Algoritmo criptográfico RSA para encriptación y desencriptación de imágenes

Ángel Plácido Espinal Huaman
Universidad Nacional San Antonio Abad del Cusco
Ing. Informática y de Sistemas
Cusco, Perú
16454@unsaac.edu.pe

Sebastian Israel Macedo Gheiler
Universidad Nacional San Antonio Abad del Cusco
Ing. Informática y de Sistemas
Cusco, Perú
164243@unsaac.edu.pe

Michael Antonni Mamani Quinta
Universidad Nacional San Antonio Abad del Cusco
Ing. Informática y de Sistemas
Cusco, Perú
164566@unsaac.edu.pe

Bruno Pérez Tomaylla
Universidad Nacional San Antonio Abad del Cusco
Ing. Informática y de Sistemas
Cusco, Perú
130322@unsaac.edu.pe

Resumen—**Los grandes avances en la criptografía han sido posibles gracias al desarrollo que se ha producido en el campo de las matemáticas y las ciencias de la computación. Hoy más que nunca la criptografía a través de los algoritmos matemáticos, se utiliza para proteger la información y dotar de seguridad a las comunicaciones y las entidades que se comunican. Como muchas otras áreas relacionadas a la computación criptográfica está en evolución constante, presionada de un lado por avances de la naturaleza científica, de otro lado por sus aplicaciones comerciales.**
El presente documento responde a la necesidad de realizar criptografía matemática para la encriptación de un archivo texto y archivo imagen (JPG, PNG) a través del algoritmo RSA.

Palabras clave—Criptografía, RSA, Protección de datos

I. Introducción

En las empresas y organizaciones la información es un activo de considerable valor, debido a esto la importancia de emplear mecanismos de encriptación que garanticen su seguridad, confiabilidad e integridad. Hoy más que nunca la criptografía a través de los algoritmos matemáticos, se utiliza para proteger la información y dotar de seguridad a las comunicaciones y a las entidades que se comunican. Los grandes avances de la criptografía han sido posibles gracias al desarrollo que se ha producido en el campo de las matemáticas y las ciencias de la computación. Históricamente primero han aparecido los códigos basados en una única clave conocidos como sistemas de clave privada. Estos tienen el inconveniente de que el remitente y el receptor deben estar de acuerdo, con anterioridad, en los valores de la clave. Teniendo en cuenta que por seguridad deberán cambiar la clave de vez en cuando, ¿de qué modo podrían hacerlo de una manera segura? Esta dificultad se supera con la aparición de los sistemas criptográficos de clave pública que permite a todos los remitentes utilizar la clave pública del emisor para la encriptación de un mensaje que sólo el destinatario podrá recuperar pues será el único poseedor de la clave privada.

En este trabajo se utilizará la metodología cualitativa presentando el método desarrollado en 1978 por R. L. Rivest, A. Shamir y L. Adleman y que es conocido como sistema criptográfico RSA por las iniciales de sus autores. Basa su seguridad en la dificultad de factorizar números primos muy grandes aunque como todo sistema de encriptación de clave pública, el RSA puede estar sujeto a ataques con el fin de obtener el mensaje original o descubrir la clave privada. En este trabajo se describirá la implementación del algoritmo RSA en el lenguaje de programación C Sharp que permite interactuar con el usuario mediante una interfaz que genera claves públicas y privadas, así como la encriptación y desencriptación de los archivos.

En este trabajo nos ubicamos en este contexto, desde el punto de punto de vista computacional. Nos enfocamos con el análisis y diseño de soluciones, así como con los algoritmos. Hemos hecho el esfuerzo de consultar los artículos originales.

II. Estado de Arte

Las organizaciones deben considerar la criptografía como una de las herramientas esenciales para asegurar la información y establecer más estudios y análisis para entender el funcionamiento, estructura y características de los algoritmos que puedan utilizar [8]. En la transmisión de la información 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 [7].

