ISSN: 2007-4948

https://repository.uaeh.edu.mx/revistas/index.php/xikua/issue/archive

XIKUA Boletín Científico de la Escuela Superior de Tlahuelilpan
13º Congreso Internacional de Computación
Inteligencia artificial: Presente y futuro
Red Iberoamericana de Computación
Vol. 12, Número Especial (2024) 13-19

Análisis de Eficiencia del Algoritmo RSA en Entornos Móviles de Bajo Desempeño Computacional

Efficiency Analysis of the RSA Algorithm in Low Computational Performance Mobile Environments

Rocío Rodríguez Guerrero [a], *Carlos A. Vanegas* [b], *Gerardo Castang Montiel* [c]

Abstract:

This article shows the results of a research project carried out at the Universidad Distrital Francisco Jose de Caldas with the implementation of the encryption algorithm proposed by Rivest, Shamir and Adleman (RSA) in mobile computing environments and low computational resources, for this purpose, tests were carried out with the algorithm for prime numbers with a smaller number of bits (between three and nine bits), which make up the prime numbers necessary for the generation of public and private keys, in order to validate the computational efficiency and whether it is sufficient to use this type of key to protect information in such computing environments.

Keywords:

RSA encryption, mobile environments, asymmetric cipher, information security.

Resumen:

Este artículo muestra los resultados de un proyecto de investigación realizado en la Universidad Distrital Francisco José de Caldas con la implementación del algoritmo de cifrado propuesto por Rivest, Shamir y Adleman (RSA) en entornos computacionales móviles y bajos recursos computacionales, para tal efecto, se realizaron pruebas con el algoritmo para números primos con menor número de bits (entre tres y nueve bits), los cuales conforman los números primos necesarios para la generación de las claves públicas y privadas, con la finalidad de validar la eficiencia computacional y si es suficiente utilizar este tipo de clave para proteger la información en tales entornos computacionales.

Palabras Clave:

Cifrado RSA, sistemas móviles, cifrado asimétrico, seguridad de la información.

[a] Universidad Distrital Francisco José de Caldas | Maestría en Gestión y Seguridad de la Información- Tecnología en Sistematización de Datos | Bogotá| Colombia, https://orcid.org/0000-0002-2956-9650, Email: rrodriguezg@udistrital.edu.co

[b] Universidad Distrital Francisco José de Caldas | Maestría en Gestión y Seguridad de la Información -Tecnología en Sistematización de Datos | Bogotá | https://orcid.org/0000-0002-8173-8167, Email: cavanegas@udistrital.edu.co

[c] Universidad Distrital Francisco José de Caldas | Ingeniería en Telemática-Tecnología en Sistematización de Datos | Bogotá |Colombia, https://orcid.org/0000-0001-9788-5121, Email: gacastangm@udistrital.edu.co

Fecha de recepción: 09/04/2024, Fecha de aceptación: 08/05/2024, Fecha de publicación: 01/07/2024

DOI: https://doi.org/10.29057/xikua.v12iEspecial.12694

Introducción

El uso de sistemas informáticos, la información que se almacena y transmite, su protección y la utilización adecuada de los recursos computacionales impulsa el estudio y desarrollo de soluciones que brinden seguridad y sean al mismo tiempo eficientes computacionalmente [1].
Es por ello, que el desarrollo de esta investigación se orienta en analizar y adaptar algoritmos de cifrado asimétricos en entornos móviles analizando su desempeño computacional en estos entornos.
Se define como un Criptosistema al conjunto de algoritmos que garantizan la seguridad de la información enviada a través de un proceso de comunicación, donde se utilizan técnicas criptográficas que puedan prevenir y contrarrestar posibles ataques, proporcionando confidencialidad e integridad en los datos. La seguridad de la información, según la norma ISO 27001:2013, consiste en la preservación de su confidencialidad, integridad y disponibilidad, así como de los sistemas implicados en su tratamiento dentro de una organización [2].

