Preparación de Artículos revista Vínculos: Ciencia, Tecnología y Sociedad
Fecha de envío: 05/11/2021
Fecha de recepción: 09/11/2021
Fecha de aceptación: 21/12/2021

Análisis del rendimiento entre los algoritmos simétricos de Blowfish y AES

Performance analysis between Blowfish and AES symmetric algorithms

Doris Maritza Ruano-Daza[1], Brayan David Valiente-Simbaqueba[2], Luis Fernando Guevara-Triana[3], Maicol Stiven Pérez-Poblador[4]

Resumen: La información es un recurso esencial para el desarrollo de proyectos que requieren la comunicación constante entre un emisor y receptor, a partir de un conjunto de protocolos que permiten la interacción entre diversas aplicaciones de manera asíncrona. Sin embargo, presentan riesgos de seguridad como el robo, destrucción, filtración y modificación, que pueden traer consecuencias a nivel personal y organizacional. Para garantizar que esa información cumpla con los principios de confidencialidad, integridad y disponibilidad es necesaria la aplicación de un algoritmo criptográfico. Por tal razón, se realizó una comparación del rendimiento con respecto al encriptado y desencriptado de los algoritmos AES (Advanced Encryption Standard) y el algoritmo Blowfish; empleando el lenguaje de programación Java y un conjunto de librerías para la

[1] Tecnólogo en Sistematización de Datos, Universidad Distrital Francisco José de Caldas, Bogotá-Colombia. Correo electrónico dmruanod@correo.udistrital.edu.co
[2] Tecnólogo en Sistematización de Datos, Universidad Distrital Francisco José de Caldas, Bogotá-Colombia. Correo electrónico bdvalientes@gmail.com. ORCID: https://orcid.org/0000-0003-1728-5287
[3] Tecnólogo en Sistematización de Datos, Universidad Distrital Francisco José de Caldas, Bogotá-Colombia. Correo electrónico luigtria@gmail.com ORCID: https://orcid.org/0000-0002-3260-8657
[4] Tecnólogo en Sistematización de Datos, Universidad Distrital Francisco José de Caldas, Bogotá-Colombia. Correo electrónico maeperezp@correo.udistrital.edu.co

ilustración de gráficos, basados en el tamaño del archivo y el tiempo que tarda en ejecutarse los dos procesos; permitiendo elegir el más adecuado, acorde a las necesidades de los interesados.

Palabras clave: AES, Algoritmo de clave simétrica, Algoritmo de criptográfico, Blowfish.

Abstract*:* Information is an essential resource for the development of projects that require constant communication between a sender and receiver, based on a set of protocols that allow interaction between various applications asynchronously. However, they present security risks such as theft, destruction, leakage and modification, which can have consequences on a personal and organizational level. To guarantee that this information complies with the principles of confidentiality, integrity and availability, the application of a cryptographic algorithm is necessary. For this reason, a performance comparison was made with respect to encryption and decryption of the AES (Advanced Encryption Standard) algorithms and the Blowfish algorithm; using the Java programming language and a set of libraries for graphic illustration, based on the size of the file and the time it takes to execute the two processes; allowing to choose the most suitable, according to the needs of the interested parties.

Keywords: AES, Symmetric-key algorithm, Cryptographic algorithm, Blowfish.

1. Introducción

Debido al avance de las tecnologías, la comunicación ha sufrido un cambio drástico con respecto a la forma como se envía y se recibe la información, se pasó de un entorno físico a uno prácticamente virtual; donde los datos viajan a través de un canal y se distribuyen dependiendo del sistema o proceso que los requieran. Por tal motivo, esto ha provocado que surjan nuevas vulnerabilidades en cuanto a la seguridad y protección de la información; como la manipulación por parte de terceros, suplantación de identidad, robo y modificación. Una medida para mitigarlas

es a través de la implementación de algoritmos criptográficos que según Álvaro Gómez realiza unas transformaciones sobre el texto original, conocido como texto claro, para obtener un texto modificado, conocido como texto cifrado. Asimismo, mediante el procedimiento inverso, se puede recuperar el texto original [1], con el fin de alcanzar características de confidencialidad, integridad y autenticación. Por otra parte, los algoritmos criptográficos se dividen en simétricos, que utilizan la misma clave tanto para cifrar como para descifrar; en asimétricos, que requieren una clave publica y una privada para el cifrado y descifrado; y en funciones hash que "por medio de un proceso matemático convierten los datos en caracteres de longitud fija" [2].

