Andres Felipe Gómez García - 202021189

Andrés Felipe Pereira - 202310782

Caso #3 – Canales Seguros

1. Descripción

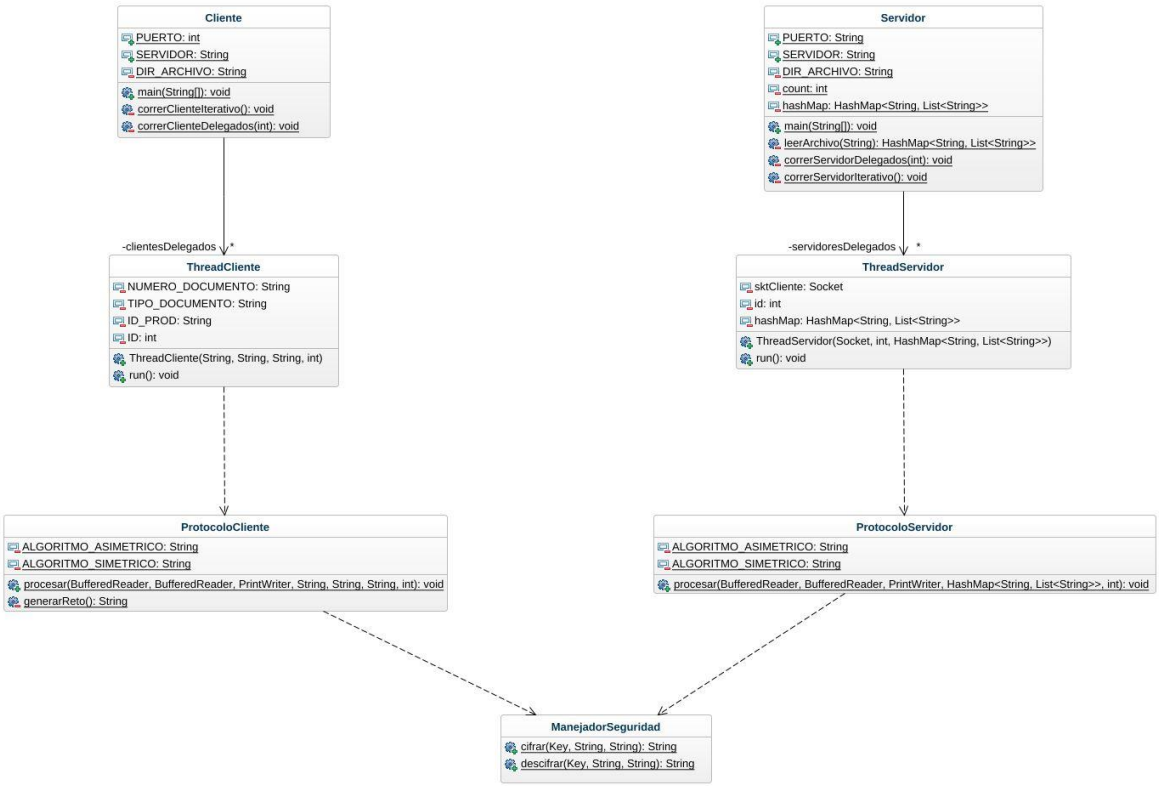

Figura 1. Diagrama UML de clases del programa.

En la figura 1 se observa el diagrama UML del programa se observan 6 clases.

1. **Clase Cliente**:
   - Esta clase contiene el método main que inicia la aplicación cliente. Permite al usuario elegir entre dos modos de operación: modo iterativo o modo con delegados (múltiples clientes concurrentes).
   - **Métodos importantes**:
     - correrClienteIterativo(): Ejecuta el cliente en modo iterativo, leyendo datos de un archivo y procesándolos uno por uno (Asi 32 veces).
     - correrClienteDelegados(int cantidadClientes): Ejecuta múltiples instancias del cliente en paralelo, utilizando la clase ThreadCliente para cada cliente delegado.
2. **Clase ThreadCliente**:
   - Representa un cliente delegado que se ejecuta en un hilo separado.
   - **Constructor**: Inicializa los datos necesarios (número de documento, tipo de documento, ID del producto, y un identificador del cliente).

- **Método run()**: Establece la conexión con el servidor y utiliza el ProtocoloCliente para manejar la comunicación.

