Ciencia Latina
Revista Multidisciplinar

DOI: https://doi.org/10.37811/cl_rcm.v6i3.2269

Validación del modelo criptográfico de punto a punto basado en la obsolescencia programada usando ataque de fuerza bruta

Alan Rodrigo Corini Guarachi
https://doi.org/10.37811/cl_rcm.v6i3.2269
Universidad Nacional Siglo XX
La Paz - Bolivia

RESUMEN

La obsolescencia programada es un concepto que se utiliza en el área de comercial, productos alimenticios, equipos tecnológicos, entre otros, tienen la característica principal de tener un tiempo de vida útil, pasada la fecha el producto ya no puede ser consumido en el caso de los alimentos, en productos tecnológicos la obsolescencia programada ocasiona que el producto presente fallas, o se considere obsoleto.

La obsolescencia programada se empezó a utilizar en redes sociales, con el denominativo "Contenido efímero", el precursor fue SnapChat que utilizaba mensajes que duraban 24 horas, pasada la fecha el contenido se eliminaba sin dejar rastro del mismo, en los últimos años existen varias filtraciones de seguridad exponiendo series de televisión, películas de entretenimiento, entre otras filtraciones que ocasionaron perdidas millonarias a las empresas productoras, el concepto de obsolescencia programada puede ser aplicado al cifrado y descifrado de información, utilizando múltiples contraseñas hasta que la información sea publica, el modelo debe ser compatible con los algoritmos que se utilizan para cifrar la información.

Palabras clave: *seguridad; cifrado; descifrado; obsolescencia programada; combinaciones aleatorias; ataque de fuerza bruta*

Correspondencia: alan_corini_g@hotmail.com
Artículo recibido: 20 abril 2022. Aceptado para publicación: 05 mayo 2022.
Conflictos de Interés: Ninguna que declarar
Todo el contenido de **Ciencia Latina Revista Científica Multidisciplinar**, publicados en este sitio están disponibles bajo Licencia Creative Commons (cc) BY.
Como citar: Corini Guarachi, A. R. (2022). Validación del modelo criptográfico de punto a punto basado en la obsolescencia programada usando ataque de fuerza bruta. Científica Multidisciplinar, 6(3), 981-1000. DOI: https://doi.org/10.37811/cl_rcm.v6i3.2269

Validation of the end-to-end cryptographic model based on planned obsolescence using brute force attacks

ABSTRACT

Planned obsolescence is a concept that is used in the commercial area, food products, technological equipment, among others, they have the main characteristic of having a useful life time, after the date the product can no longer be consumed in the case of food, in technological products, programmed obsolescence causes the product to present failures, or to be considered obsolete.

Planned obsolescence began to be used in social networks, with the denomination "Ephemeral content", the precursor was SnapChat that used messages that lasted 24 hours, after the date the content was eliminated without leaving a trace of it, in recent years there are several security leaks exposing television series, entertainment movies, among other leaks that caused millionaire losses to production companies, the concept of programmed obsolescence can be applied to the encryption and decryption of information, using multiple passwords until the information is public, the model must be compatible with the algorithms that are used to encrypt the information..

Keywords: *security; encryption; decryption; planned obsolescence; random combinations; brute force attack.*

1. INTRODUCCIÓN

En la seguridad informática es necesario incorporar nuevos métodos de cifrado de información, en el año 2022 se incorpora Blockchain como Framework de ciberseguridad para IIoT escrito por Llante, Marceles, & Amador, (2022), que garantiza la integridad de la información.

Existen tres formas de cifrar la información, por ejemplo, la simétrico, asimétrico e hibrida como se detalla a continuación (Gutierrez, 2017):

1. El cifrado simétrico solo utiliza una clave para cifrar y descifrar el mensaje, que tiene que conocer el emisor y el receptor previamente y este es el punto débil del sistema, la comunicación de las claves entre ambos sujetos, ya que resulta más fácil interceptar una clave que se ha transmitido sin seguridad.
2. El cifrado asimétrica se basa en el uso de dos claves: la pública (que se podrá difundir sin ningún problema a todas las personas que necesiten mandarte algo cifrado) y la privada (que no debe de ser revelada nunca).
3. El cifrado hibrido es la unión del cifrado simétrico y asimétrico.

