Estudio del desempeño de algoritmos de criptografía para almacenamiento de archivos encriptados en la nube

*Haciendo uso de una maquina AWS

1st Dilan Esteban Rey Sepulveda
2190397
Escuela de Ingeniería de sistemas.
Bucaramanga, Santander

2nd Sergio Hernando Barón Rivera
2201885
Escuela de Ingeniería de sistemas.
Bucaramanga, Santander

***Abstract*—Este articulo presenta la investigación enfocada en el ámbito de la seguridad informática (criptografía) para servicios de computación en la nube, más específicamente una maquina virtual alojada en la nube de Amazon Web Services (AWS). Se analizará el rendimiento otorgado por un par de algoritmos de criptografía implementados en Python en aspectos tales como tamaño de almacenamiento del archivo encriptado / tiempo de almacenamiento, a su vez se comparará estos desempeños por el otorgado por los mismos algoritmos ejecutados en una maquina local.**

***Index Terms*—AWS, Computación en la nube, criptografía.**

I. Introduction

La computación en la nube se refiere al uso del software de infraestructura en red y la capacidad que proporciona recursos al entorno bajo demanda. La información se almacena en servidores centralizados y se almacena en caché temporalmente en clientes que pueden incluir computadoras de escritorio, portátiles, dispositivos portátiles y otros dispositivos. La complejidad de la nube se puede reducir simplemente reduciéndola a miles de primitivas y unidades funcionales comunes replicadas. Estas complejidades crean muchos problemas relacionados con la seguridad, así como con todos los aspectos de la computación en la nube. La nube suele tener una arquitectura de seguridad única, pero tiene muchos clientes con demandas diferentes. El desafío de los problemas de seguridad surge debido al hecho de que tanto los datos del cliente como el programa residen en las instalaciones del proveedor.

Este articulo desarrolla una evaluación de 2 técnicas/algoritmos de cifrado seleccionadas, a saber, Fernet (de la librería cryptography de phyton)y el cifrado Cesar. La evaluación del desempeño de estos algoritmos se realiza midiendo el tamaño resultante del archivo después de encriptar/desencriptar y a su vez el tiempo que toma cada algoritmo para realizar estos procesos, esto en ambos ambientes de ejecución (nube y local).

El objetivo principal de este articulo es investigar y estudiar distintos tipos de algoritmos que se pueden implementar en una nube de Amazon Web Services (AWS) para un microservicio de almacenamiento de documentos confidenciales.

II. Computación en la nube

Esta sección ofrece una descripción general de la computación en la nube y la seguridad de los datos en la computación en la nube. Existen muchas definiciones que intentan abordar la nube desde la perspectiva de académicos, arquitectos, ingenieros, desarrolladores, gerentes y consumidores. La definición más simple de computación en la nube es “mover la computación desde una sola PC de escritorio/centro de datos a Internet” [1], [2].

A. Aspectos importantes

Los aspectos más importantes del Cloud Computing incluyen:

- Acceso a Demanda.
- Escalabilidad.
- Servicios Medibles y Gestionados.
- Acceso Ubicuo.
- Modelo de Pago por Uso.
- Servicios Compartidos.
- Elasticidad.

B. Seguridad de la computación en la nube

Si bien el costo y la facilidad de uso son dos grandes beneficios de la computación en la nube, existen importantes preocupaciones de seguridad que deben abordarse al considerar trasladar aplicaciones críticas y datos confidenciales a entornos de nube públicos y compartidos. Para abordar estas preocupaciones, el proveedor de la nube debe desarrollar controles suficientes para proporcionar el mismo o mayor nivel de seguridad que el que tendría la organización si no se utilizara la nube.

III. Metodología

A. Proceder

Inicialmente se realiza el registro en Amazon Web Services, se lanza una instancia EC2 t2.micro (capa gratuita AWS) con un SO Ubuntu Server 22.04. La conexión remota a la instancia se realiza mediante SSH usando el programa MobaXTerm. Se codifican los distintos algoritmos en programas .py, cada uno para su respectivo proceso (Encriptar o Desencriptar).

Se corren los distintos programas sobre cada uno de los archivos prueba (5 veces cada uno) y se hacen las mediciones de las métricas a tomar en cuenta, para su posterior análisis.

B. Algoritmos

- Fernet (Libreria 'cryptography' de Python): Es una implementación de criptografía autenticada simétrica (también conocida como "clave secreta") [3]. A grandes rasgos funciona de la siguiente manera:
  - **Generación de clave:** Primero, se genera una clave secreta aleatoria que se utilizará tanto para cifrar como para descifrar los datos.
  - **Creación del objeto Fernet:** Se crea un objeto Fernet utilizando la clave generada. Este objeto se utiliza para realizar operaciones de cifrado y descifrado.
  - **Cifrado de datos:** Para cifrar datos, se utiliza el objeto Fernet para aplicar el algoritmo AES en modo CBC. Los datos originales se dividen en bloques y se realiza la operación de cifrado en cada bloque. Además, se añade un código de autenticación para garantizar la integridad de los datos.
  - **Descifrado de datos:** Para descifrar datos, se utiliza el mismo objeto Fernet y la clave correspondiente. Se realiza la operación inversa al cifrado para obtener los datos originales.
- El método César: Es un algoritmo de cifrado muy simple que se basa en desplazar cada letra de un texto una cantidad fija de posiciones en el alfabeto. Es un tipo de cifrado por sustitución, donde cada letra del texto original es reemplazada por otra letra que se encuentra un número fijo de posiciones hacia adelante o hacia atrás en el alfabeto. A grandes rasgos funciona de la siguiente manera:
  - **Seleccionar un desplazamiento (clave):** Se elige un número entero que determina cuántas posiciones se desplazarán las letras. Este número se conoce como la "clave".
  - **Recorrer el texto original:** Cada letra del texto original se desplaza hacia adelante en el alfabeto por la cantidad de posiciones determinada por la clave. Si la letra se encuentra al final del alfabeto, el desplazamiento se realiza de manera circular (vuelve al principio).
  - **Texto cifrado:** El resultado es el texto cifrado, donde cada letra ha sido reemplazada por la letra desplazada.

