fernando Solís [1, 2], Diego Pinto [2], Santiago Solís [1]

Resumen:

Los nuevos estilos y formas de vivir conllevan al mayor uso de redes inalámbricas, siendo el dispositivo móvil una herramienta para la transmisión de datos, los cuales son susceptibles a las amenazas en los canales de transmisión en la red. La seguridad informática cumple un papel muy importante para garantizar la disponibilidad, privacidad e integridad de la información, una de las técnicas que ayuda en ésta tarea es la criptografía, cuyo fundamento es transformar un mensaje de modo que sea inentendible salvo para los que posean la clave para descifrarlo. La investigación se enfoca en el uso del algoritmo RSA entre dispositivos móviles, los datos cifrados se envían por canales de comunicación llamados hilos que mediante fórmulas y procesos ejecutados en el servidor, ayudarán a ejecutar el cifrado y descifrado de los datos. Para llevarlo a cabo se diseñó e implementó un prototipo para el intercambio de datos entre dispositivos móviles de manera inalámbrica, efectuando pruebas de rendimiento con tres nodos con el fin de mejorar la seguridad. Los resultados muestran la eficiencia del algoritmo y adicionalmente su funcionalidad, los tiempos de cifrado y descifrado son rápidos frente al envío de información sin ningún método o algoritmo utilizado.

Palabras clave: seguridad informática; criptografía; RSA; dispositivos móviles; sistema Android

Abstract:

The new styles and ways of life lead to greater use of wireless networks, the mobile device being a tool for data transmission, which are susceptible to threats in the transmission channels in the network. IT security plays a very important role in guaranteeing the availability, privacy and integrity of information, one of the techniques that helps in this task is cryptography, whose foundation is to transform a message so that it is unintelligible except for those who have the Key to decipher it. The research focuses on the use of the RSA algorithm between mobile devices, the encrypted data is sent through communication channels called threads that through formulas and processes executed on the server, will help to execute the encryption and decryption of the data. To carry it out, a prototype for the exchange of data between mobile devices wirelessly was designed and implemented, conducting performance tests with three nodes to improve the security. The results show the efficiency of the algorithm and additionally its functionality, the times of encryption and decryption are fast against the sending of information without any method or algorithm used.

Keywords: IT security; cryptography; RSA; mobile devices; android system

[1] Instituto Superior Rumiñahui - ISTER, Sangolquí – Ecuador (fsolis@ister.edu.ec, wsolis@ister.edu.ec)
[2] Universidad de las Fuerzas Armadas – ESPE, Sangolquí – Ecuador (efsolis@espe.edu.ec, djpinto@espe.edu.ec)

1. Introducción

La movilidad es una de las ventajas que ofrecen las redes inalámbricas (Karygiannis & Owens, 2002), pero sus características presentan problemas de seguridad en comparación con las redes tradicionales (Barajas, 2004). Una de las desventajas en las redes inalámbricas es que cualquier dispositivo que se encuentre dentro del rango de la señal puede hacer uso de ella. El uso de tecnología inalámbrica incrementa la posibilidad de intercambiar información, además aumenta la inseguridad en la transmisión de los datos, es por eso que protegerla se ha vuelto una prioridad.

Se propone construir un prototipo para el intercambio de datos entre dispositivos móviles en una red inalámbrica, para ello se implementa el método de encriptación RSA (Cipriano, 2008), se analiza el uso del protocolo de negociación del estándar TLS (seguridad de la capa de transporte), además el uso de sockets permitirá establecer la conexión entre el cliente y el servidor.

Para asegurar la transmisión de datos desde un smartphone con sistema android (Inc., 2016) al servidor, se encripta la información para su viaje en los canales de comunicación a fin de garantizar su integridad (Shabtai, Fledel, Kanonov, Elovici, Dolev, & Glezer, 2010).

El cifrado se realiza mediante un proceso instalado en el dispositivo móvil. El mensaje cifrado obtenido mediante el algoritmo RSA y la clave privada del servidor, viajan por el hilo de comunicación hasta el servidor local. Una vez que llega al servidor es descifrado mediante la llave privada, y es cifrado nuevamente con la llave pública del usuario destino, cuando llega al usuario receptor el descifrado se realiza utilizando el método RSA inversa.