Existen varios algoritmos que gestionan solo una clave para cifrar la información, todo empezó con el código cesar, que consistía en mover el alfabeto en n posiciones, en la actualidad se manejan varios métodos por ejemplo empezaremos con el DES (Data Encryption Standard), es un algoritmo criptográfico diseñado para cifrar y descifrar datos utilizando bloques de 8 bytes y una clave de 64 bits. (IBM, 2009)

El algoritmo DES, después de varias pruebas y estudios se determinó que puede ser vulnerado en menos de 24 horas, de ahí es cuando nace el cifrado TDES, utiliza el algoritmo DES en tres ciclos, haciendo que la longitud se extienda a 112 bits, esta versión es más segura que el algoritmo anterior. (Elbinario.net, 2016)

Existen una gran variedad de algoritmos criptográficos, podemos mencionar los siguientes:

- Algoritmo AES es un algoritmo de cifrado por bloques, es decir los datos a cifrar se dividen en bloques de tamaño fijo (128 bits) y cada bloque se representa como una matriz de 4x4 bytes llamada estado (Pousa, Sanz, Naiouf, & Giusti, 2018)
- Blowfish es un algoritmo inventado por Bruce Schneier en 1993 de encriptación simétrica además de ser un codificador de bloques, el cual divide un mensaje en

bloques de una longitud de 64 bits durante la encriptación y la desencriptación. (Aravena, 2018)

- Algoritmo Twofish es un algoritmo de cifrado por bloques simétricos que fue publicado en el año 1998, esta relacionado con el algoritmo Blowfish, el algoritmo trabaja con bloques de 128 bits y una clave de 128 bits, 192 bits hasta 256 bits, el algoritmo utiliza el cifrado de Feistel, trabaja con una clave de 64 bits en cada iteración. (Bazzi, 2021)

Entre los algoritmos asimétrico, tenemos a los siguientes:

- El algoritmo ElGamal se desarrolló en base al protocolo Diffie-Hellman. La falta de una patente para un algoritmo lo convierte en una alternativa más económica que el algoritmo RSA. Las operaciones más complejas en exponenciación son las operaciones de módulo y las operaciones de multiplicación. (Zhumaniezov, 2018)
- Algoritmo MD5 es un algoritmo hash de 128 bits representado como un número de 32 dígitos hexadecimales, ha dejado de considerarse seguro, el año 2004 se anuncio la ruptura de la función hash MD5. (Romero & Leon, 2018)
- Algoritmo SHA es un conjunto de funciones de hash de cifrado. Existen varias versiones renovadas, la primera, la SHA-0 y la más reciente, la SHA-3. A partir de la versión SHA-2 están formados por diversas funciones como SHA-224, SHA-256, SHA-384 y SHA-512 la cual ha sido la más utilizada para la protección de datos y de información. (Elbinario10.net, 2016)
- Algoritmo SHA-3 se basa en permutaciones sin clave, utiliza un enfoque llamado construcción de esponja y compresión que es un modelo de permutación aleatoria, el algoritmo permite que la salida se extienda a cualquier longitud deseada. (Andrade, 2019)

En el año 2015, se realizó una comparación de los algoritmos DES, AES y 3DES, concluye que se necesita un tiempo necesario para utilizar todas las claves posibles en 50 mil millones segundos, éstos demostraron que AES es mejor que DES y 3DES. (Vargas & Yuri, 2015)

Project Zero el nombre del equipo de hackers de Google para mejorar la seguridad en Internet, ha realizado la primera colisión SHA-1 de la historia. SHA-1 es una función hash que permite crear una huella digital única de un archivo. SHA-1 es considerado teóricamente vulnerable desde 2005 y el Instituto Nacional de Estándares Tecnológicos

(NIST) lo ha considerado obsoleto desde 2011. Sin embargo, continúa utilizándose en innumerables entornos. (UPM, 2017)

SHA-1[1] fue creado en 1995 por la NSA, ganando mucha popularidad en los años posteriores, un grupo de investigadores de Google y del CWI de Amsterdam han realizado el primer ataque de colisión de hash a SHA-1. (Velasco, 2017)

El ataque de colisiones se hizo muy famoso en estas épocas con la esperanza de mostrar la ineficiencia del algoritmo una de las pruebas fue también con el algoritmo MD5. En el año 2007 un investigador llamado Nat McHugh encontró una vulnerabilidad muy grave en el hash MD5. A partir de dos imágenes totalmente diferentes sacadas de Internet consiguieron que gracias a un ordenador en pocas horas se computara un código idéntico MD5 en ambas imágenes. Esto fue posible atacando los principales puntos de inflexión de este tipo de hash y añadiendo una cadena binaria al final del archivo para conseguir engañar al código. (Velasco, 2017)

1.1. Generación de Contraseñas Seguras

La empresa Panda Security aconseja crear una contraseña utilizando 10 trucos empezando con el SUDOKU o partir de una frase. (Pandasecurity, 2016)

Existen varias formas de averiguar una contraseña empezando por los ataques más comunes como ser:

- Fuerza Bruta
- Diccionario
- Ataque de cumpleaños
- Pseudo-colisiones
- Complejidad lineal
- Ataques de correlación

Entre los menos comunes se encuentra el criptoanálisis lineal y el ataque por métodos algebraicos, aparte de estos métodos la contraseña puede ser descifrada por el usuario.