3. **Clase ManejadorSeguridad**:
   - Proporciona métodos para cifrar y descifrar texto utilizando algoritmos de criptografía simétrica y asimétrica.
   - **Métodos importantes**:
     - cifrar(Key llave, String algoritmo, String texto): Cifra el texto utilizando la llave y el algoritmo especificados.
     - descifrar(Key llave, String algoritmo, String textoCifrado): Descifra el texto cifrado utilizando la llave y el algoritmo especificados.
4. **Clase ProtocoloCliente**:
   - Implementa el protocolo de comunicación del lado del cliente.
   - **Método procesar()**: Realiza varias operaciones, incluyendo la autenticación del cliente, el intercambio de claves simétricas y asimétricas, y la consulta del estado de un producto. Utiliza la clase ManejadorSeguridad para las operaciones de cifrado y descifrado.
   - **Método generarReto()**: Genera un reto aleatorio que se utiliza para la autenticación del cliente.
5. **Clase ProtocoloServidor**:
   - Implementa el protocolo de comunicación del lado del servidor.
   - **Método procesar()**: Similar al método del cliente, maneja la autenticación del cliente, el intercambio de claves, y proporciona la respuesta a la consulta del estado del producto.
6. **Clase Servidor**:
   - Esta clase contiene el método main que inicia la aplicación servidor. Permite al usuario elegir entre dos modos de operación: modo iterativo o modo con delegados (múltiples servidores concurrentes).
   - **Métodos importantes**:
     - leerArchivo(String csvFile): Lee los datos de un archivo CSV y los almacena en un HashMap.
     - correrServidorIterativo(): Ejecuta el servidor en modo iterativo, procesando las solicitudes de los clientes uno por uno.
     - correrServidorDelegados(int cantidadServidores): Ejecuta múltiples instancias del servidor en paralelo, utilizando la clase ThreadServidor para cada servidor delegado.
7. **Clase ThreadServidor**:
   - Representa un servidor delegado que se ejecuta en un hilo separado.

- **Constructor**: Inicializa el socket del cliente, un identificador del servidor, y un HashMap que contiene los datos.
    - **Método run()**: Maneja la comunicación con el cliente utilizando el ProtocoloServidor.

En el zip está el código fuente, una carpeta data con los archivos de la llave pública y otro con la privada, también hay dos archivos con los datos que usan los clientes y el servidor. Adicionalmente, hay una carpeta docs que tiene los archivos de excel con todo el manejo de datos y el informe.

2. Instrucciones de Ejecución

Para ejecutar el programa, siempre se debe iniciar primero el servidor. El servidor ofrece dos opciones de ejecución: modo iterativo, ingresando "1" cuando se le solicite, y modo delegado, ingresando "2". Si se selecciona el modo delegado, también deberá especificar el número de delegados, que puede ser 4, 16 o 32. Después de iniciar el servidor, debe ejecutar el cliente, el cual presenta las mismas opciones de ejecución que el servidor. En modo iterativo, se ingresa "1", y en modo delegados, se ingresa "2" y se especifica el número de clientes delegados que se desea ejecutar. Es crucial que la información utilizada en el servidor sea idéntica a la utilizada en el cliente. Si no se mantiene esta consistencia, pueden surgir fallos tanto en la interrupción de la comunicación como en los resultados obtenidos. Recuerde seguir estos pasos de manera ordenada para asegurar el correcto funcionamiento del sistema y evitar posibles errores. Después de finalizada la ejecución, en la consola del Cliente se mostrará cómo han sido atendidos los clientes, y en la consola del Servidor se mostrará cuánto ha tardado la codificación del reto para los dos tipos de cifrado.

3. Descripción de Esquema de Llaves Asimétricas

```
KeyPairGenerator generator = KeyPairGenerator.getInstance("RSA");
generator.initialize(1024);
KeyPair keyPair = generator.generateKeyPair();
PublicKey pubKey = keyPair.getPublic();
PrivateKey privKey = keyPair.getPrivate();

FileOutputStream archivoPublica = new FileOutputStream("publica");
ObjectOutputStream oosPublica = new ObjectOutputStream(archivoPublica);

oosPublica.writeObject(pubKey);

FileOutputStream archivoPrivada = new FileOutputStream("privada");
ObjectOutputStream oosPrivada = new ObjectOutputStream(archivoPrivada);

oosPrivada.writeObject(privKey);

oosPublica.close();
oosPrivada.close();
```

Figura 2. Código de generación y guardado de llaves asimétricas.