Durante el proceso de cifrado y descifrado es necesario acceder a los datos reservados para el método, y a continuación mostrar el mensaje descifrado (Meza, 2010). Este algoritmo ha sido desarrollado utilizando como software: Android Studio, mismo que genera una aplicación la cual se ha instalado en los dispositivos móviles utilizados. A fin de incrementar el nivel de seguridad, se manejan y almacenan las claves en los hilos de comunicación los cuales se activan siempre que el servidor lo requiera. Este algoritmo proporciona un alto nivel de seguridad por el hecho de utilizar números primos muy grandes y distintos, los números forman la clave privada y pública de cada una de las partes, otro aspecto a considerar es la factorización del producto de dichos números, el algoritmo RSA fija que como mínimo los números primos sean de 155 dígitos (512 bits), asegurando la integridad y la confidencialidad de los datos (Young & Young, 2006) (Sierra & Lerch, 2014).

El uso de protocolos de seguridad y métodos de encriptación dependen su funcionamiento del método matemático con el cual fueron elaborados (Pino & Hernández, 2000), permitiendo el cifrado y descifrado de la información, esto conlleva al continuo desarrollo y mejoramiento de la seguridad de la información para garantizar la confidencialidad, disponibilidad y autenticidad (Scolnik, 2014); actualmente los métodos producen excelentes resultados reflejados en el uso de recursos computacionales.

2. Materiales y Métodos

Se desarrolló un prototipo utilizando la arquitectura cliente-servidor la cual utiliza varios nodos, un servidor (computadora portátil) que se comunicará con los dispositivos móviles (clientes), los cuales establecen una negociación con el uso del protocolo TLS, con la utilización de socket's que servirán en la conexión entre el servidor y los clientes y poder intercambiar datos mediante el uso de una red inalámbrica. Para poder establecer una sesión de intercambio de datos es necesario que la APP generada se instale en los dos dispositivos con la configuración adecuada, y conectar los tres dispositivos a la red inalámbrica Ad Hoc.

Se utilizó Agile Unified Process (AUP) (Kruchten, 2004) (Ambler, 2002), ya que tiene el enfoque del desarrollo de software del proceso unificado, añadiendo un proceso mucho más rápido, serializando procesos detalladamente y generando versionamiento de la aplicación de acuerdo a las diferentes etapas de desarrollo (Li & Wang, 2010). La metodología utilizada en el proyecto es ágil, ya que actualmente el proceso de elaboración debe optimizar el uso de recursos tangibles e intangibles, por ejemplo el tiempo utilizado para la elaboración del proyecto.

2.1 Diseño del sistema

El prototipo está compuesto por dos partes: el módulo de negociación y el modulo para intercambio de datos, teniendo en cuenta que los procesos variarán dependiendo del dispositivo en el que se ejecuten.

La *Figura 1* muestra que una vez que se ha establecido la red inalámbrica se realizará la conexión del servidor y los clientes asignando sus respectivas direcciones IP, luego se levantan los servicios y enrutamiento a nivel de servidor, y se ejecutan el cifrado y descifrado usando como parámetros los números primos generados que servirán para garantizar la integridad, confidencialidad y autenticidad de los datos, utilizando el protocolo TLS para establecer una sesión segura.