El RSA puede ser aplicado en sistemas que trabajan con big data para proteger la información que esta almacenado en servidores de terceros [3]. Los ataques al criptosistema RSA con exponente de cifrado corto han iluminado el interés de utilizar exponentes de descifrado tan grandes como sea posible, de hecho, para evitar estos grandes ataques se suele recomendar que el tamaño del exponente de descifrado sea aproximadamente el mismo que el del módulo del criptosistema [10]. Por otro lado, RSA puede sufrir ataques en fuerza bruta para poder desencriptar la información; existen métodos en el cual se pueden proporcionar mayor seguridad, uno de ellos es el algoritmo de Jordan Totient, este algoritmo utiliza doble cifrado, dobles claves públicas y privadas [2]; también existe una forma modificada del algoritmo RSA para aumentar la seguridad de los datos durante el intercambio de datos a través de la red, incluyendo K-nearest Neighbor Algorithm [5]. En la actualidad la encriptación y desencriptación del algoritmo RSA consume la mayor parte de los recursos informáticos, pero con la ayuda de la programación paralela en una GPU se puede optimizar el tiempo gastado en la ejecución del algoritmo RSA usando técnicas como transformadas de Fourier y el método de Newton para diseñar un nuevo algoritmo paralelo [4].

También se utiliza el sistema de computación matemática MAPLE V R4 como soporte didáctico para introducir la criptografía y especialmente el algoritmo de clave pública RSA [6].

III. APORTE

A. Marco teórico

a) Criptografía: Proviene del griego kryptos que significa graphia y escrictura segun Deffie & Hellman(1976) es el sistema para resolver 2 problemas de seguridad que son: privacidad y autentificacion de los datos, la criptografica se encarga de diseñar, transformar el mensaje cifrado que puede ser factible a su conocimiento[13].Se define a la criptografía como una ciencia encargada de diseñar funciones capaces de transformar mensajes legibles a mensajes cifrados de tal manera que esta transformación y su inversa sólo puede ser factible al conocimiento de una o más llaves [12].

b) Imagen digital: Es un arreglo de pixeles, representada de una forma bidimensional a partir de una matriz binaria (0's y 1's). son llamados tambien bitmap que es correspondido a un mapa de bits el muestro de imagenes digitales se produce de dos pasos importantes que son: el muestreo y la cuantificación; el muestreo es el proceso de dividir imagen y la cuantificación es el proceso de asignar un valor entero a cada pixel [14].

c) Clave pública (CCP): El esquema CCP usa una clave para el cifrado y una clave diferente para el descifrado. El CCP moderno se describió por primera vez utilizando un sistema de cifrado de dos claves en el que dos partes podían entablar una comunicación segura a través de un canal de comunicaciones seguro sin tener que compartir una clave secreta. El PPC, es una de las claves que se designa como clave pública y puede anunciarse tan ampliamente como desee el propietario. La otra clave se designa como clave privada y nunca se revela a otra parte. RSA es una de las primeras y más comunes implementaciones de PPC que se utiliza hoy en día para el intercambio de claves o firmas digitales. La principal ventaja de este método es que la administración de claves en una red requiere la presencia de solo una TTP funcionalmente confiable, en contraposición a una TTP incondicionalmente confiable. Dependiendo del modo de uso, es posible que el TTP solo se requiera "fuera de línea", en lugar de hacerlo en tiempo real. Muchos esquemas de clave pública producen mecanismos de firma relativamente eficientes. La clave utilizada para describir la función de verificación pública es típicamente mucho más pequeño que para la contraparte de clave simétrica [1].