Tomando en cuenta que el uso del dispositivo móvil va en aumento, según cifras del Radicati Group site [3], más del 63% usan Internet en todo el mundo, de estos usuarios el 92,4% utiliza un teléfono móvil para conectarse a Internet, el informe muestra que un estimado de crecimiento en la cantidad de usuarios de esta tecnología en el mundo supere los 6.800 millones para el 2023, estos dispositivos son empleados para el esparcimiento, las noticias, el trabajo y por supuesto la comunicación.

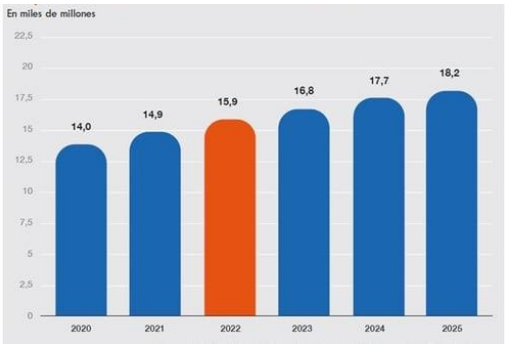


Figure 1. Perspectivas de crecimiento del número de móviles en el mundo 2020-2025. Fuente: [3]

Análisis Del Criptosistema RSA

La criptografía tiene como objetivo principal cifrar la información, con la finalidad de protegerla [4], para ello, emplea algoritmos que suelen usar una o más claves, de tal manera que la comunicación entre emisor y receptor se realice con seguridad y privacidad.

La criptografía se ubica en la rama de la criptología junto con el criptoanálisis, este último, dedicado al estudio de todo lo relacionado con técnicas para descifrar la información. Algunas de las características del criptoanálisis aplicadas a un sistema criptográfico o criptosistema son [5]:

- Asegurar su robustez y resistencia.
- Descubrir debilidades para evitar posibles ataques.
- Fortalecerlo, de forma que se vaya aumentando su seguridad.

Los métodos criptográficos empleados por los criptosistemas en el momento de realizar el cifrado, según el uso de las claves se clasifican en asimétricos y simétricos; la tabla Nro. 1 describe algunas de sus cualidades.

	Clave pública	Clave privada
Gestión de Claves	Sólo es necesario memorizar la clave privada del emisor y la pública del receptor.	Hay que memorizar un número muy alto de claves (N^2)
Longitud y Espacio de Claves	La clave es del orden de miles de bits.	La clave es del orden de centenares de bits.
Vida de una Clave	La duración de la clave suele ser larga.	La duración de la clave es muy corta, normalmente caduca al finalizar la sesión.
Autenticación	Al existir una clave pública y otra privada se puede autenticar el mensaje y al emisor.	Solo es posible autenticar el mensaje.
Velocidad de Cifra	La velocidad de cifra es muy lenta.	La velocidad de cifra es muy alta.
Uso	Son utilizados sobre todo para intercambios de claves y firma digital.	Son algoritmos utilizados sobre todo para cifrado.

Tabla 1. Criptosistemas asimétricos vs simétricos
Fuente: (Escobar ,2011).

Los autores Rivest, Shamir y Adleman propusieron un criptosistema de clave pública, basado en el producto de dos primos p y q seleccionados aleatoriamente [6], estos números multiplicados permiten obtener un número n.

$$n = p \cdot q \quad \text{(Ecuación 1).}$$

El siguiente paso consiste en generar el conjunto de enteros para seleccionar los nuevos números p y q, usando la función φ de Euler,

$$\varphi(n) = (p - 1)(q - 1) \quad \text{(Ecuación 2).}$$

Con base en lo anterior se calculan dos parámetros, la clave pública y la clave privada (e, d), las cuales se encuentran relacionadas en la siguiente expresión:

$$e \cdot d = 1 + k \cdot \varphi(n) \text{ para un entero k. (Ecuación 3).}$$

Es importante mencionar que el número k sea menor que φ(n) y coprimo.