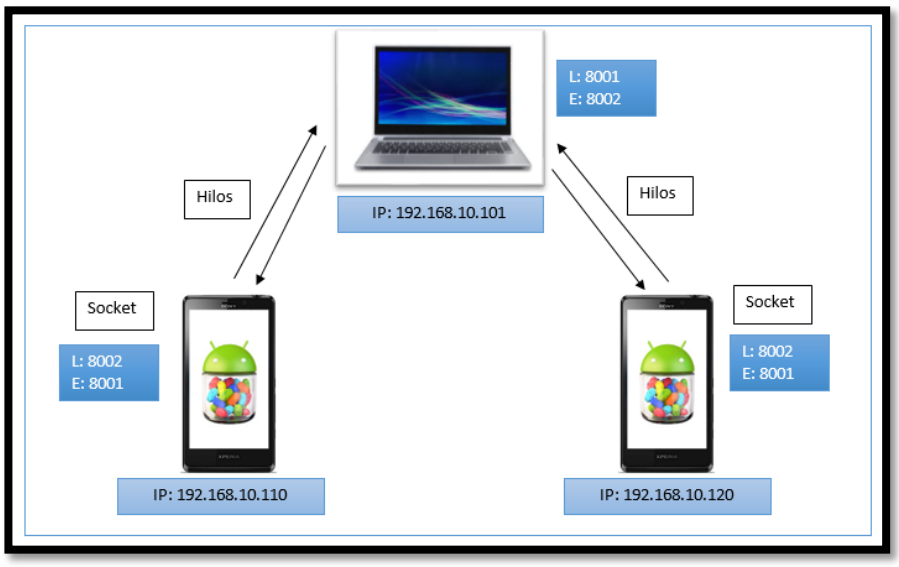

Figura 1. Funcionamiento del sistema

El propósito de la utilización de llaves de sesión es para entablar la negociación entre los dispositivos y el servidor, el módulo de intercambio de datos se ejecuta mediante la utilización de hilos de comunicación que funcionan de manera bidireccionales. Adicional a esto se hace uso de la interfaz MAP en java (Cadenhead, 2014) (Montero Miguel, 2014) que permite representar una estructura de datos para almacenar pares “clave/valor”, en el prototipo se almacenará “Dirección IP / Hilo de envío”.

El prototipo de mensajería requiere que los dispositivos móviles y el servidor de aplicaciones mantengan intercambio de información, para realizar ésta comunicación se utiliza un formato de intercambio de datos denominado JSON (*JavaScript Object Notation*) (Downes, Belliveau, Samet, Rahman, & Savoie, 2010) , una de las mayores ventajas del formato es que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías.

2.2 Mapeo de comunicaciones

Para la implementación del prototipo se utilizó uno de los patrones de desarrollo de software más utilizados, el Modelo – Vista – Controlador (MVC), este patrón se basa en separar los datos y la lógica de negocio de la aplicación de la interfaz del usuario (Buschmann, Meunier, Rohnert, Sommerlad, & Stal, 1996). Para ello MVC define componentes para la representación de la información, y por otro lado para la interacción del usuario. MVC está diseñado para facilitar la tarea del desarrollo de aplicaciones mediante la reutilización de código y la separación de conceptos, a la vez que contribuye en gran medida al mantenimiento y escalabilidad de dichas aplicaciones.

Para la aplicación del método de encriptación RSA se utilizó programación en java, usando la técnica de programación que facilite el intercambio de información entre el código en java y los dispositivos móviles. Para la lectura de las llaves y el hilo de comunicación de dispositivos se aplica Stream, ya que al momento de conectarse el cliente este almacenará en la llave una dirección IP y el hilo que se generará a continuación con cada uno de los dispositivos que realicen la petición al servidor de aplicaciones.

2.3 Diagrama de secuencia

La *Figura 2* muestra el proceso de cifrado de los mensajes que se envían desde un smartphone hacia otro teniendo en cuenta los actores y plataformas involucradas directamente, es necesario describir el levantamiento de los servicios en el servidor y consecuentemente en la aplicación, con esto se ejecuta la conexión y el proceso de los hilos de comunicación que servirá de canal de envío y recepción de los mensajes para ejecutar la programación establecida tanto en el servidor como en el celular el método de cifrado del RSA.