El ataque de fuerza bruta es el más popular, pero a medida que crece la longitud de la cadena, crece exponencialmente el número de combinaciones a probar de tal manera que genera una explosión combinacional. (Lopez, 2013)

[1] abreviatura de Secure Hash Algorithm 1

El usuario muchas veces se queja del robo de su información, de la pérdida de sus cuentas, sin darse cuenta que ellos mismos son quienes ocasionan este tipo de situaciones y malas experiencias. (Infante, 2013)

Para mejorar la seguridad se deberá utilizar dos o más contraseñas que se generan a partir de una y cifrar cada archivo con esta contraseña en un orden que proteja el archivo, el método de protección puede ser cualquiera, empezando del algoritmo DES o utilizar la protección de cualquier programa que solicite una contraseña para cifrar los archivos, todos los algoritmos utilizados para la criptografía serán vulnerados dentro de algunos años con software o hardware nuevo, pero si se refuerza el cifrado podremos garantizar una protección adicional, estas combinaciones se deberán determinar en un tiempo.

1.2. Ataque de Fuerza Bruta

Un ataque de fuerza bruta ocurre cuando el atacante emplea determinadas técnicas para probar combinaciones de contraseñas con el objetivo de descubrir las credenciales de una potencial víctima y así lograr acceso a una cuenta o sistema. Existen diferentes tipos de ataque de fuerza bruta, como el "credential stuffing", el ataque de diccionario, el ataque de fuerza bruta inverso o el ataque de password spraying. Generalmente, los ataques de fuerza bruta tienen mayor éxito en los casos en los que se utilizan contraseñas débiles o relativamente fáciles de predecir, En cuanto al hardware que se utiliza, cuanta más potencia se tenga más combinaciones por segundo se podrán evaluar, mientras que en lo que respecta el software, existen programas que son utilizados desde hace tiempo para aplicar la fuerza bruta en el descifrado de contraseña. (Albors, 2020)

El ataque de fuerza bruta se puede implementar en el lenguaje de programación Python para vulnerar unas contraseñas débiles de archivos zip, Mendoza (2021), Un ataque de fuerza bruta se puede detectar mediante su comportamiento y se puede detectar si se hace un monitoreo y análisis de los recursos del sistema, existen parámetros que identifican una saturación del sistema como indica la investigación de Mogollon, Gonzales, Sancho, & Jimenez, (2021), se implementan mecanismos de bloqueo a conexiones remotas que intentan obtener accesos utilizando el Framework Fail2ban que inhabilita las conexiones sospechosas, Barrientos (2021), la aplicación Suricata

permite detectar ataques de tipo Dos y Fuerza bruta cuando esta funcionado, Perdigon (2022), entre otras herramientas que permiten detectar un comportamiento anormal.

2. ESTRATEGIAS METODOLÓGICAS O MATERIALES Y MÉTODOS

El modelo criptográfico de punto a punto basado en la obsolescencia programada, se basa en utilizar múltiples contraseñas a partir de una, estas contraseñas se utilizan por un tiempo determinado que se requiera proteger el secreto, pasado ese tiempo, la información es publica para su uso, para validar el modelo se utilizara un generador de combinaciones y la probabilidad de vulnerar el modelo mediante el ataque de fuerza bruta.

2.1. Generación de combinaciones

Para generar combinaciones, se puede tomar como base la técnica de los anagramas, donde desordenamos los caracteres para descubrir una oración, este caso se planea definir en varios niveles, como por ejemplo se muestra siguiente:

- Movimientos en segunda dimensión
- Operaciones algebraicas
- Movimientos en tercera dimensión

Para poder generar las combinaciones a partir de una contraseña se deberá seguir los siguientes pasos:

Paso 1

Generar la clave simétrica o asimétrica, solo se podrá utilizar una clave para este fin, se puede utilizar cualquier algoritmo anteriormente descrito por ejemplo DES.

Paso 2

En este paso hay que definir las combinaciones que tiene que generar en un tiempo t. además, deberá generar el tiempo de procesamiento, estas dos variables definirán la cantidad como indica la siguiente formula:

$$N_c = \frac{T}{R}$$

Donde

N_c es el número de combinaciones

T es el tiempo

R es la repetición

Por ejemplo, si T = 50 s. y R = 1 entonces el número de combinaciones que generará será 50.

Tomando en caso anterior, si el T = 500 ms. Y R = 1 entonces se generar 500 combinaciones.

Paso 3

Se debe generar todas las combinaciones posibles a partir de cada nivel descrito anteriormente.