Partiendo de lo anterior, se eligieron dos algoritmos simétricos: el AES (Estándar de cifrado avanzado) que es uno de los más populares y el Blowfish que "es el sustituto principal de los algoritmos DES o IDEA" [3], con el objetivo de establecer un análisis de rendimiento para los procesos de encriptación y desencriptación de archivos, teniendo como variables el tamaño y el tiempo. Esta comparación se hará aplicando el lenguaje de programación de Java junto con algunas librerías que permiten desarrollar de manera más ágil este tipo de algoritmos; además, otras para la visualización de graficas con los resultados obtenidos al realizar la carga de los archivos. También, se quiere llegar a determinar cuál algoritmo es más efectivo y observar los diversos comportamientos de acuerdo con las propiedades establecidas en Megabytes (MB) de los ficheros.

1.1. Algoritmos Criptográficos

"Son algoritmos que pueden encriptar texto en lenguaje natural para hacerlo ilegible, y para que sea desencriptado con el fin de recuperar el texto original; fueron diseñados para resistir ataques de diferente índole" [4]. Además, funcionan de dos maneras; un cifrado por bloques, que agrupa los bits en cuadros de longitud fija; y un cifrado de flujo, donde los dígitos de texto se combinan con un conjunto de claves aleatorias que son cifradas una a la vez (se utiliza para audio o video).

Por otro lado, estos algoritmos se clasifican en criptografía simétrica, criptografía asimétrica y algoritmos HASH.

1.2. Algoritmo de clave simétrica

"Conjunto de algoritmos diseñados para cifrar un mensaje utilizando una única clave conocida tanto por el emisor como el receptor, de manera que el documento cifrado solo puede descifrarse conociendo dicha clave secreta" [5]. Los algoritmos más usados son: DES, TRIPLEDES, IDEA, BLOWFISH y AES.

1.3. Advanced Encryption Standard (AES)

Es un algoritmo de cifrado simétrico desarrollado por Vicent Rijeme y Joan Daemen criptógrafos belgas. Fue presentado en 1997 bajo el nombre de "Rindael" en el concurso que organizó el Instituto Nacional de Normas y Tecnologías (NIST) y fue elegido el mejor algoritmo de cifrado; es uno de los más utilizados actualmente. El gobierno de los Estados Unidos anunció en el año 2003 que el algoritmo era muy seguro y podía ser usado para la protección nacional de información y hasta la fecha no se conocen ataques eficientes.

"Las características del algoritmo AES es que procesa bloques completos de texto de 128 bits y maneja claves estándar de 128, 192 o 256 bits" [6], se vale de una matriz de estados cuyas celdas van cambiando de valor según los procesos que ejecuta el algoritmo usando sustitución o permutación en otros casos operaciones polinómicas.

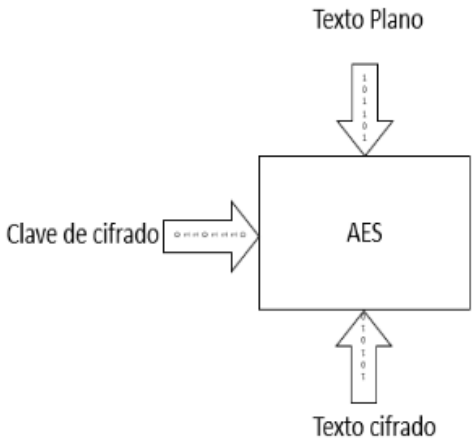
Figura 1. Matriz de estados algoritmo AES.

	0	1	2	3
0	S (0,0)	S (0,1)	S (0,2)	S (0,3)
1	S (1,0)	S (1,1)	S (1,2)	S (1,3)
2	S (2,0)	S (2,1)	S (2,2)	S (2,3)
3	S (3,0)	S (3,1)	S (3,2)	S (3,3)

Fuente: Elaboración propia.

"Este algoritmo consta de once rondas las cuales son 1 ronda inicial, 9 rondas que pasan por 4 fases y 1 ronda final de 3 fases" [7]. Para iniciar con el cifrado debemos tener como entrada el texto claro y la clave de cifrado lo que nos arrojan como resultado el texto cifrado.

Figura 2. Matriz de estados algoritmo AES.