2.4 Descripción del sistema

En la *Figura 3* se observa el proceso que se realiza al momento de enviar un mensaje del cliente 1 al cliente 2. El cliente 1 envía un mensaje (m), dicho mensaje debe estar encriptado para poder

viajar por los hilos de conexión, es por eso que se lo multiplica por la clave pública del servidor (Kps) con lo que se obtiene el mensaje cifrado (M). M viaja hasta llegar al servidor en donde es descencriptado, para que el servidor pueda conocer cuál es el mensaje (m) se debe multiplicar el mensaje cifrado (M) por la clave privada del servidor. Para continuar con el envío el servidor debe encriptar nuevamente el mensaje (m) multiplicándolo por la clave pública (kp2) obteniendo así el mensaje encriptado (N) que será el que viaje por los hilos de conexión hasta llegar al destino del mensaje (cliente 2). El destinatario (cliente 2) para ver el mensaje original debe multiplicar el mensaje encriptado (N) por la clave privada (Kp2) obteniendo así la información requerida.

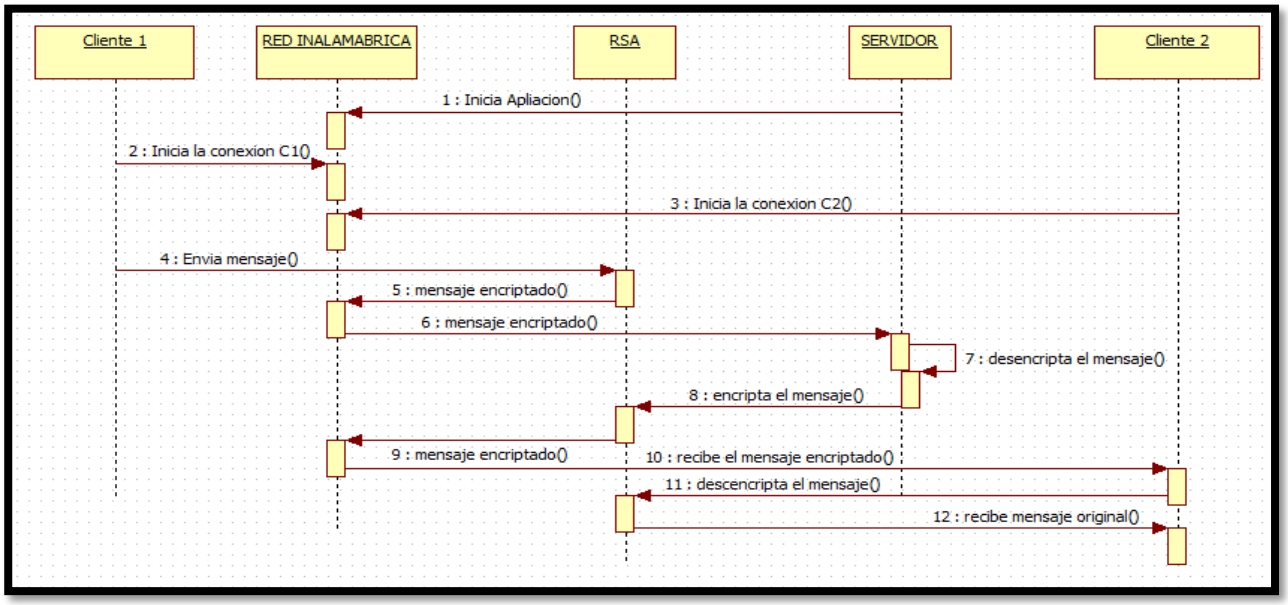

Figura 2. Proceso de encriptación

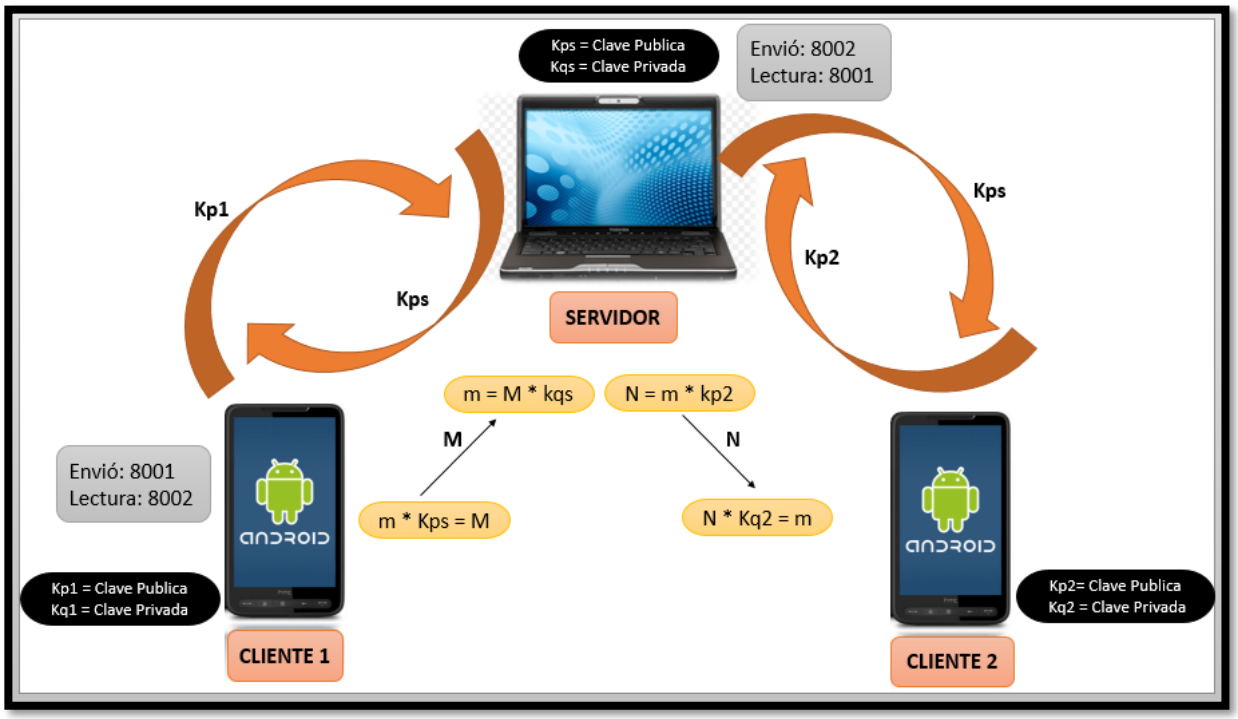

Figura 3. Proceso del sistema

2.5 Desarrollo del prototipo