Primer nivel

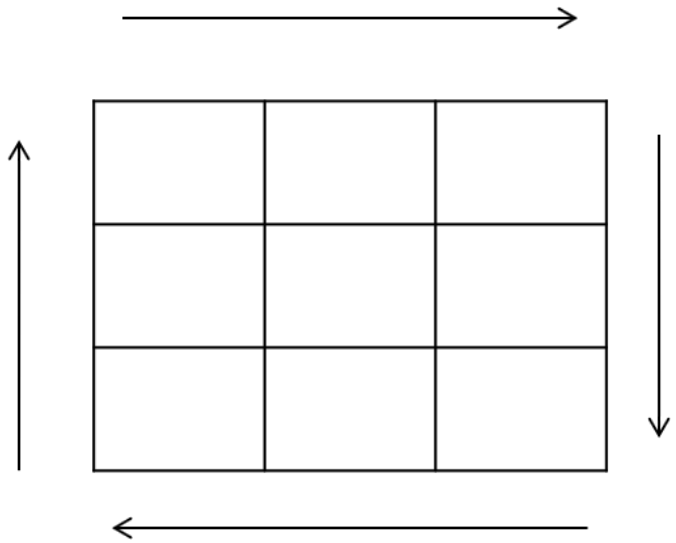
En el primer nivel se definirán los movimientos de distintas formas, inicialmente se llevará a una matriz que realizará diferentes operaciones.

Sub nivel 1

En este sub nivel se definirá los datos en una matriz y se realizaran diferentes movimientos alrededor de la matriz.

Figura 1.

Primer subnivel de movimientos, alrededor de la matriz,



Elaboración Propia.

Se observa en la Figura 1, que los movimientos son de rotación, por ejemplo, podemos partir de una cadena inicial C0 con valor “abcdefghi”, esta iterara con los siguientes movimientos que se reflejaran en C1:

Movimiento de C0 a la derecha

C1 = “iabcdefgh”

Movimiento de C0 a la izquierda

C1 = “bcdefghia”

Movimiento de C0 a la arriba

C1 = “defghibca”

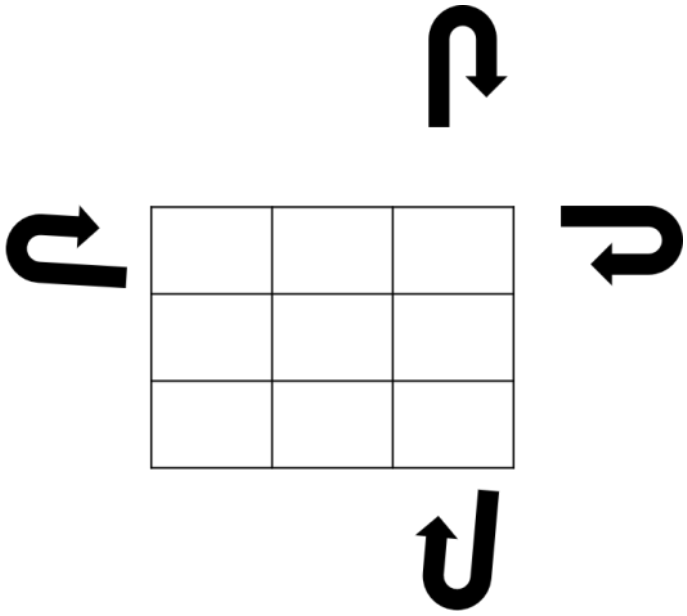
Movimiento de C0 a la abajo

C1 = “ighabcdef”

Sub nivel 2

Figura 2.

Segundo nivel, de movimientos,



Elaboración Propia.

Cada subnivel es distinto, en este subnivel se realizarán operaciones en todo sentido, pero respetando los movimientos descritos en la Figura 1.

Por ejemplo, tomando en cuenta la siguiente cadena inicial denominada C0 con valor “abcdefghi” y los cambios se reflejan en C1:

Movimiento de C0 a la derecha en la fila 0

C1 = “cabdefghi”

Movimiento de C0 a la izquierda en la fila 1

C1 = “abcefdghi”

Movimiento de C0 a la arriba en la columna 1

C1 = “aecdhfgbi”

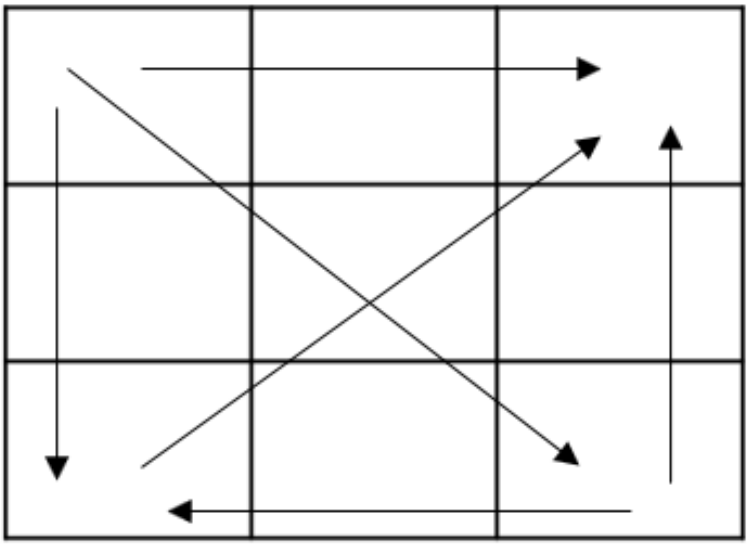
Movimiento de C0 a la abajo en la columna 2

C1 = “abidecghf”

Sub nivel 3

Figura 3.