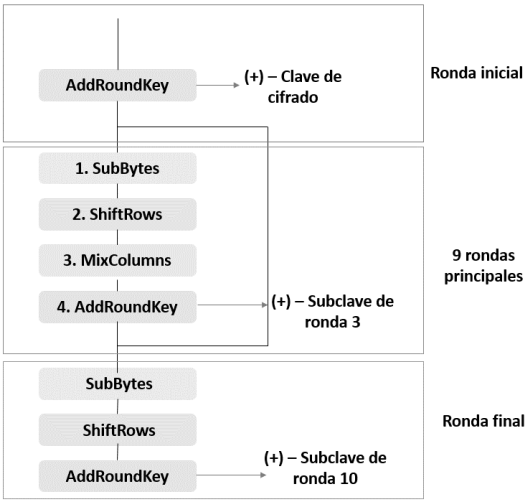


Fuente: Elaboración propia.

El estado y la clave de cifrado debe estar en notación hexadecimal, el cifrado AES consta de dos partes: proceso de cifrado y el cálculo de subclaves.

- **Proceso de cifrado**
  - **Ronda inicial:** se tiene el estado el cual debe pasar por una función a una operación llamada AddRoundkey que hace una operación XOR junto a la clave de cifrado.
  - **9 rondas principales**: se deben llevar las operaciones SubBytes, ShiftRows, MixColumns y AddRoundKey las cuales se explican más adelante.
  - **Ronda final**: se aplica la operación SubBytes, ShiftRows y AddRoundkey y se utiliza la subclave de la ronda 10 aplicando la operación XOR [7].

Figura 3. Proceso de Cifrado.



Fuente: Elaboración propia.

Los tipos de transformación de cifrado son cuatro [8]:

- **SubByte:** se tiene una matriz 4X4 y una tabla de sustitución de bytes (ver figura 4) se obtiene una nueva matriz así: tomamos la primera posición de la matriz de estado y su primer valor lo reemplazamos por el valor de x en la tabla de sustitución de bytes y el segundo valor en y, y así con cada posición en la matriz de estados.

- **ShiftRows:** se rotará 1 byte es decir una celda en la segunda fila, para la tercera fila se rotarán 2 bytes o lo que es igual a dos celdas y para la 4 fila 3 bytes o lo que es igual a tres celdas.
- **MixColumns:** se toma una columna y se multiplica con un matriz determinada y este resultado es el que se usan en la nueva matriz se realiza lo mismo con las tres columnas siguientes.
- **AddRoundkey:** se tiene un bloque inicial y la subclave de la ronda y consiste en tomar la primera columna de la matriz inicial con la primera columna de la subclave y se aplica XOR.

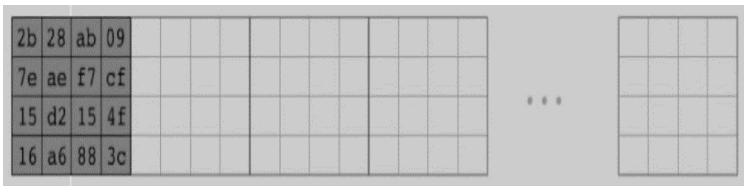
Figura 4. Tabla de sustitución de bytes [9].

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

- **Cálculo de subclaves**

Se tiene 1 ronda inicial, 9 rondas principales y 1 ronda final por lo tanto se requieren 11 subclaves parciales y se debe completar las siguientes 4 matrices (ver figura 5).

Figura 5. Cálculo de subclaves [7].



- Primero se tiene la clave de cifrado de 128 bits.
- Para la columna número 5 se debe tomar la columna número 4 y se aplica la operación llamada RotWord la cual cambia las celdas.
- Una vez intercambiada se usa la transformación Subbytes y se asigna el valor que corresponda al igual que el proceso anterior.
- Luego se realiza una operación XOR y el resultado se ubica en la columna 5.

1.4. Blowfish

Es un algoritmo criptográfico de clave simétrica que se puede utilizar como reemplazo directo de DES o IDEA. Fue desarrollado por Bruce Schneider, un asesor y criptógrafo independiente.[3]

- **Propiedades**

Como propiedades se encontraron las siguientes:

Tabla 1. Propiedades del algoritmo Blowfish.

Propiedad	**Descripción**
Cifrado de bloque	64 bits
Longitud de clave	Tamaño variable de 32 bits a 448 bits.
N.º de subclaves	18
N.º de casillas de sustitución	4, cada una con 512 entradas de 32 bits.

Fuente: elaboración propia.

- **Características**

En la tabla 2 se ilustran las características más relevantes que engloban al algoritmo de Blowfish.

Tabla 2. Características algoritmo Blowfish.

Concepto	Descripción
Rápido	Estado de cifrado blowfish en microprocesadores de 32 bits.
Compacto	Puede ejecutarse en menos de 5 KB de memoria.
Simple	Usa solo operaciones como la suma XOR y la búsqueda de tablas
Seguro	Todavía no existe un ataque practico contra el cifrado.
Libre	No esta sujeto a ninguna patente.