La aplicación móvil se desarrolló usando como herramienta Android Studio versión 4.1 KitKat para la compatibilidad de los smartphones. En la *Figura 4* se observa la interfaz cuando el usuario se conecta al sistema, misma que funciona como página principal, dentro de este escenario aparecerán los iconos que permitirán el envío y recepción de los mensajes.

Figura 4. Aplicación en el terminal del cliente

La *Figura 5* muestra el escenario para registrar nuevos contactos a partir de la dirección IP que se almacenará en la base de datos predeterminada del dispositivo.

Figura 5. Registro de contactos

EL software del servidor de aplicaciones se desarrolló con la herramienta Eclipse versión Kepler con JDK 1.7 donde se encuentra el módulo de cifrado y descifrado como se observa en la *Figura 6*, la tecnología utilizada es Jax-rs para servicios resful en java para web, la ventaja del uso de

ésta tecnología es que las aplicaciones son multiplataforma y se ejecutan sobre cualquier ambiente.

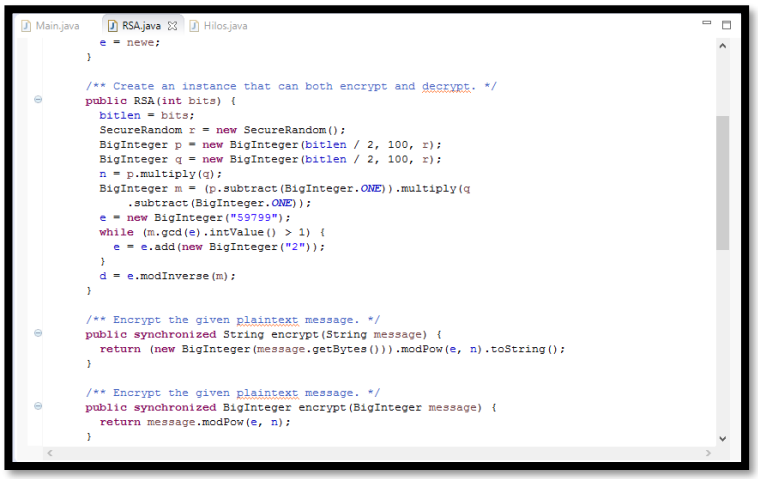

Figura 6. Aplicación del servidor

En la *Figura 7* se observa que el componente al no ser controlado por un actor humano sino por el sistema denominado ServidorMensa no tendrá interfaces gráficas, pero se mostrará en la consola las peticiones de conexión junto con la dirección IP, la fecha y la hora de la solicitud.