d) Clave privada (SKC): El método SKC utiliza una única clave tanto para el cifrado como para el descifrado. Los esquemas generalmente se clasifican como cifrados de flujo o cifrados en bloque. Los cifrados de flujo operan en un solo bit (byte o palabra de computadora) a la vez e implementan algún tipo de mecanismo de retroalimentación para que la clave cambie constantemente, mientras que el esquema de cifrado de bloques encripta un bloque de datos a la vez usando la misma clave en cada bloque. El principal inconveniente de este método es el error de propagación porque un bit distorsionado en la transmisión dará como resultado n bits distorsionados en el lado receptor. Aunque los cifrados de flujo no propagan errores de transmisión, son periódicos, por lo tanto, el flujo de claves se repetirá eventualmente. Esto normalmente da como resultado el uso de mecanismos de firma digital con claves grandes para la función de verificación pública o el uso de un TTP [1].

d) Algoritmo RSA

Es un algoritmo criptográfico creado por Rivest, Shamir y Adlman en 1978 le permitirá al usuario a tener la confidencialidad de su información y es uno de los algoritmos criptográficos más conocido y usado [15]. RSA utiliza dos claves: una clave pública, formada por los números e y n; y una clave privada formada por los números d y n [16][17].

Las claves se generan de la siguiente manera:

- Escoger dos números aleatorios grandes primos p y q. Estos deben mantenerse secretos
- Calcula n=p*q que será el modulo para la clave pública y privada.
- Calcula el Totiente ϕ de *n:* $\phi(n) = (p-1) * (q-1)$
- Escoge un entero e, tal que e sea co-primo con $\phi(n)$ y $1 < e < \phi(n)$. El par de números (n, e) constituye la clave pública.
- Calcular d tal que $e.d = 1 mod \phi(n)$, d puede ser encontrado usando la función extendida de algoritmos de Euclides. El par de números (n, d) constituye la clave privada.

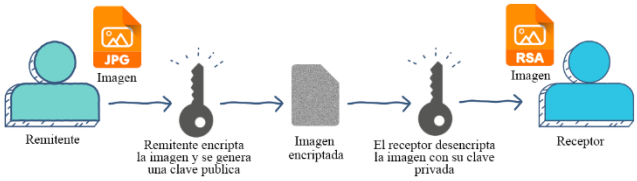

Figura 1. Diagrama de colaboración

B. Algoritmo RSA para imagenes

Debido al formato de las imágenes hay una reducción del espacio de trabajo con $Z_n = 256$ por la limitación de paleta de color, se deben de buscar números primos p y q diferentes, cuyo producto resulte el máximo posible y abarque el mayor rango de los tonos de píxeles, para que se trabaje sobre un nuevo espacio de trabajo $Z_{n'}$ donde $n' = pq$ y $n' \leq n$. Teniendo en cuenta estas restricciones podemos usar el algoritmo de generación de claves para RSA y obtener las claves pública y privada [18].

Dado que $n' \leq n$ existirán $r = n - n'$ tonos de píxel que quedarán fuera del espacio $Z_{n'}$ y que por lo tanto no serán cifrados. Se sabe que $r > 0$ dado que no existen números primos p y q diferentes tal que $pq = 256$, concluiríamos que es un hecho que quedarán tonos de píxel sin cifrar. Los r tonos de píxel serán aquellos que se encuentren entre n' y n-1, entonces, serán cercanos al tono 255, lo que significa que estarán formados por tonos claros [18]. Para solucionar el problema de los tonos de pixel sin cifrar se tomaría en cuenta agregar un paso más al algoritmo en el cual sería:

- Para cada matriz componente de Imagen: R, G y B, fraccionar en f submatrices. Distribuir las f submatrices de R, G y B siguiendo una misma secuencia para las tres y obteniendo así las nuevas matrices R1, G1 y B1, que son las componentes de la nueva imagen ImFrag (Es una imagen temporal que será la criptoimagen).

a) Algoritmo para generar claves RSA

Para la creación de claves pública y privada, se recurre a las matemáticas, se debe obtener dos números primos que al multiplicarse se obtendrá un valor que luego será calculado por la función Totient y un Phi que represente una condición

de primos única, el algoritmo es representado por una cota $O(log\ n)^3$ ya que depende de los valores generados por p y q.