Primer subnivel de movimientos, intercambios de valores de la matriz



Elaboración Propia.

Se observa en la Figura 3 un intercambio de los valores de la matriz, pueden ser en distintas direcciones, por ejemplo, podemos partir de una cadena inicial C0 con valor "abcdefghi", esta iterara con los siguientes movimientos

Intercambiando en C0 la primera columna:

C1 = "gbcdefahi"

Intercambiando en C0 la primera fila:

C1 = "cbadefghi"

Intercambiando en C0 de forma diagonal:

C1 = "ibcdefghi"

Intercambiando en C0 de forma doble diagonal:

C1 = "ibgdefcha"

Intercambiando en C0 de forma de cruz:

C1 = "ahcfedgbi"

Segundo Nivel

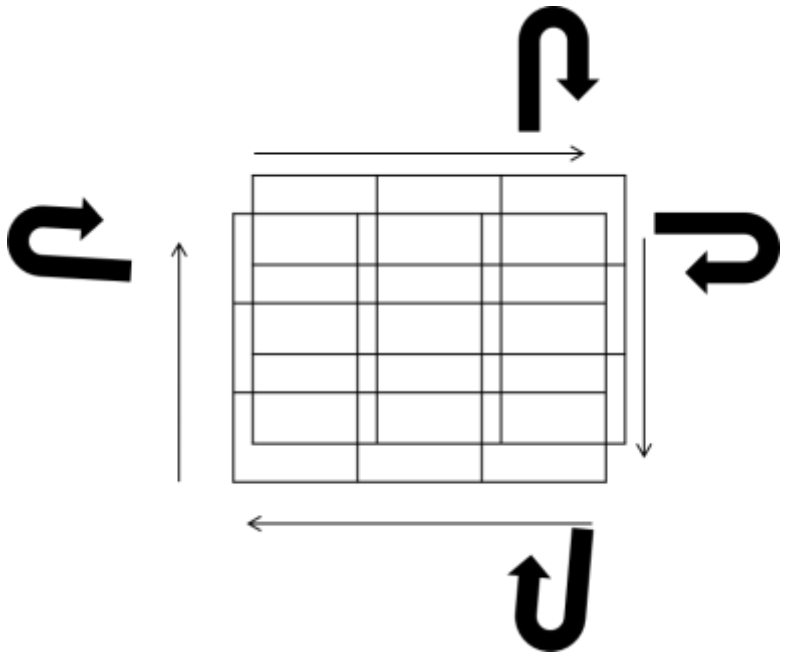
En este nivel se realizará operaciones matemáticas para el tratamiento de matrices, como por ejemplo la transposición, recorrer la matriz de forma diagonal, invertir entre otras operaciones algebraicas.

Tercer Nivel

Este nivel es teórico, se deberá realizar movimientos en 3 dimensiones, además se podría utilizar los movimientos del primer y segundo nivel.

Figura 4.

Tercer nivel, de movimientos, en tercera dimensión



Elaboración Propia.

Generación de combinaciones

Aplicando todos los subniveles se podrá generar todas las combinaciones a partir de una contraseña, utilizando la siguiente formula podemos obtener los siguientes datos:

$$N_c = \frac{T}{R}$$

- Cadena Original ->abcdef123456
- T = 50(s)
- M= 50
- Ejecución = 1 (s)
- Código Generado

- DG;BG;AG;AF;DE;BE;CC;BF;BF;AF;CC;CE;AA;CF;DF;DF;CE;AE;DF;DD;BB;BB;DF;DD;BE;DE;CF;DE;DG;CF;CG;CC;CF;DG;BF;DD;BG;DE;CE;DG;AE;CF;DG;AA;AA;AG;BB;AG;CG;AF

Código generado

- 4--->*b*daefc24163*5*
- 5--->b*d*aefc24163*5*
- ……..
- 48--->3b2d**ec1f46**5a
- 49--->3b2dc**e1f46**5a
- Cifrado>3b2dc**e1f46**5ª

Existe la posibilidad que los valores que se generen se repitan, por ejemplo:

- 45--->*b*daefc24163*5*
- 59--->b*d*aefc24163*5*
- ……..
- 482---> b*d*aefc24163*5*
- ……..
- 569--->3b2dc**e1f46**5a
- 999--->**e3b2dc1f**5a46

En una generación de 1000 combinaciones se generó un duplicado no continuado, en la primera aparición está en el número 5 y la siguiente en el número 482.

2.2. Tiempo de descifrado

Para determinar el tiempo de descifrado estimado se trabajará con la estructura del ataque de fuerza bruta.