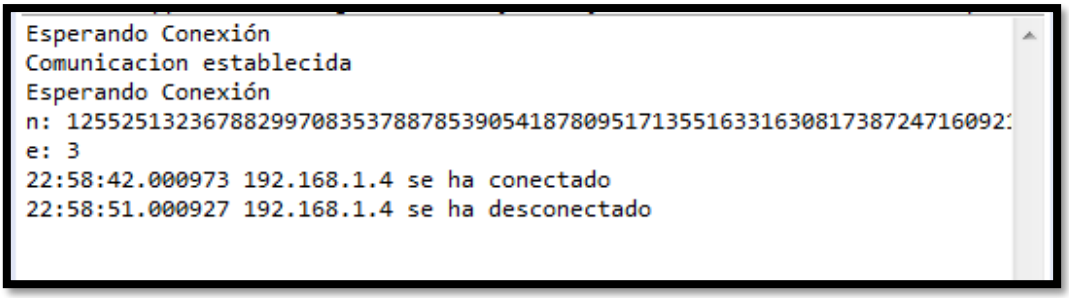

Figura 7. Consola de peticiones

3. Resultados y Discusión

En la *Tabla 1* se detalla el sistema de cifrado, se efectuaron varias pruebas de rendimiento cambiando el valor del exponente de la clave pública que utiliza la variable e. Mientras más cifras se tiene en el exponente de la clave pública, se tiene mayor seguridad de la información por el hecho que los tiempos de codificación y decodificación son mayores, y por ende el uso de recursos computacionales. Lo cual no permitirá que los ataques informáticos puedan descifrar la información, garantizando su autenticidad, confidencialidad y disponibilidad.

En la *Figura 8* se muestra el tiempo empleado en milisegundos por el método de cifrado RSA tanto en 1024 y 2048 bits al ser ejecutados en el servidor de aplicación para la distribución de datos en un ambiente inalámbrico.

Tabla 1. Pruebas de rendimiento con dígitos primos

A) Seguridad con 2 dígitos primos, e = 2				
número de cifras	número primo	tiempo para codificar	tiempo para decodificar	tiempo de proceso
2	29	108	114	6
2	43	695	702	7
2	67	109	116	6
2	97	139	146	7
			Promedio	6,5
B) Seguridad con 3 dígitos primos, e = 3				
número de cifras	número primo	tiempo para codificar	tiempo para decodificar	tiempo de proceso
3	223	806	816	10
3	479	558	567	9
3	773	645	655	10
3	997	781	790	9
			Promedio	9,5
C) Seguridad con 4 dígitos primos, e = 4				
número de cifras	número primo	tiempo para codificar	tiempo para decodificar	tiempo de proceso
4	1153	975	990	15
4	5393	173	200	27
4	7673	337	347	10
4	9733	311	325	14
			Promedio	16,5
D) Seguridad con 5 dígitos primos, e = 5				
número de cifras	número primo	tiempo para codificar	tiempo para decodificar	tiempo de proceso
5	10301	727	755	28
5	12421	770	800	30
5	98575	296	360	64
5	59799	340	360	20
			Promedio	35,5

La *Tabla 2* muestra los tiempos de procesamiento en el servidor utilizando el protocolo de cifrado RSA 1024 bits en dispositivos móviles ligeros (PDA), con la finalidad de evaluar el resultado de los procesos realizados anteriormente.