Fuente: elaboración propia.

- **Procesos**

  El algoritmo se divide en dos etapas:

  - **Generación de subclaves:** convierte la clave hasta 448 bits en subclaves que suman un total de 7168 bits.
  - **Cifrado de datos:** se itera una función simple 16 veces. Cada ronda contiene una permutación dependiente de la clave y una sustitución de claves y datos .

- **Algoritmo**

  Pasos cifrado y descifrado:

  1 Un mensaje de texto sin formato de 64 bits se divide primero en 32 bits.

2 Con los 32 bits “izquierdos” se hace un XOR con el primer elemento de la matriz P para crear un valor llamado Pi, se ejecutará una función de transformación llamada F.

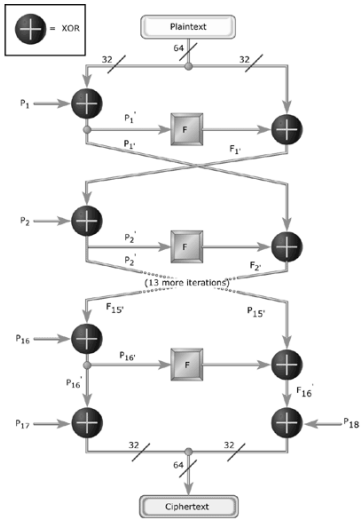
3 Luego se hace un XOR con los 32 bits “derechos” del mensaje para producir un nuevo valor llamado F.

4 F luego reemplaza la mitad “izquierda” del mensaje y P reemplaza la mitad “derecha”.

5 El proceso se repite 15 veces más con miembros sucesivos de la matriz P.

6 Los P y F resultantes se someten a XOR con las dos ultimas entradas de la matriz P (entradas 17 y 18) y se recombinan para producir el texto cifrado de 64 bits.

Figura 6. Proceso de cifrado algoritmo Blowfish [10].

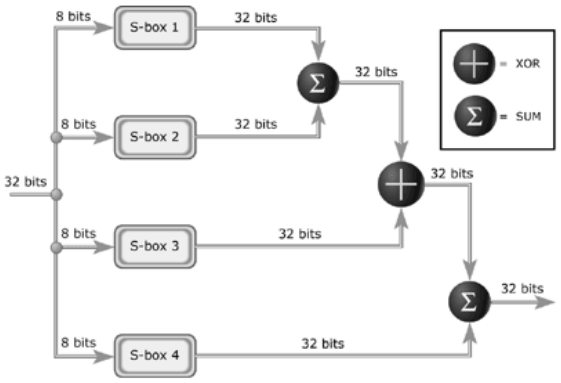


Pasos Función F:

1 La función divide una entrada de 32 bits en cuatro bytes y los usa como índices en una matriz S.

2 Luego, los resultados de la búsqueda se agregan y se aplican XOR juntos para producir la salida.

Figura 7. Función F algoritmo Blowfish [10].



Nota: dado que el algoritmo Blowfish es simétrico, se utiliza el mismo proceso para el descifrado y el cifrado. La única diferencia es que la entrada al cifrado es texto sin formato y para el descifrado sería el texto cifrado[12].

2. Análisis y construcción de los algoritmos AES y Blowfish

En el marco teórico se abordaron la funcionalidades y las etapas que realizan tanto el algoritmo AES como el algoritmo Blowfish para los procesos de cifrado y descifrado de la información, teniendo presente lo anterior, ahora se va a explicar el proceso de desarrollo de software que se tuvo en cuenta para implementar dichos algoritmos.

- **Características del equipo donde se realizaron las mediciones**

  Para hacer esta medición se contó con un equipo con las siguientes propiedades.

Tabla 3. Propiedades del equipo de medición.

Propiedad	Descripción
Sistema operativo	Windows 8.1
Procesador	Intel Corei5 – tercera generación
Frecuencia	2.40 GHz
Memoria RAM	12GB

Fuente: elaboración propia.

- **Desarrollo de software**

  Dentro de este análisis se desarrolló un software que tiene como fin la implantación de los algoritmos simétricos AES y Blowfish; utilizando el lenguaje de programación Java, junto con la aplicación de la librería **Cipher**, que permite utilizar instancias de su clase principal e ir manejando los algoritmos de encriptación que tiene creados. Para entrar más a fondo con el código fuente, se irán citando algunos fragmentos y explicando su funcionamiento.