Para la generación de las llaves del algoritmo asimétrico se realizó un proyecto temporal dado que solo es necesario generarlas una vez. En particular, como se muestra en la figura 2, en un método main se instanció un KeyPairGenerator especificando que el algoritmo a

usar era “RSA”. Luego, se llamó el método initialize(1024) para obtener las llaves de 1024 bits y se extrajeron utilizando los métodos getPublic() y getPrivate() de KeyPair, guradando los resultados en las variables PublicKey pubKey y PrivateKey privKey respectivamente. Finalmente, se utilizaron dos instancias de FileOutputStream y ObjectOutputStream para guardar las llaves en sus archivos correspondientes llamados “publica” y “privada”.

4. Resultados y Análisis

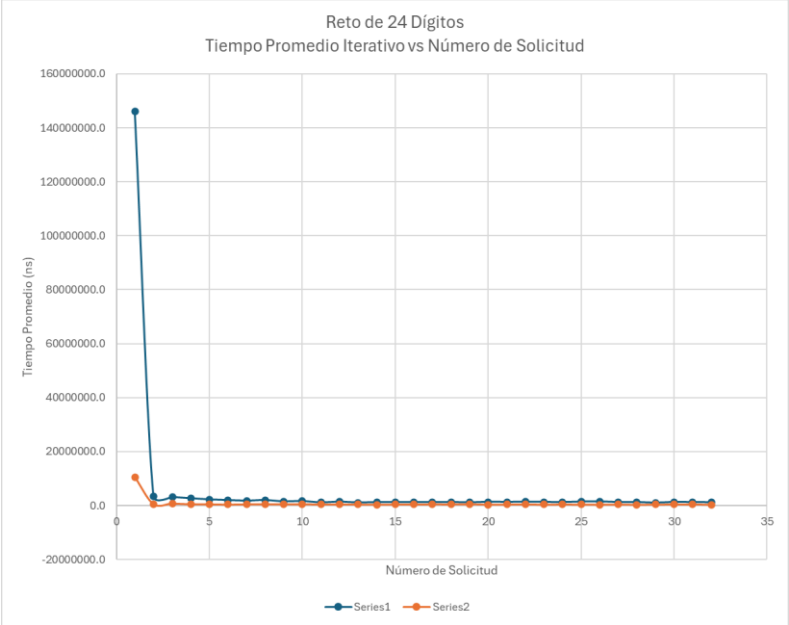

Figura 3. Gráfica ilustrando el carácter anómalo de la primera solicitud del modo iterativo para el reto de 24 dígitos.

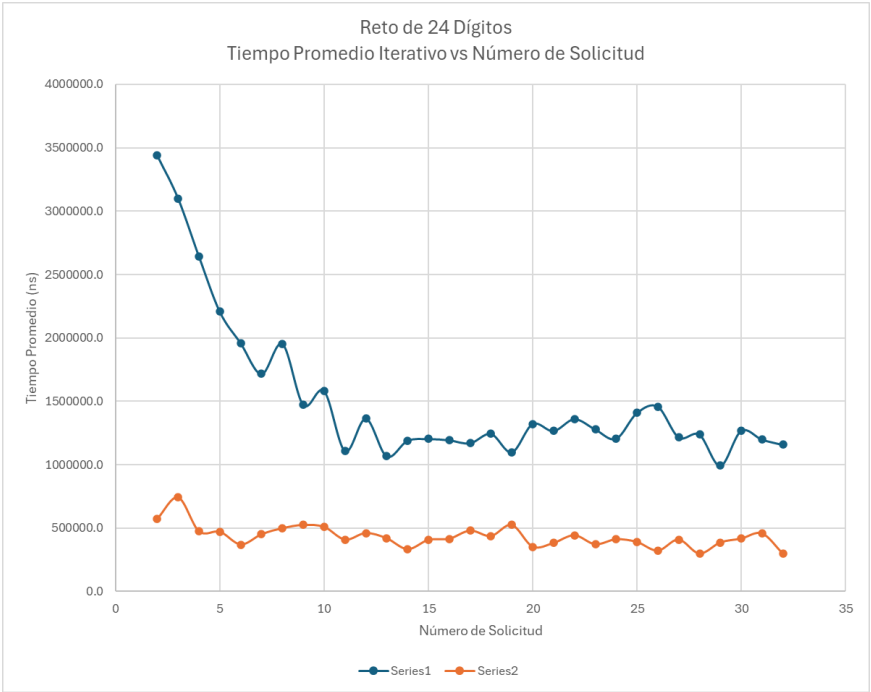

Figura 4. Gráfica ilustrando el carácter anómalo de las solicitudes 2 a 5 del modo iterativo para el reto de 24 dígitos.