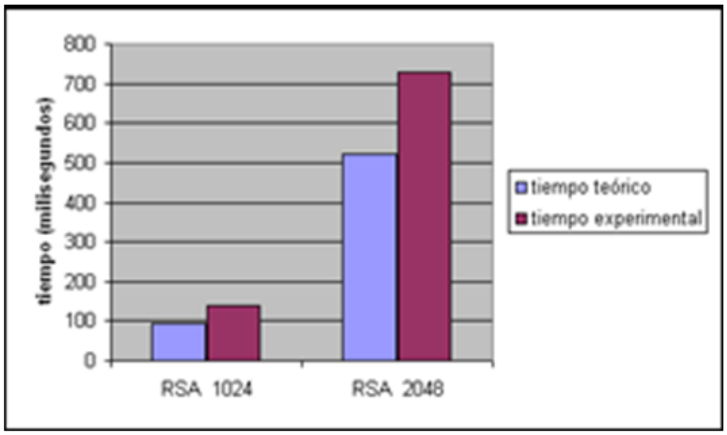

Figura 8. Tiempo teórico vs. Tiempo experimental en RSA

Los aspectos fundamentales en los resultados obtenidos reflejan similitud en el factor de procesos basados en el tiempo de ejecución, el mismo que es el principal componente que refleja el uso de números primos complejos, lo que requiere mayor número de recursos computacionales para llegar a optimizar el paso de encriptado y desencriptado en los dispositivos móviles, como se refleja en la *Tabla 3*.

Tabla 2. Tiempo estimado promedios (Milisegundos)

Hp (Cliente) / HP (Servidor)	THP (ms)	Ipaq (Client) / HP	THP (ms)
RSA 1024	22.8532	RSA 1024	95.0540
Curva 160 P	4.1814	Curva 160 P	6.87932
Curva 163 K	7.6730	Curva 163 K	22.6636
Curva 163 R	8.8217	Curva 163 R	24.0402
Híbrido usando curva 160 P	4.5136	Híbrido usando curva	6.2398
Híbrido usando curva 163 K	6.6657	Híbrido usando curva	14.7248
Híbrido usando curva 163 R	7.3579	Híbrido usando curva	15.3606
RSA 2048	139.7652	RSA 2048	522.5822
Curva 224 P	5.7201	Curva 224 P	13.4501
Curva 233 K	12.3072	Curva 233 K	39.0817
Curva 233 R	14.1477	Curva 233 R	40.8836
Híbrido usando curva 224 P	8.8119	Híbrido usando curva	16.0517
Híbrido usando curva 233 K	12.8829	Híbrido usando curva	30.5531
Híbrido usando curva 233 R	13.7194	Híbrido usando curva	31.5004

En el trabajo propuesto por (Carrera, 2010) se estudia los compromisos existentes entre una adecuada seguridad y factores como el desempeño y consumo de energía de aplicaciones ejecutándose sobre teléfonos celulares y asistentes digitales. Los resultados obtenidos muestran que la seguridad en las aplicaciones móviles no es gratuita y que su correcta implementación

requiere una selección cuidadosa de cada uno de los parámetros utilizados por los mecanismos criptográficos.

Tabla 3. Tiempo estimado promedios (Milisegundos) (R.= Resultado)

	Codificar (prom.)	Decodificar (prom.)	R.1	R.2
Trabajo Colaborativo	727	755	741	810
Trabajo Actual	770	800	785	800

En el trabajo desarrollado por (Miranda Arto, 2010) se expone sobre la mejora del rendimiento en los dispositivos móviles y el acceso a internet han hecho posible la comunicación a través de diferentes canales. Dichas conexiones seguras, sean cableadas o inalámbricas, se consiguen mediante la utilización de protocolos de seguridad, basados en algoritmos criptográficos. Estos algoritmos son seleccionados basándose en los objetivos de seguridad definidos en el protocolo de seguridad a utilizar. Entre ellos se incluyen algoritmos de encriptación simétricos y asimétricos, utilizados para proporcionar autenticación y encriptación de los datos, así como algoritmos basados en funciones hash, y, de esa manera, conseguir integridad en los mensajes intercambiados.