Se debe satisfacer:

$$e \cdot d \equiv 1 \pmod{\varphi(n)} \quad \text{(Ecuación 4)}.$$

Es decir:

$$d = e\text{-}1 \pmod{\varphi(n)} \quad \text{(Ecuación 5)}.$$
$$e = d\text{-}1 \pmod{\varphi(n)} \quad \text{(Ecuación 6)}.$$

Generando el conjunto de claves pública y privada

$$K\,pb = (e, n) \quad \text{(Ecuación 7)}.$$
$$K\,pr = (d, n) \quad \text{(Ecuación 8)}.$$

La clave resultante es obtenida a partir de las dos componentes:

$$<Kpb, Kpr> \quad \text{(Ecuación 9)}.$$

La fortaleza del algoritmo RSA consiste en la dificultad de encontrar la clave privada a partir de la clave pública ya depende de la longitud de los números primos p y q, pero la debilidad consiste en la cantidad de procesamiento y tiempo computacional requerido para llevar a cabo este proceso, es decir la obtención de las claves y el cifrado de los mensajes dependiendo de la longitud de este puede consumir mucho tiempo y también muchos recursos computacionales.

Implementación del algoritmo RSA en un entorno móvil.

Se implementó el algoritmo RSA en los dispositivos móviles utilizando la herramienta Kivy para la interfaz gráfica y la lógica de programación en el lenguaje Python [7]. Posteriormente se realizó el procedimiento para convertir dicho código en un archivo .apk (Android Application Package), el cual se puede instalar en un dispositivo móvil que tenga el sistema operativo Android. El entorno de la aplicación se observa en la figura 2.

La aplicación móvil consta de tres ventanas:

- En la primera ventana permite digitar los números primos P y Q. En esta ventana con la lógica de programación se valida que los dos números sean primos y que ninguno de los campos de texto quede sin valores.

- En la segunda ventana se imprime el valor de N, el valor de la función Euler, los números primos generados según los valores de P y Q, un numero primo seleccionado aleatoriamente de los números primos generados, la llave pública, la llave privada y el tiempo procesamiento para obtener dicha información en milisegundos.

Figure 2. Ventana inicial de la aplicación móvil RSA Fuente: autores

- En la tercera ventana se permite digitar el mensaje a encriptar/desencriptar y con dicho mensaje se realiza el proceso de encriptamiento y desencriptamiento para ser realizados, como también, se calcula el tiempo que dura dicho procesamiento en milisegundos.

Cuando se ejecuta la aplicación móvil y se digita como valor 11 para P y 23 para Q, y al pulsar el botón Digitar mensaje, posteriormente, al digitar “Hola mi gente colombiana 123456 ^%$” y luego pulsar el botón Encriptar/Desencriptar, se obtendría un entorno de salida como se observa en la figura 3.

Figure 3. Proceso de encriptación y desencriptación del mensaje en la aplicación móvil. Fuente: autores

Entorno de Pruebas del Algoritmo RSA en Entornos Móviles

Se realizaron las pruebas con los conjuntos de números primos en cuatro secciones o grandes grupos, relacionados con la longitud de los números primos en bits. Por ejemplo, en la primera sección se seleccionaron los números (7,11), (13,19), (29,37), (43,59), (61,71), (73, 89).

En la segunda sección se realizaron distintas combinaciones entre números cercanos (97, 103), (109, 127), y otros un poco más distanciados, (137,151) y así hasta la combinación (199,223), es decir desde el rango de los tres bits hasta los ocho bits para la generación de los números primos seleccionados.

En la medida que la generación de los números primos aumentaba, el tiempo de generación de las claves aumentaba, pero no se podía medir, por lo que era necesario incluir la funcionalidad para calcular el tiempo de generación de clave; también, se realizaron pruebas con distintas combinaciones de un mensaje corto:

- Caso Nro. 1. Prueba del algoritmo rsa en móviles
- Caso Nro. 2. Prueba del algoritmo rsa en móviles
- Caso Nro. 3. Prueba del algoritmo rsa en móviles
- Caso Nro. 4. Prueba del algoritmo rsa en móviles

Y así sucesivamente, con el fin de poder validar que se procesará un mensaje diferente y visualizar los resultados en la pantalla del móvil.