  - **Implementación algoritmo AES**

  Para llevar a cabo la construcción del algoritmo AES, se creó una clase llamada **CryptoAES**, en esta clase se establecerán las siguientes funciones:

  1 **Función (encrypt):** esta función recibe tres parámetros (key, file1, file2), donde:

  1.1 key: es el valor que va a utilizar el algoritmo para encriptar el archivo.

  1.2 File1: el archivo a encriptar.

  1.3 File2: nuevo archivo encriptado.

Figura 8. Función encrypt AES.

```
public static void encrypt(String key, File inputFile, File outputFile)
        throws CryptoException {
    doCrypto(Cipher.ENCRYPT_MODE, key, inputFile, outputFile);
}
```

Fuente: Elaboración propia.

2 **Función (decrypt):** esta función recibe tres parámetros (key, file1, file2), donde:

2.1 key: es el valor que va a utilizar el algoritmo para encriptar el archivo.

2.2 File1: el archivo a desencriptar.

2.3 File2: nuevo archivo desencriptado.

Figura 9. Función decrypt AES.

```
public static void decrypt(String key, File inputFile, File outputFile)
        throws CryptoException {
    doCrypto(Cipher.DECRYPT_MODE, key, inputFile, outputFile);
}
```

Fuente: Elaboración propia.

3 **Función (doCrypto):** esta función recibe cuatro parámetros (cipherMode, key, file1, file2), donde:

3.1 cipherMode: es el valor que indicará si el proceso es encriptar o desencriptar.

3.2 key: es el valor que va a utilizar el algoritmo para encriptar el archivo.

3.3 File1: el archivo a encriptar o desencriptar, dependiendo del valor de **cipherMode**.

3.4 File2: el nuevo archivo generado dependiendo del valor de **cipherMode**.

Figura 10. Función doCrypto AES.

```
private static void doCrypto(int cipherMode, String key, File inputFile,
        File outputFile) throws CryptoException {
    try {
        Key secretKey = new SecretKeySpec(key.getBytes(), ALGORITHM);
        Cipher cipher = Cipher.getInstance(TRANSFORMATION);
        cipher.init(cipherMode, secretKey);

        FileInputStream inputStream = new FileInputStream(inputFile);
        byte[] inputBytes = new byte[(int) inputFile.length()];
        inputStream.read(inputBytes);

        byte[] outputBytes = cipher.doFinal(inputBytes);

        FileOutputStream outputStream = new FileOutputStream(outputFile);
        outputStream.write(outputBytes);

        inputStream.close();
        outputStream.close();

    } catch (NoSuchPaddingException | NoSuchAlgorithmException | InvalidKeyException |
            BadPaddingException | IllegalBlockSizeException | IOException ex) {
        throw new CryptoException("Error encrypting/decrypting file", ex);
    }
}
```

Fuente: Elaboración propia.

Dentro de la clase, se inicializan dos variables, que indican el algoritmo que se va a implementar.

Figura 11. Inicialización de variables para el algoritmo AES.

```
private static final String ALGORITHM = "AES";
private static final String TRANSFORMATION = "AES";
```

Fuente: Elaboración propia.

- **Implementación algoritmo Blowfish**

  Para llevar a cabo la construcción del algoritmo Blowfish, se creó una clase llamada **Crypto Blowfish**, en esta clase se establecen las mismas funciones que se explicaron anteriormente para el algoritmo AES (**encrypt, decrypt, doCrypto**) pero se debe agregar la variable que indican el tipo de algoritmo a implementar que sería Blowfish:

Figura 12. Inicialización de variables para el algoritmo Blowfish.

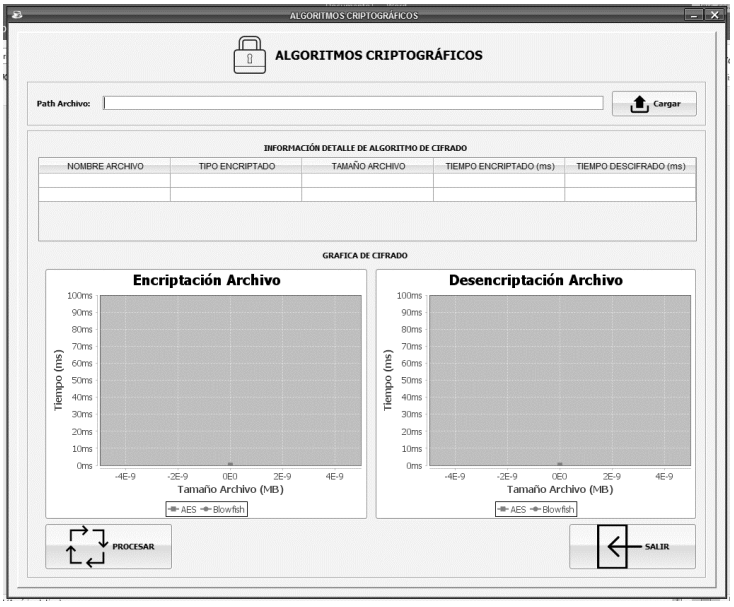
```
private static final String ALGORITHM = "Blowfish";
```

Fuente: Elaboración propia.