En lo referente al estudio de (Sierra & Lerch, 2014), el algoritmo RSA se basa en la complejidad de algunas funciones, como por ejemplo la multiplicación de dos números primos muy grandes y la factorización del resultado. Si bien ya existen granjas de GPU que son capaces de encontrar contraseñas de 8 caracteres en unas cinco horas, el algoritmo sigue siendo totalmente válido, ya que únicamente hace falta escoger números de 1024 bits para hacer que el proceso de factorización sea computacionalmente tan costoso en tiempo que no vale la pena iniciar el proceso de cálculo de la factorización

4. Conclusiones y Recomendaciones

El tiempo de cifrado y descifrado es mayor cuando el número primo sea mayor a tres cifras, es decir mayor seguridad, mayor tiempo de cifrado y descifrado. La ventaja de la utilización del algoritmo RSA se basa en que mientras más grande sea el exponente de la clave pública mayor será el tiempo que tome en cifrar y descifrar el mensaje. Tomando en cuenta que todo dependerá de lo que el usuario quiera: mayor seguridad o rapidez en la obtención de los datos.

Los trabajos futuros deben enfocarse en reforzar el intercambio seguro de información, para ello se deben combinar técnicas esteganográficas con criptografía. Además se debe mejorar la velocidad en los procesos de cifrar y descifrar, lo que involucra mayor seguridad en menor tiempo.

Bibliografía

Ambler, S. (2002). *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process.* John Wiley & Sons.

Barajas, S. (2004). Protocolos de seguridad en redes inalámbricas. *Universidad Carlos III de Madrid*.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns.* Wiley.

Cadenhead, R. (2014). *JAVA 8 (Programación).* ANAYA MULTIMEDIA.

Carrera, E. (2010). The Cost of Security in Mobile Devices. *3rd International Congress in Telecommunications, Information Technology and Communications.* Quito.

Cipriano, M. (2008). Factorización de N: recuperación de factores primos a partir de las claves pública y privada. *XIV Congreso Argentino de Ciencias de la Computación - CACIC.* Chilecito, La Rioja.

Downes, S., Belliveau, L., Samet, S., Rahman, M., & Savoie, R. (2010). Managing digital rights using JSON. *Consumer Communications and Networking Conference (CCNC), IEEE*.

Inc., G. (2016). *Android*. Recuperado el 2016, de https://www.android.com/intl/es_es/

Karygiannis, T., & Owens, L. (2002). Wireless network security 802.11, Bluetooth and handheld devices - NIST.

Kruchten, P. (2004). *The Rational Unified Process: An Introduction.* Addison–Wesley.

Li, J., & Wang, X. (2010). Research and Practice of Agile Unified Process. *IEEE Xplore Digital Library*.

Meza, J. (Noviembre de 2010). Cifrado y descifrado asimétrico con RSA utilizando C#/Mono. Recuperado el Abril de 2016, de http://blog.jorgeivanmeza.com/2010/11/cifrado-y-descifrado-asimetrico-con-rsa-utilizando-cmono/

Miranda Arto, P. (Diciembre de 2010). Consumo energético de algoritmos criptográficos y protocolos de seguridad en dispositivos móviles Symbian.

Montero Miguel, R. (2014). *JAVA 8 (Guía Práctica).* ANAYA MULTIMEDIA.

Pino, C., & Hernández, C. (2000). *Criptología y Seguridad de la Información: Actas de la VI Reunión Española Sobre Criptología y Seguridad de la Información: VI RECSI.* Ra-Ma.

Scolnik, H. (2014). *Qué es la Seguridad Informática.* Buenos Aires: Paidós SAICF.

Shabtai, A., Fledel, Y., Kanonov, U., Elovici, Y., Dolev, S., & Glezer, C. (2010). Google Android: A Comprehensive Security Assessment. *IEEE Xplore Digital Library*.

Sierra, J., & Lerch, D. (2014). *Esteganografía y estegoanálisis.* 0xWORD Computing S.L.

Young, A., & Young, M. (2006). An Elliptic Curve Asymmetric Backdoor inOpenSSL RSA Key Generation.