El tiempo de cifrado y descifrado del mensaje aumenta en la medida que se aumentan los números primos, pero las variantes no son muy significativas en los primeros rangos de los números seleccionados

Para las secciones superiores se realizaron las pruebas, el algoritmo funciona y se demora en generar las claves, las pruebas también se realizan en otros dispositivos móviles, con el mismo tipo de mensaje. Las dos últimas pruebas se realizaron con las combinaciones (479, 491) y (499, 503), es decir hasta los nueve bits para la generación de los números primos.

El proceso de cifrado y descifrado de mensajes en algunos casos se demora, en otras ocasiones se reinicia la aplicación, lo que implica realizar varios intentos con el mismo conjunto de números primos, el proceso se realiza codificando y decodificando los mensajes de prueba. El tiempo de generación de las claves aumenta, el consumo computacional es alto (la pantalla queda negra por unos momentos), pero se generan las claves y se procesan los mensajes propuestos. Los rangos de los números primos tienen distintos números de bits para codificar los mensajes como se observa en la tabla 2.

Número de bits	Potencia	Rango
5	(2^5)-1	0-31
6	(2^6) - 1	0-63
7	(2^7) - 1	0-127
8	(2^8) - 1	0-255
9	(2^9) - 1	0-511
10	(2^10) - 1	0-1023

Tabla No 2. Rango de los números primos según el número de bits seleccionado

Las pruebas se realizaron en tres dispositivos móviles con características de hardware y procesamiento similares. Se realizan las pruebas calculando el tiempo de cifrado y descifrado del mensaje, y el tiempo de generación de las llaves públicas y privadas.

RESULTADOS

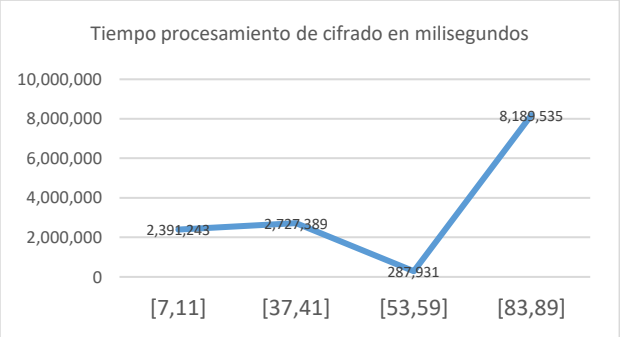
En el primer caso se realizaron las pruebas en los dispositivos móviles cuando se calcula el tiempo de cifrado y descifrado del mensaje. En el segundo caso se incluye el tiempo de generación de llaves, aplicando el algoritmo RSA [8] en el dispositivo móvil correspondiente.

La tabla Nro. 3 presenta los datos generados para el rango de los números primos, N, Z(n), las llaves que se generaron y el tiempo de procesamiento de cifrado para el caso de mensaje No1 para el rango de los números primos [7 y 89], es decir para números primos seleccionados entre tres y siete bits.

Números primos P-Q	N=P*Q	Z(n)	Nro. primo seleccionado aleatoriamente	Llave publica (e, n)	Llave privada (d, n)	Tiempo procesamiento en milisegundos
[7,11]	77	60	41	41, 77	41,77	2.391.243
[37,41]	1517	1440	841	841, 1517	601, 1517	2.727.389
[53,59]	3127	3016	2221	2221, 3127	1085, 3127	287.931
[83,89]	7387	7216	3067	3067,7387	4515,7387	8.189.535

Tabla No 3. Datos generados por la app movil RSA para caso de mensaje No1.

La gráfica Nro. 1 muestra el tiempo de procesamiento en milisegundos correspondiente al cifrado del mensaje para el caso Nro. 1 para el rango de los números primos [7,89].



Gráfica Nro. 1. Tiempo de procesamiento en milisegundos cifrado del mensaje para el caso Nro. 1.