3. Ejecución del software de medición

Se creó una interfaz aplicando la librería **Look and Feel** y dos gráficos usando la librería **JFreeChart**. El primer grafico corresponde al cifrado del archivo y el segundo al descifrado. Asimismo, el eje "Y" representa el tiempo de ejecución medido en (milisegundo ms), y el eje "X" el tamaño del archivo a encriptar. Por otra lado, se encuentra una tabla donde se evidencia la información con las propiedades de los archivos que se cargaron: nombre, tipo de encriptado, el tamaño, el tiempo de encriptación y el tiempo de desencriptación.

Figura 13. Interfaz gráfica del software.

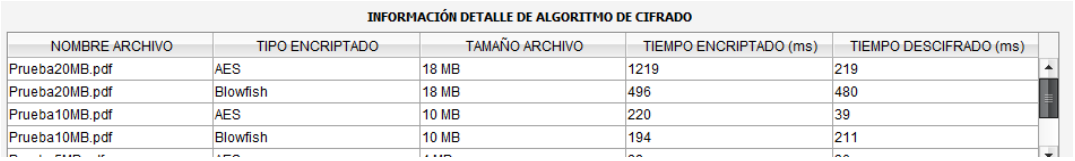


Fuente: Elaboración propia.

Para la prueba se utilizaron 4 archivos de diferentes tamaños 1MB, 4MB, 10MB y 18MB; cabe aclarar que el software soporta tanto ficheros con un tamaño más grande, como para ficheros de un tamaño más pequeño. Posteriormente, estos archivos se van cargando al programa donde de manera automática se va a diligenciando la tabla con la información de cada archivo.

- **Archivos de tamaño de 18MB y 10 MB**

Figura 14. Información de archivos cargados de 18MB y 10MB.



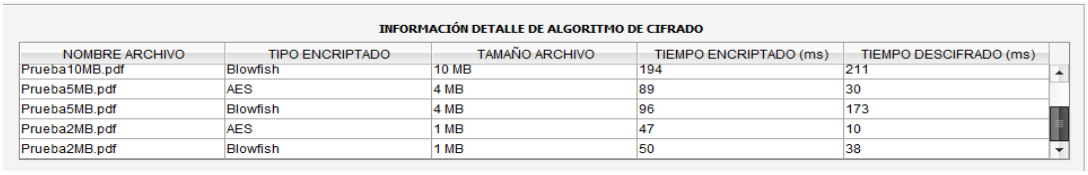
INFORMACIÓN DETALLE DE ALGORITMO DE CIFRADO

NOMBRE ARCHIVO	TIPO ENCRIPTADO	TAMAÑO ARCHIVO	TIEMPO ENCRIPTADO (ms)	TIEMPO DESCIFRADO (ms)
Prueba20MB.pdf	AES	18 MB	1219	219
Prueba20MB.pdf	Blowfish	18 MB	496	480
Prueba10MB.pdf	AES	10 MB	220	39
Prueba10MB.pdf	Blowfish	10 MB	194	211

Fuente: Elaboración propia.

- **Archivos de tamaño de 4MB y 1MB**

Figura 15. Información de archivos cargados de 4MB y 1MB.



INFORMACIÓN DETALLE DE ALGORITMO DE CIFRADO

NOMBRE ARCHIVO	TIPO ENCRIPTADO	TAMAÑO ARCHIVO	TIEMPO ENCRIPTADO (ms)	TIEMPO DESCIFRADO (ms)
Prueba10MB.pdf	Blowfish	10 MB	194	211
Prueba5MB.pdf	AES	4 MB	89	30
Prueba5MB.pdf	Blowfish	4 MB	96	173
Prueba2MB.pdf	AES	1 MB	47	10
Prueba2MB.pdf	Blowfish	1 MB	50	38

Fuente: Elaboración propia.

Tabla 4. Resultados del cargue de archivos.