Figura 5. *Combinaciones para una contraseña*. Elaboración Propia.

$$\boldsymbol{c = a^l}$$

Donde:

c - Combinaciones posibles

a - Alfabeto utilizado

l - Longitud de la contraseña

En la Figura 5 se puede observar el alfabeto (Cantidad de letras mayúsculas, minúsculas, números y símbolos), longitud de la contraseña. Las operaciones por segundo de la

maquina donde se realice el ataque de fuerza bruta dependerá de las características técnicas de la misma, debe probar todas las combinaciones posibles para obtener la contraseña.

Figura 6. *Tiempo de procesamiento de todas las combinaciones para una contraseña. Elaboración Propia*

$$\boldsymbol{b = \frac{a^l}{p}}$$

Donde:

b - Tiempo de procesamiento parcial

p - Procesamiento por segundo

a - Alfabeto utilizado

l - Longitud de la contraseña

El modelo utiliza múltiples contraseñas, para ser vulnerable se deberá utilizar todas las contraseñas que se utilizaron para cifrar.

Figura 7. *Tiempo en segundos para vulnerar el modelo, usando un ataque de fuerza bruta. E*laboración Propia.

$$F_t = \frac{a^l}{p} + \frac{a^l}{p} + \cdots + \frac{a^l}{p}$$

$$F_t = \sum_{i=1}^{n} \frac{a^l}{p}$$

Donde:

F_t - Tiempo Final de descifrado

p - Procesamiento por segundo

a - Alfabeto utilizado

l - Longitud de la contraseña

n - Numero de contraseñas generadas

Para que el modelo sea eficiente el valor $F_t > T_{op}$, Donde T_{op} es el tiempo de descifrado de la información.

2.3. Probabilidad de vulneración mediante ataque de fuerza bruta

Para obtener la probabilidad de obtener todas las contraseñas utilizadas, tomando en cuenta la Figura 5, se puede obtener la probabilidad de cada intento.

Figura 8. *Probabilidad de obtener una contraseña. Elaboración Propia*.

$$P = \frac{1}{a^l}$$

Donde:

P - Probabilidad

a - Alfabeto utilizado

l - Longitud de la contraseña

El Modelo utiliza varias contraseñas con carácter finito, se debe aplicar el concepto de probabilidad conjunta, para averiguar la probabilidad de acertar todas las contraseñas.

Figura 9. *Probabilidad conjunta para descubrir todas las contraseñas*

$$P(c_1) = \frac{1}{a^l}$$

$$P(c_2) = \frac{1}{a^l}$$

…

$$P(c_n) = \frac{1}{a^l}$$

Donde:

P - Probabilidad

c_i – Contraseña de un intento i

a - Alfabeto utilizado

l - Longitud de la contraseña

Tomando en cuenta la Figura 9, se puede obtener la probabilidad conjunta para descifrar todas las contraseñas.

Figura 10. *Probabilidad conjunta. Elaboración Propia*

$$P(c_1 \cup c_2 \cup \ldots \cup c_n) = \frac{1}{a^l} \times \frac{1}{a^l} \times \ldots \times \frac{1}{a^l}$$

$$P(c_1 \cup c_2 \cup \ldots \cup c_n) = \prod_{i=1}^{n} \frac{1}{a^l}$$

Donde:

P - Probabilidad

c_i – Contraseña en la iteración i

a - Alfabeto utilizado

l - Longitud de la contraseña

3. RESULTADOS Y DISCUSIÓN

Para validar el método de cifrado de punto a punto basado en la obsolescencia programada se utilizará las ecuaciones de la Figura 7 y Figura 10. Con los siguientes valores:

Figura 11. *Tiempo de descifrado utilizando la Figura 7. Elaboración Propia*.

Tiempo de Descifrado (T_{op}). – Tiempo de descifrado del archivo

Valores: 60 segundos

Alfabeto (a). – Son todos los caracteres que se utilizaran para poder descifrar la contraseña empleada.

Valores: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890/*-+°|!”#$%&/()=?¡[]{}-.,;:

Longitud (l). – Longitud de la contraseña utilizada.

Valores: 10

Procesamiento (p). – Procesamiento de una computadora estándar de 298,000,000,00 cálculos por segundo.

Valores: 298,000,000,000

$$F_t = \frac{93^{10}}{298{,}000{,}000{,}000} + \frac{93^{10}}{298{,}000{,}000{,}000} + \cdots + \frac{93^{10}}{298{,}000{,}000{,}000}$$

$$F_t = 9744610212\ s$$

Según el resultado de la Figura 11, en un tiempo de 60 segundos, utilizando una contraseña de 10 caracteres, utilizando 60 contraseñas, se descifraría la información en un total de 9744610212 segundos, que equivale a 308.79 Años, el valor obtenido es mayor a 60 segundos, por lo que demuestra que el modelo presenta resistencia a un ataque de fuerza bruta.