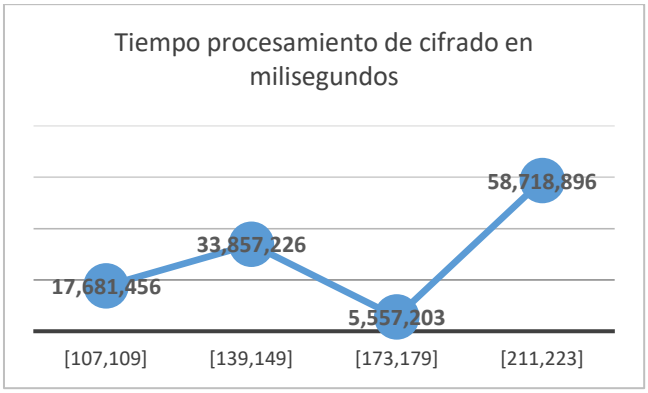
En este caso se observó que en la medida que el número primo seleccionado aleatoriamente aumenta, es decir, cuando el valor del parámetro *e* aumenta y de igual forma el parámetro d, el tiempo de procesamiento en milisegundos aumenta, esto se presenta de manera secuencial del primero al cuarto rango.
En los tres primeros conjuntos de valores el tiempo de procesamiento en milisegundos no varía significativamente, pero para el cuarto rango, el tiempo de procesamiento se incrementa por un factor aproximado igual a tres, posiblemente se debe a que la diferencia entre el valor de los parámetros e y d es mayor en el cuarto rango.
La diferencia en tiempos entre el primer y cuarto rango son sesenta milisegundos aproximadamente, y supera por poco los ochenta milisegundos en el cuarto rango, es decir que oscila entre los veinte y los ochenta milisegundos aproximadamente.
La tabla Nro. 4 presenta los datos generados para el rango de los números primos, N, Z(n), las llaves que se generaron y el tiempo de procesamiento de cifrado para el caso Nro. 1 para el rango de los números primos [107 y 223], es decir para números primos seleccionados entre siete y ocho bits.

Números primos P-Q	N=P*Q	Z(n)	Nro. primo seleccionado aleatoriamente	Llave publica (e, n)	Llave privada (d, n)	Tiempo procesamiento en milisegundos
[107,109]	11663	11448	10201	10201,11663	7225,11663	176.81456
[139,149]	20711	20424	18293	18293,20711	11549,20711	338.57226
[173,179]	30967	30616	4553	4553,30967	1049,30967	55.57203
[211,223]	47053	46620	45677	45677,47053	6773,47053	587.18896

Tabla No 4. Datos generados por la app movil RSA para caso No1 para números primos [107,223]

La gráfica Nro. 2 muestra el tiempo de procesamiento en milisegundos correspondiente al cifrado del mensaje para el caso Nro. 1 para el rango de los números primos [107,223].

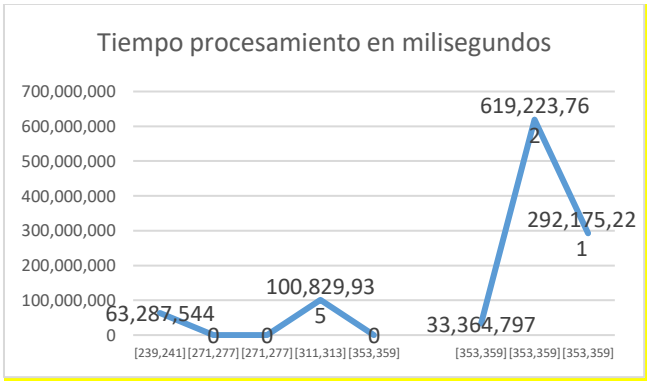
Se observa de la gráfica anterior que en la medida que el número primo seleccionado aleatoriamente aumenta, el tiempo de procesamiento en milisegundos aumenta, aunque en este caso no se presenta el aumento del número primo de manera secuencial ni ordenada.