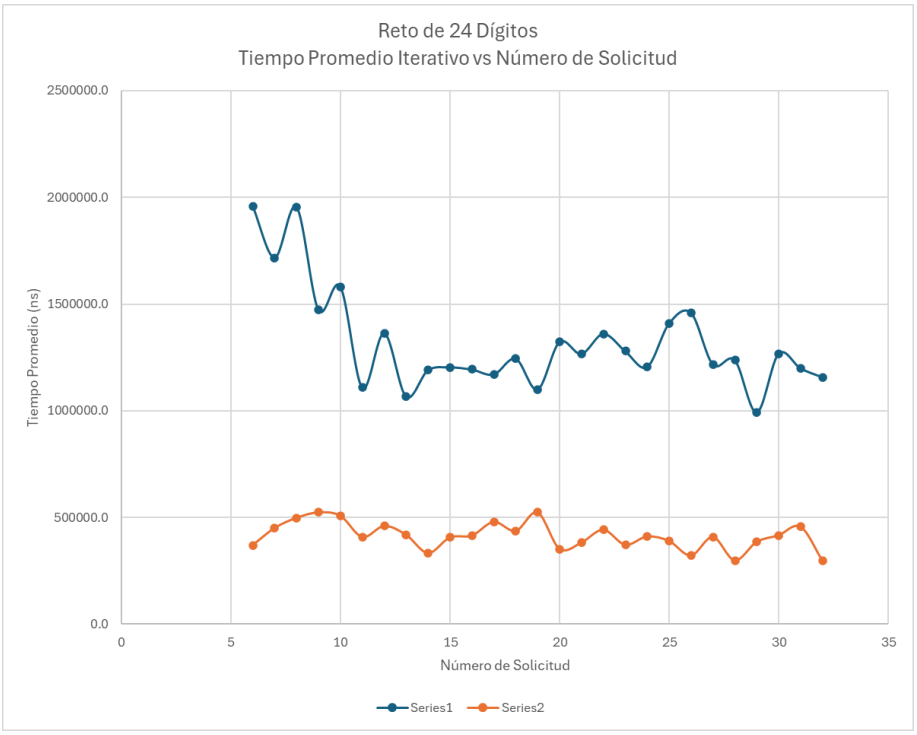

Figura 5. Gráfica ilustrando que de la solicitud 6 en adelante del modo iterativo se estabilizan los tiempos de cifrado para el reto de 24 dígitos.

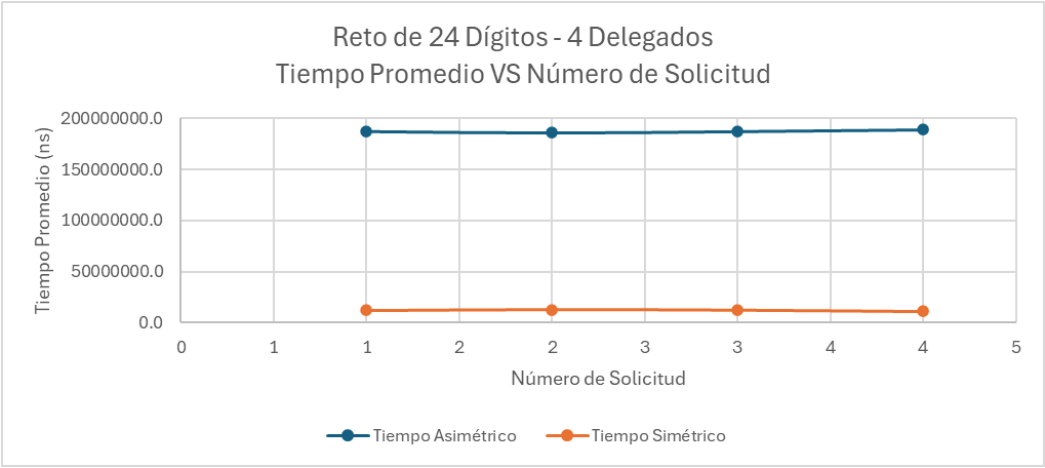

Figura 6. Gráfica de los tiempos promedio de las solicitudes del modo de 4 delegados para el reto de 24 dígitos.