Figura 12. *Probabilidad de obtener todas las contraseñas, utilizando la Figura 10. Elaboración Propia*

$$P(c_1 \cup c_2 \cup \ldots \cup c_{60}) = \frac{1}{93^{10}} \times \frac{1}{93^{10}} \times \ldots \times \frac{1}{93^{10}}$$

$$P(c_1 \cup c_2 \cup \ldots \cup c_{60}) = 1{,}23971E - 18$$

Según el resultado de la Figura 12, la probabilidad de obtener las 60 contraseñas empleadas en cifrar la información es de 1,23971E-18.

En el presente trabajo se desarrolló el estudio inicial sobre el modelo de cifrado de punto a punto basado en la obsolescencia programada, el cual permite cifrar y descifrar con múltiples contraseñas, a partir de una contraseña, esto permite proteger la información por un tiempo determinado.

El Modelo es robusto, cuando se utiliza una contraseña por segundo y el procesador de una computadora que utiliza 298 mil millones de cálculos por segundo, dando como resultado que la una computadora tardara en descifrar todas las contraseñas en un tiempo de 308.79 años, en el caso que se utilice otro computador con mayor potencia se puede reducir el tiempo de descifrado.

El ataque de fuerza bruta trabaja sobre un alfabeto que contiene letras mayúscula, minúscula, símbolos, números y caracteres especiales, para el presente trabajo se utilizó un alfabeto de 93 caracteres como se describe anteriormente, el alfabeto no es completo, debido a que faltan caracteres especiales, que se encuentran en el código ASCII, espacios, arroba, exponente, acento entre otros, un ataque de fuerza bruta para que sea efectivo deberá utilizar todos los símbolos, números, alfabeto y caracteres especiales.

El modelo de cifrado de punto a punto basado en obsolescencia programada es compatible con diferentes algoritmos de cifrado, su fortaleza radica en el numero de contraseñas empleadas para descifrar la información.

El modelo tiene una característica de tener la propiedad de ser de carácter finito, tiene la característica principal de tener una fecha de caducidad, por el cual la información puede ser liberada antes que el atacante pueda obtener todas las contraseñas utilizadas.

Siempre se recomienda emplear las contraseñas seguras, recomendaciones que realizan las empresas de Antivirus, debido a que una contraseña con números y símbolos tiende a emplear más recursos del procesador, una contraseña insegura es vulnerable a un ataque de fuerza bruta, ataque de diccionario u otro ataque similar, descifrar una contraseña insegura puede consumir menos recursos del computador.

4. CONCLUSIÓN O CONSIDERACIONES FINALES

El modelo de cifrado de punto a punto basado en obsolescencia programada, demostró ser resistente bajo un ataque de fuerza bruta, de manera teórica, la generación de varias contraseñas a partir de una contraseña principal permite reforzar la seguridad del modelo, cifrando recursivamente la información por un tiempo limitado.

Se debe emplear contraseñas seguras de acuerdo a las recomendaciones de las empresas de Antivirus.

El modelo de cifrado es compatible con los algoritmos de cifrado que utilizan una contraseña para cifrar o descifrar la información.

Para generar cada combinación se utiliza métodos de rotación, sustitución, operaciones algebraicas, haciendo que cada combinación sea diferente de la anterior o la siguiente.

El modelo de cifrado de punto a punto basado en obsolescencia programada puede ser utilizado en procesos de licitación, convocatorias públicas, series o películas de televisión o de carácter de estreno que evitara que se filtre la información antes de su publicación ya planificada.

5. LISTA DE REFERENCIAS

Albors, J. (24 de Junio de 2020). WeLiveSecurity. Recuperado el 1 de Mayo de 2022, de WeLiveSecurity By Eset: https://www.welivesecurity.com/la-es/2020/06/24/que-es-ataque-fuerza-bruta-como-funciona/

Andrade, A. (7 de Enero de 2019). Características y aplicaciones de las funciones resumen criptográficas en la gestión de contraseñas. *Características y aplicaciones de las funciones resumen criptográficas en la gestión de contraseñas*. Alicante, España. Recuperado el 2 de Mayo de 2022, de http://hdl.handle.net/10045/96849

Aravena, M. (2018). Selección e Implementación de Librería Java para E-Voting. *Selección e Implementación de Librería Java para E-Voting*. Concepcion, BioBio,

Chile. Recuperado el 6 de Mayo de 2022, de http://repobib.ubiobio.cl/jspui/handle/123456789/2717

Barrientos, M. (2021). Mecanismo de bloqueo a conexiones remotas que intentan accesos por fuerza bruta para Nova-LTSP. *UCIENCIA 2021*. Recuperado el 6 de Mayo de 2022, de https://repositorio.uci.cu/jspui/handle/123456789/9636

Bazzi, B. (2021). Análisis de la seguridad en KeePassX. *Análisis de la seguridad en KeePassX*. Madrid, España. Recuperado el 7 de Mayo de 2022, de https://oa.upm.es/67990/