Gráfica No 2. Tiempo de procesamiento en milisegundos cifrado del mensaje para el caso No1 para números primos [107,223].

El tiempo de procesamiento en segundos varía significativamente en todos los rangos, especialmente del tercer al cuarto rango, en la medida que aumenta el valor para los parámetros *e* y *d*; hay una diferencia en tiempo de quinientos treinta milisegundos entre el tercer y cuarto rango, es decir, hay una variación desde los cincuenta y cinco milisegundos hasta los quinientos ochenta y siete milisegundos.

La gráfica Nro. 3 muestra el tiempo de procesamiento en milisegundos correspondiente al cifrado del mensaje para el caso Nro. 1 para el rango de los números primos [239 y 359], es decir para los números primos seleccionados entre ocho y nueve bits.

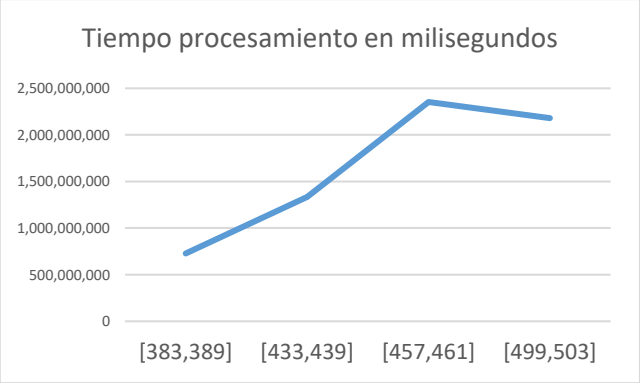


Gráfica No 3. Tiempo de procesamiento para el tercer rango de los números primos seleccionados.

En el tercer rango de números primos entre [239 y 359] se presenta mayor inestabilidad en cuanto a los datos y resultados, ya que la aplicación se reinicia varias veces dependiendo del valor del número primo seleccionado aleatoriamente en el intervalo, es decir, cuando es un valor alto, generalmente la aplicación se reinicia; tampoco se presenta una secuencia estructurada ni ordenada en cuanto al número primo seleccionado aleatoriamente y tampoco se cumple la premisa que en la medida que

aumenta el número primo seleccionado aleatoriamente el tiempo de procesamiento aumenta.

En este rango, en el primer intervalo el valor del parámetro *e* es menor al valor del parámetro *d*, en el segundo intervalo el valor del parámetro *e* es mayor al valor del parámetro *d*; en el tercero el valor de *e* es menor al valor de *d*; en el cuarto intervalo se encuentran el menor y el mayor valor del tiempo de procesamiento en milisegundos, cuando el valor del parámetro *e* es mayor al valor del parámetro *d*, se obtiene el menor tiempo.
Existe una diferencia de aproximadamente cinco mil ochocientos milisegundos entre el menor valor de trescientos treinta milisegundos y el mayor valor de seis mil doscientos milisegundos respectivamente. Para este rango, el mayor tiempo de procesamiento se aproxima a los seis puntos dos segundos.
La gráfica Nro. 4 muestra el tiempo de procesamiento en milisegundos correspondiente al cifrado del mensaje para el caso Nro. 1 para el rango de los números primos [383 y 503], es decir para números primos entre ocho y nueve bits.



Gráfica No 4. Tiempo de procesamiento para el cuarto rango de los números primos seleccionados.

En el cuarto rango de números primos entre [383 y 503] el tiempo de procesamiento se encuentra en segundos, el cual oscila de los siete puntos tres segundos a los veintitrés punto seis segundos, existiendo una diferencia de dieciséis segundos aproximadamente entre el mayor y el menor valor del tiempo de procesamiento.
En este caso, en el primer intervalo, el valor del parámetro *e* es menor al valor del parámetro *d* siendo relativamente cercanos y se obtiene el menor tiempo de procesamiento; en el segundo intervalo, el valor del parámetro *e* es menor al valor del parámetro *d* y la diferencia es significativa entre ellos; en el tercero el valor del parámetro *e* es menor al valor del parámetro *d* siendo la diferencia entre ellos bastante significativa y se obtiene el mayor tiempo de procesamiento y en el cuarto intervalo, ambos parámetros son bastante altos pero cercanos, y el valor del parámetro *e* es mayor al valor del parámetro *d*.
El menor valor se presenta en el primer intervalo, y el mayor valor se presenta en el tercer intervalo respectivamente, dentro del cuarto rango