SALIDA: Clave pública RSA *(e,n)* y clave privada *(d,n).*
1: Elegir dos números primos p y q de por lo menos 512 bits.
2: Calcular $n = pq$ y $\phi(n) = (p-1)(q-1)$.
3: Elegir un entero e impar arbitrario con $1 < e < \phi(n)$ y tal que cumpla $(e, \phi(n)) = 1$.
4: Calcular el entero d que satisfaga $1 < d < \phi(n)$ y $e^d \equiv 1 mod\ \phi(n)$.
5: Devolver *(n, e, d).*
6: Fin.

b) Algoritmo RSA para cifrar

Para el cifrado, se utiliza una función que calcula el dato cifrado, se necesita extraer los valores de cada pixel y transformar estos valores con la ayuda de la función RSA. La función RSA se desempeña con una complejidad de $O(log\ n)$ ya que d se divide en cada iteración.

Requiere: Imagen: imagen a cifrar, e: clave publica
Asegurar: *CriptoIm*: Resultado de cifrar Imagen
1: Mientras i <x número de filas de Imagen hacer
2: Mientras j < número de columnas de Imagen hacer
3: $Pixel \leftarrow Imagen_k[i,j]$
4: $R1 \leftarrow RSA(Pixel.R + 10, e, n)$
5: $G1 \leftarrow RSA(Pixel.G + 10, e, n)$
6: $B1 \leftarrow RSA(Pixel.B + 10, e, n)$
7: $CriptoIm[i,j] \leftarrow RGB(R1\%256, G1\%256, B1\%256)$
8. Fin Mientras
9. Fin Mientras
10. Devolver CriptoIm

c) Algoritmo RSA para descifrar:

Para el descifrado se siguen los mismos pasos del algoritmo de encriptación de imágenes excepto en los pasos 4-6, en el cual se debe usar el método de descifrado de RSA y cuyos parámetros son la intensidad del píxel en una determinada posición de un componente de la imagen y la clave privada respectiva. La complejidad de esta función es la misma que se utiliza en el anterior algoritmo, dependiendo de la función RSA.

Requiere: *ImagenCifrada*: imagen a descifrar, *n*: clave publica, *d*: clave privada
Asegurar: *CriptoIm*: Resultado de cifrar Imagen
1: Mientras *i* < número de filas de Imagen hacer
2: Mientras j < número de columnas de Imagen hacer
3: $Pixel \leftarrow ImagenCifrada[i,j]$
4: $R1 \leftarrow RSA(Pixel.R, d, n) - 10$
5: $G1 \leftarrow RSA(Pixel.G, d, n) - 10$
6: $B1 \leftarrow RSA(Pixel.B, d, n) - 10$
7: $CriptoIm[i,j] \leftarrow RGB(R1\%256, G1\%256, B1\%256)$
8. Fin Mientras
9. Fin Mientras
10. Devolver CriptoIm

C. Arquitectura

La arquitectura que se utilizará para el funcionamiento del algoritmo RSA será implementado por dos capas, una interfaz visual y clases de control, para el modelamiento se utiliza el Lenguaje Unificado de Modelado (UML), considerando los siguientes diagramas:

a) Diagrama de colaboración

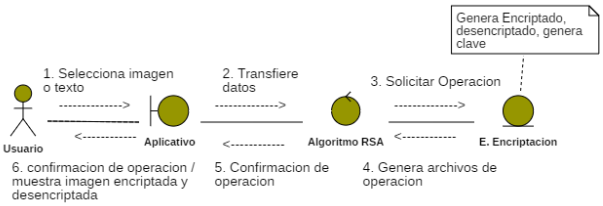

Figura 2. Diagrama de colaboración

b) Prototipo de interfaz

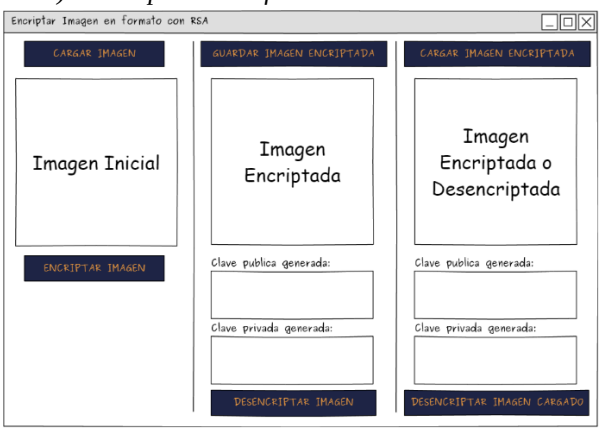

Figura 3. Prototipo de interfaz del aplicativo

IV. Diseño de Aplicativo

El aplicativo para Algoritmo criptográfico RSA para encriptación y desincriptación de imágenes se desarrollo en lenguage de programacion C Sharp, teniendo como producto final:

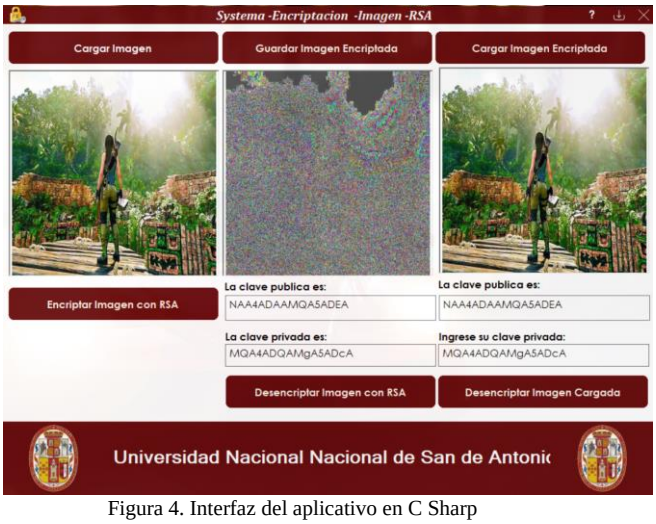

Figura 4. Interfaz del aplicativo en C Sharp

El aplicativo contiene un formulario principal que contiene botones para cargar una imagen, Encriptar imagen con RSA, Guardar imagen encriptada, Desencriptar imagen

con RSA, Cargar imagen encriptada y Desencriptar imagen cargada en formato RSA. Con el botón ‘Cargar una imagen’ exportamos una imagen de cualquier formato del almacenamiento interno o externo, esta imagen será encriptada con el botón ‘Encriptar imagen con RSA’, este botón instancia al módulo RSA_Encriptar() el cual utiliza el módulo power que sirve para calcular los factores primos, la estructura que genera un arreglo de números primos grandes y la función totiente; este botón como resultado genera una imagen encriptada, esta imagen encriptada con el botón Guardar imagen encriptada se guarda con extensión RSA generando un clave primaria y una clave publica; estas claves por definición del algoritmo RSA sirven para compartir de forma segura una imagen. Con el botón Cargar imagen encriptada se carga una archivo con extensión RSA, para procesar este archivo el receptor necesita de una clave publica y privada.

V. Validación de resultados y discusión

Es un algoritmo muy simple, pero el verdadero desafío será de generar la clave pública y la clave privada que se basan en números primos grandes y también no pueden ser arbitrarias. Si elegimos números primos pequeños, entonces puede ser posible un ataque de fuerza bruta, pero si elegimos un gran número de números primeros se notará la complejidad. El uso de este algoritmo se ha extendido hasta tal punto que se publicó el RSA Cryptography Standard, que contiene las recomendaciones para la implementación de métodos criptográficos de clave pública basados en el algoritmo. Esta documentación contiene algunas nociones básicas sobre cifrado y descifrado de imagen, Además contiene algunos esquemas donde se describen las operaciones de cifrado y el proceso inverso. Uno de las ventajas de este algoritmo es que se puede usar para el manejo de las llaves y resuelve el problema en la distribución de las llaves, más allá de utilizar algoritmos seguros con altos niveles de protección es necesario que los usuarios sean cuidadosos en el uso de las contraseñas y precavidos en cuanto el tipo de sitios que utilizan para intercambiar información sensible.