Elbinario.net. (11 de Abril de 2016). *Elbinario.net*. Recuperado el 15 de Enero de 2019, de Elbinario.net: https://elbinario.net/2016/04/11/algoritmos-de-cifrado-ii/

Elbinario10.net. (05 de Abril de 2016). *Elbinario10.net*. Recuperado el 20 de Enero de 2019, de Elbinario10.net: https://elbinario.net/2016/04/05/algoritmos-de-cifrado-i/

Gutierrez, P. (15 de Agosto de 2017). *Genbeta:dev*. Recuperado el 15 de Enero de 2019, de Genbeta:dev: https://www.genbeta.com/desarrollo/tipos-de-criptografia-simetrica-asimetrica-e-hibrida

IBM. (1 de Diciembre de 2009). *IBM Knowledge Center*. Recuperado el 15 de Enero de 2019, de IBM Knowledge Center: https://www.ibm.com/support/knowledgecenter/es/SSGU8G_11.50.0/com.ibm.sec.doc/ids_en_010.htm

Infante, M. (2013). Criptografía y psicología de la contraseña: generando una contraseña fuerte para diferentes servicios. *Apuntes de Ciencia & Sociedad, 3*, 74. Recuperado el 15 de Febrero de 2019, de https://dialnet.unirioja.es/servlet/articulo?codigo=5042962

Llante, Y., Marceles, K., & Amador, S. (2022). Tecnología Blockchain adaptable para un framework de ciberseguridad en IIoT. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*. doi:10.1109/RITA.2022.3166857

Lopez, V. (2013). Papel de la Explosión Combinacional en Ataques de Fuerza Bruta. *Investigación e Innovación en Ingenierías, 1*(1), 28-32. Recuperado el 20 de Febrero de 2019, de http://revistas.unisimon.edu.co/index.php/innovacioning/article/view/2069

Mendoza, E. (2021). Aplicación de Fuerza Bruta con Diccionario de Datos, para vulnerar contraseñas débiles en archivos comprimidos Zip. *Revista PGI Investigación, Ciencia y Tecnologia en Informatica*, 23-25. Recuperado el 3 de Mayo de 2022, de https://ojs.umsa.bo/ojs/index.php/inf_fcpn_pgi/article/view/99

Mogollon, O., Gonzales, M., Sancho, J., & Jimenez, J. (2021). Deteccion de anomalías en ataques de contenedores de software mediante la monitorizacion y el analisis de recursos del sistema. *RECSI*, 119-124. Recuperado el 12 de Mayo de 2022, de http://www.recsi2020.udl.cat/static/site/files/MogollonGonzalezSanchoJimenez-XVI-RECSI.pdf

Pandasecurity. (23 de Febrero de 2016). *PandaSecurity*. Recuperado el 15 de Marzo de 2019, de PandaSecurity: https://www.pandasecurity.com/spain/mediacenter/seguridad/10-trucos-para-crear-contrasenas-seguras/

Perdigon, R. (2022). Suricata como detector de intrusos para la seguridad en redes de datos empresariales. *Ciencia UNEMI*, 44-53. doi:https://doi.org/10.29076/issn.2528-7737vol15iss39.2022pp44-53p

Pousa, A., Sanz, V., Naiouf, M., & Giusti, A. (2018). Rendimiento del algoritmo AES sobre arquitecturas de memoria compartida. *Rendimiento del algoritmo AES sobre arquitecturas de memoria compartida* (págs. 73-82). La Plata: Repositorio Institucional de la UNLP. Recuperado el 5 de Mayo de 2022, de http://sedici.unlp.edu.ar/handle/10915/73037

Romero, M., & Leon, M. (4 de Agosto de 2018). Criptografía, Cifrando Sistemas Inteligentes. *Revista HashTag*. doi:https://doi.org/10.52143/2346139X.612

UPM. (3 de Mayo de 2017). *Escuela Tecnica Superior de Ingenieria de Sistemas Informaticos*. Recuperado el 5 de Enero de 2019, de https://www.etsisi.upm.es/google-anuncia-primer-ataque-colision-sha-1

Vargas, M., & Yuri, T. (Junio de 2015). *Dialnet*. Recuperado el 12 de Enero de 2019, de Dialnet: https://dialnet.unirioja.es/servlet/articulo?codigo=5286657

Velasco, R. (7 de Noviembre de 2017). *RedesZone.net*. Recuperado el 13 de Enero de 2019, de RedesZone.net: https://www.redeszone.net/2014/11/07/se-generalizan-los-ataques-de-colision-md5-demostrando-la-ineficacia-del-algoritmo/

Zhumaniezov, A. (2018). Investigación y optimización del Sistema de Cifrado ElGamal. *Revista Dilemas Contemporáneos*. Recuperado el 10 de Mayo de 2022, de https://dilemascontemporaneoseducacionpoliticayvalores.com/index.php/dilemas/article/view/716/1069