Análisis General de los Resultados en los Intervalos de Selección Propuestos

En el primer rango el valor de *e* y *d* son iguales, en los dos siguientes rangos el valor de *e* es mayor al valor de *d*, y en el cuarto rango el valor de *e* es menor al valor de *d*, lo que posiblemente influya en el resultado del tiempo de procesamiento en milisegundos. Para este rango, el mayor tiempo de procesamiento se aproxima a los cien milisegundos. En el segundo rango, el valor del parámetro *e* siempre es mayor al valor del parámetro *d*, lo que posiblemente influya en el tiempo de procesamiento en milisegundos. Para este rango, el mayor tiempo de procesamiento se aproxima a los seiscientos milisegundos.
En el tercer rango, cuando el valor del parámetro *e* es menor al valor del parámetro *d*, por un factor aproximado de 3.5, se tiene el mayor valor del tiempo de procesamiento. En el cuarto rango, el valor de los tiempos de procesamiento no se presenta de manera ordenada respecto a los intervalos del rango, siguiendo la secuencia de tiempo menor-mayor-mayor-menor.

Conclusiones

El algoritmo RSA implementado para móviles usando el sistema operativo Android encripto el mensaje, pero presentó algunos inconvenientes al momento de desencriptar el mensaje usando números primos en el rango de números primos de 7 a 11, así como el tiempo de generación de las claves tiende a aumentar a medida que se eleva el rango de los números primos.
En el tercer rango la aplicación móvil presenta una mayor inestabilidad, generalmente se cierra y no encripta ni desencripta los mensajes.
Hasta los primeros intervalos del tercer rango se obtuvo el tiempo de procesamiento en milisegundos.
En el cuarto rango de números primos se requiere un mayor tiempo de procesamiento y mayores recursos computacionales para la generación de las claves y el cifrado de los mensajes.
En el cuarto rango se presenta la mayor confiabilidad y estabilidad para el cifrado y descifrado de mensajes en un dispositivo móvil utilizado.

Referencias

[1] Muñoz H, Zapata L, Requena D, Ricardo L. (2019). Riesgos informáticos y alternativas para la seguridad informática en sistemas contables en Colombia. Revista Venezolana de Gerencia, vol. 2, Universidad del Zulia, Venezuela. Disponible en: https://www.redalyc.org/articulo.oa?id=29063446029

[2] Torres J. (2021). Evaluación de vulnerabilidades de seguridad en Software Android en el año 2021. Tecnología vital. Pág 11-15. disponible en https://revistas.ulatina.ac.cr/index.php/tecnologiavital/article/view/465/585

[3] Ditrendia. (2023). Todas las estadísticas sobre móviles que deberías conocer MWC23. Blog Digital Marketing Trends.

Disponible: https://mktefa.ditrendia.es/blog/estadisticas-sobre-móviles-2023#:~:text=Se%20estima%20que%20la%20cantidad,en%202023%20cercana%20al%2085%25

[4] Plaza Martín F. (2021). Manual de Criptografía Fundamentos matemáticos de la Criptografía para un estudiante de Grado. Ediciones Universidad de Salamanca. ISBN: 978-84-1311-463-7 (PDF). DOI: https://doi.org/10.14201/0DD016

[5] 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

[6] A. 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

[7] Rodriguez, R. Vanegas, C. Castang, G. (2020). Python a su alcance. Editorial UD.

[8] Fuster, A. Martínez, D. Hernández, L. Montoya, F. Muñoz, J. (2001). Técnicas Criptográficas de Protección de Datos. Alfaomega.