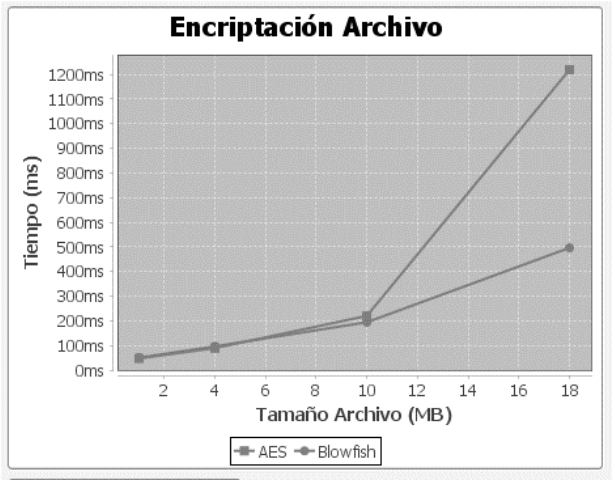
Algoritmo	Tamaño (MB)	Tiempos (ms)	
		Encriptación	Desencriptación
AES	(+/-) 1MB	47	10
	(+/-) 4MB	89	30
	(+/-) 10MB	220	39
	(+/-) 18MB	1219	219
BLOWFISH	(+/-) 1MB	50	38
	(+/-) 4MB	96	173
	(+/-) 10MB	194	211
	(+/-) 18MB	496	480

Fuente: elaboración propia.

Los resultados anteriores se graficaron en un diagrama lineal, con la intención de notar las variaciones que sufren y hacer un comparativo entre estos dos algoritmos, se encontró lo siguiente:

- **Tiempo de ejecución, para el encriptado del archivo**

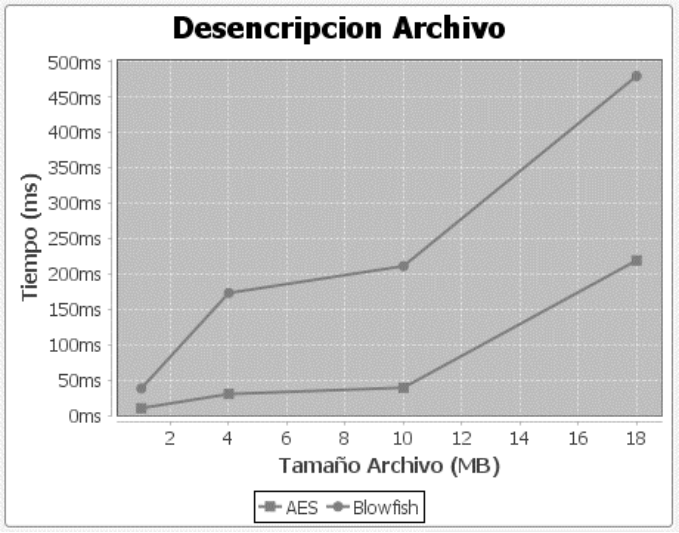
Figura 16. Gráfico del tiempo de encriptación del archivo.



Fuente: Elaboración propia.

- **Tiempo de ejecución, para el Desencriptado del archivo.**

    - **Figura 17.** Gráfico del tiempo de desencriptación del archivo.



Fuente: Elaboración propia.

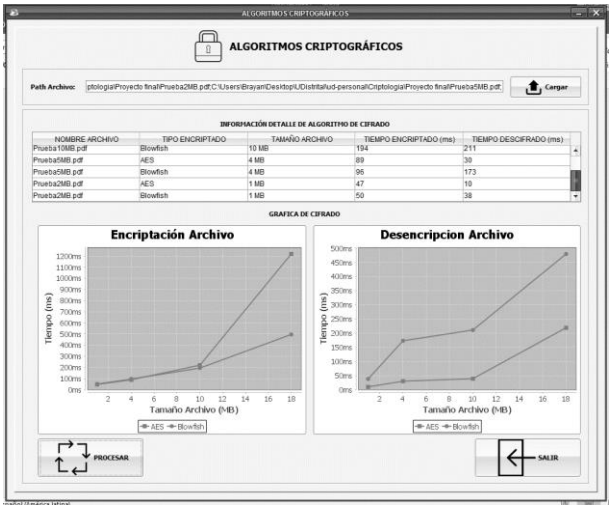
Para el proceso de encriptación del archivo, se puede observar en la figura 16 que de acuerdo con las variables de tiempo (ms) y tamaño del archivo (MB), para los archivos menores de 10 MB la diferencia en cuanto a tiempo no pasa de los 10 ms. Sin embargo, con archivos de mayor tamaño si existe una variación mayor; siendo el algoritmo AES un poco más lento en realizarlo, por lo que se puede afirmar que Blowfish es más eficiente. Por otro lado, para el proceso de desencriptado del archivo, figura 17, el algoritmo AES es mucho más rápido que el algoritmo Blowfish.