a) Entrada:

El aplicativo tiene dos entradas: Una imagen a encriptar en cualquier formato o una imagen encriptada adjunto con su clave privada

b) Salida:

El aplicativo tiene dos salidas: Una imagen encriptada que puede ser exportado con extensión RSA junto a su clave publica y privada o una imagen original producto de una imagen encriptada adjunto con su calve privada.

c) Matriz de evaluación

VI. Referencias

[1] N. Y. Goshwe, "Data Encryption and Decryption Using RSA Algorithm in a Network Environment, Engineering University of Agriculture, Makurdi," 2013.

[2] R. S. S. C. Balram Swami, "Dual Modulus RSA Based on Jordan-totient Function," pp. 1581-1586, 2016.

[3] A. A. D. P. R. K. S. K. D. Kanika Sharma, "RSA based encryption approach for preserving confidentiality of big data," *Journal of King Saud University - Computer and Information Sciences*, pp. 1319-1578, 2019.

[4] C.-H. Lin, J.-C. Liu and C.-C. Li, "Speeding Up RSA Encryption Using GPU Parallelization," *5th International Conference on Intelligent Systems, Modelling and Simulation*, pp. 529-533, 2014.

[5] S. a. G. D. Mathur, "A Modified RSA Approach for Encrypting and Decrypting Text and Images Using Multi-Power, Multi Public Keys, Multi Prime Numbers and K-Nearest Neighbor Algorithm," 2016.

[6] C. B., "Criptografía, MAPLE y RSA," *Escuela Tecnica Superior de Ingenieros de Telecomunicación*, 1998.

[7] D. P. S. S. Fernando Solís, "Seguridad de la información en el intercambio de datos entre dispositivos móviles con sistema Android utilizando el método de encriptación RSA," *Enfoque UTE*, 2017.

[8] M. Kumar, "Advanced RSA Cryptographic Algorithm for Improving Data Security," 2018.

[9] E. Milanov, "The RSA Algorithm," 2009.

[10] A. S. a. L. A. R.L. Rivest, "A Method for Obtaining Digital," *Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge*, 1983.

[11] J. M. A. D. L.Hernández Encinas, "Large decryption exponents in RSA," pp. 293-295, 2003.

[12] P. F. &. A. C. R. P. Mora Mejía, "ANÁLISIS DE EFICIENCIA Y CALIDAD ENTRE EL ALGORITMO AES CBC Y EL MAPA DE ARNOLD PARA EL CIFRADO DE IMÁGENES DIGITALES (Master's thesis)," 2018.

[13] A. l. P. Kumar M., "A new RGB image encryption algorithm based on DNA encoding and elliptic curve Diffie–Hellman cryptography,Signal Processing," pp. 87-202, 2004.

[14] I. &. S. I. Ozturk, "Analysis and comparison of image encryption algorithms. International Journal of Information Technology," pp. 108-111, 2004.

[15] I. P. W. Group, "Standard specifications for public key cryptography," vol. IEEE P1363/D20 (Draft Version 20), 2005.

[16] J. F. A. Pita, "FUNDAMENTOS MATEMATICOS DEL ALGORITMO RSA," *Universidad Autonoma de Guerrero, Mexico*, 2018.

[17] S. R. a. i. R. c. Katzenbeisser, "Springer Science+Business Media, USA," 2001.

[18] J. V. J. P. S. &. V. J. G. Rebaza12, "Método para la Aplicación de Esquemas de Clave Pública al Cifrado de Imágenes RGB".