C. Recursos computacionales

- Instancia AWS EC2 t2.micro con un vCPU Intel Xeon Scalable de hasta 3.3 GHz, 1 GiB Memoria. Sistema operativo instalado: Linux Ubuntu Server 22.04.
- Local: Intel core i5-8300H 2.30GHz, 4 procesadores físicos, 8 lógicos, 16 GB Memoria. Sistema operativo instalado: Windows 10.

D. Archivos utilizados

- lorem.txt (Tamaño: 6kb).
- titanic.csv (Tamaño: 59kb).

IV. Experimentos (Resultados)

A. Cloud

- Pruebas Fernet (Cifrado)
  - **titanic.csv** Tamaño despues de cifrado: 80kb

TABLE I: Tiempos

Numero de prueba	Tiempo
1	0.07496
2	0.05419
3	0.05317
4	0.05591
5	0.05368

  - **lorem.txt** Tamaño despues de cifrado: 8kb

TABLE II: Tiempos

Numero de prueba	Tiempo
1	0.05346
2	0.05298
3	0.05429
4	0.05516
5	0.05339

- Pruebas Cesar
  - **titanic.csv** Tamaño despues de cifrado: 60kb

TABLE III: Tiempos

Numero de prueba	Tiempo
1	0.02146
2	0.02103
3	0.02080
4	0.02060
5	0.02153

  - **lorem.txt** Tamaño despues de cifrado: 5.9kb

TABLE IV: Tiempos

Numero de prueba	Tiempo
1	0.00315
2	0.00331
3	0.00339
4	0.00333
5	0.00333

B. Local

- Pruebas Fernet (Cifrado)
  - **titanic.csv** Tamaño después de cifrado: 80kb

TABLE V: Tiempos

Numero de prueba	Tiempo
1	0.21342
2	0.20744
3	0.19628
4	0.18849
5	0.24035

  - **lorem.txt** Tamaño despues de cifrado: 8kb

TABLE VI: Tiempos

Numero de prueba	Tiempo
1	0.98137
2	0.20744
3	0.19647
4	0.19846
5	0.19248

- Pruebas Cesar
  - **titanic.csv** Tamaño despues de cifrado: 60kb

TABLE VII: Tiempos

Numero de prueba	Tiempo
1	0.13603
2	0.14000
3	0.11999
4	0.12396
5	0.13999

  - **lorem.txt** Tamaño despues de cifrado: 5.9kb

TABLE VIII: Tiempos

Numero de prueba	Tiempo
1	0.04288
2	0.00997
3	0.00997
4	0.00997
5	0.01097

V. Conclusiones

A raíz de los resultados generados se llegó a las siguientes conclusiones:

- No influyo de manera significativa el entorno en el que se usaran los distintos algoritmos de cifrado en cuanto a tamaño de los archivos se refiere, puesto que los dos archivos variaron su tamaño en igual medida y valor en ambos entornos de ejecución (Tanto cloud como local).
- La variación mas notoria se vio reflejada en cuanto a tiempo de cifrado se refiere, esto ya que en el ambiente de la maquina virtual AWS los algoritmos tuvieron un performance notablemente mayor al obtenido en el ambiente local, tomándole menor tiempo a la máquina virtual el correr estos algoritmos. Esto se debe a que estas máquinas están optimizadas para el uso único y especifico de cómputo, mientras que la maquina local, aunque tiene más recursos, se encuentra ejecutando muchos más procesos en paralelo.
- Se evidencia que algoritmo Cesar es notablemente mas veloz que el algoritmo Fernet, sin embargo esto tiene que ver de igual forma con la fortaleza de cifrado de cada uno de los algoritmos (apoyados en el tamaño del archivo encriptado, el cual aumenta en gran medida con el algoritmo Fernet).
- En el caso del algoritmo Fernet, al ser un método mucho más robusto que el César, los archivos que usamos aumentan considerablemente en tamaño cuando se ejecuta el cifrado. Se hicieron varias pruebas con diferentes archivos .csv y se encontró que el encriptador agrega hasta un 33.3% más al tamaño del archivo, y en la computadora local, al momento de la ejecución de un archivo de 190 mb; se agotó la memoria RAM al generar el nuevo archivo encriptado. Por lo tanto, Es esencial tener en cuenta esto al emplear este método en una instancia en la nube, porque podría exceder los límites de los recursos, generando costos extras.

References

[1] Luis M. Vaquero1, Luis Rodero-Merino, Juan Caceres, Maik Lindner2 "A Break in the Clouds: Towards a Cloud Definition", ACM SIGCOMM Computer Communication Review, Vol. 39, No.1, 2009

[2] P. Mell y T. Grance, 'Cloud computing definition," NIST June 2009 http://csrc.nist.gov/groups/SNS/c1oud-computing/index.html

[3] "Fernet (symmetric encryption) — Cryptography 42.0.0.dev1 documentation". Welcome to pyca/cryptography — Cryptography 42.0.0.dev1 documentation. Accedido el 29 de noviembre de 2023. [En línea]. Disponible: https://cryptography.io/en/latest/fernet/cryptography.fernet.Fernet