4. Eficacia de los algoritmos

Dentro del análisis que se realizó, se puede concluir que el algoritmo de Blowfish, trabaja más rápido durante el proceso de encriptación, pero es menos eficiente a la hora de realizar la tarea de desencriptación. De igual manera, el algoritmo AES presenta un comportamiento inverso, es más lento durante el proceso de encriptación, pero rinde más en la etapa de desencriptación. Por tal razón, para estas pruebas que se efectuaron los algoritmos tendrían una eficiencia similar, ya que

el tiempo que tarda uno en encriptar el otro lo contrarresta al momento de desencriptarlo y viceversa.

Figura 18. Resultados completos de la encriptación y desencriptación.



Fuente: Elaboración propia.

NOTA: Para su defecto de ser replicada esta prueba, se debe tener en cuenta, la capacidad o características que se tiene en el hardware, ya que esto afectara el rendimiento de uno u otro algoritmo.

5. Conclusiones

Con la aplicación de los algoritmos criptográficos se pueden reducir vulnerabilidades y problemáticas de seguridad que afectan el envío de la información a través de los sistemas que establecen una comunicación entre un emisor y un receptor. Asimismo, estos se pueden implementar en diversas herramientas ya sea a nivel empresarial o en la cotidianidad.

Por otra parte, gracias al uso de la librería de java llamada **Cipher** la implementación de los algoritmos AES y Blowfish fue cómoda, debido a que ya se tienen un conjunto de funciones tanto para el proceso de encriptación como para el de desencriptación. Con respecto a la comparación de

los dos algoritmos, se determinó que en el proceso de encriptación de los archivos el algoritmo AES es un poco más lento que el Blowfish pero en la desencriptación lo supera, por lo que se puede decir que son similares; cabe aclarar que en esta parte hay que tener en cuenta otros parámetros como el tamaño de los bloques, la longitud de la claves y las rondas que realizan.

Para futuras mejoras se podría hacer la comparación con varios algoritmos no solo de clase simétrica sino también con los asimétricos. Además, adicionar más variables para tener un análisis detallado y unas graficas robustas.

Referencias

[1] A. Gómez Vieites, " Funcionamiento de un sistema criptográfico", Madrid: RA-MA, pp. 16, 2014. [En línea]. Disponible: https://books.google.com.co/books?id=Fo-fDwAAQBAJ&pg=PA24&dq=que+es+un+Algoritmo+de+cifrado&hl=es&sa=X&ved=2ahUKEwidid_VpfnuAhUyq1kKHWaKBAYQ6AEwBHoECAkQAg#v=onepage&q&f=true

[2]. B. Donohue, "¿Qué es un Hash y como funciona?", 2018. [En línea]. Disponible: https://latam.kaspersky.com/blog/que-es-un-hash-y-como-funciona/2806/

[3] Schneier on Security, "The Blowfish Encryption Algorithm", 2019, [En línea]. Disponible: https://www.schneier.com/academic/blowfish/

[4] MDN Web, "Algoritmo criptográfico", 2020. [En línea]. Disponible: https://www.schneier.com/academic/blowfish/

[5] V. Delgado, R. Palacios, "Introducción a la criptografía: tipos de algoritmos", Anales de mecánica y electricidad, Vol. 83, no. 1, pp 43, 2006.

[6]. O. Landeta, "Algoritmos de cifrado AES y DES", 2019. [En línea]. Disponible: https://www.youtube.com/watch?v=BctlBJ2NdHk

[7]. N. Cumbicos, "Funcionamiento del Algoritmo AES", 2020. [En línea]. Disponible: https://www.youtube.com/watch?v=xOlEWLH_jkY

[8]. A. Pousa, "Algoritmo de cifrado simétrico AES aceleración de tiempo de cómputo sobre arquitecturas multicore", 2021. [En línea]. Disponible: http://sedici.unlp.edu.ar/bitstream/handle/10915/4210/Documento_completo.pdf?sequence=1&isAllowed=y

[9]. Seguridades, "Advanced Encryption Standard (AES)", 2021. [En línea]. Disponible: http://seguridaredes.blogspot.com/2013/04/normal-0-21-false-false-false-es-x-none.html

[10]. B. Gatliff, "Encrypting data with the Blowfish algorithm", 2020. [En línea]. Disponible: https://www.embedded.com/encrypting-data-with-the-blowfish-algorithm/