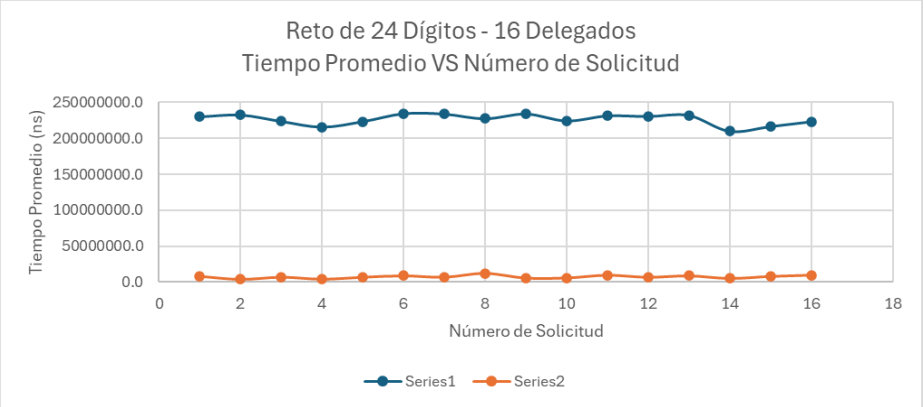

Figura 7. Gráfica de los tiempos promedio de las solicitudes del modo de 16 delegados para el reto de 24 dígitos.

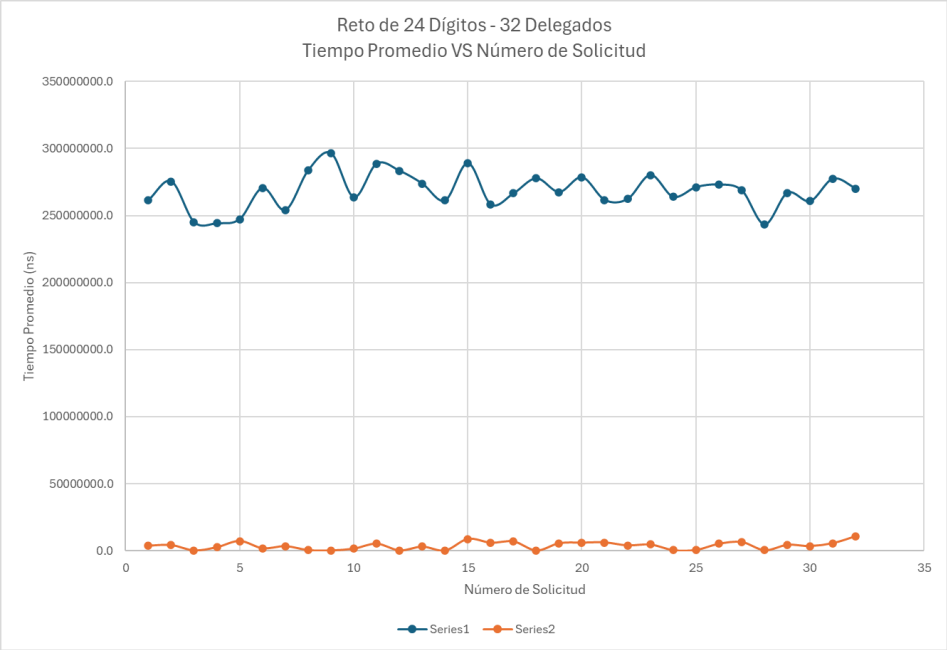

Figura 8. Gráfica de los tiempos promedio de las solicitudes del modo de 32 delegados para el reto de 24 dígitos.

RESUMEN RETO 24 DIGITOS			
/	ASIMÉTRICO	SIMÉTRICO	RELACIÓN
Iterativo	1322033,3	413203,7	3,2
4 Threads	187229775,0	12285015,0	15,2
16 Threads	226301717,5	7330236,3	30,9
32 Threads	268303381,9	3748759,4	71,6

Tabla 1. Tabla mostrando el tiempo promedio de cifrado para el reto de 24 dígitos en los casos asimétrico y simétrico de cada modo de ejecución.

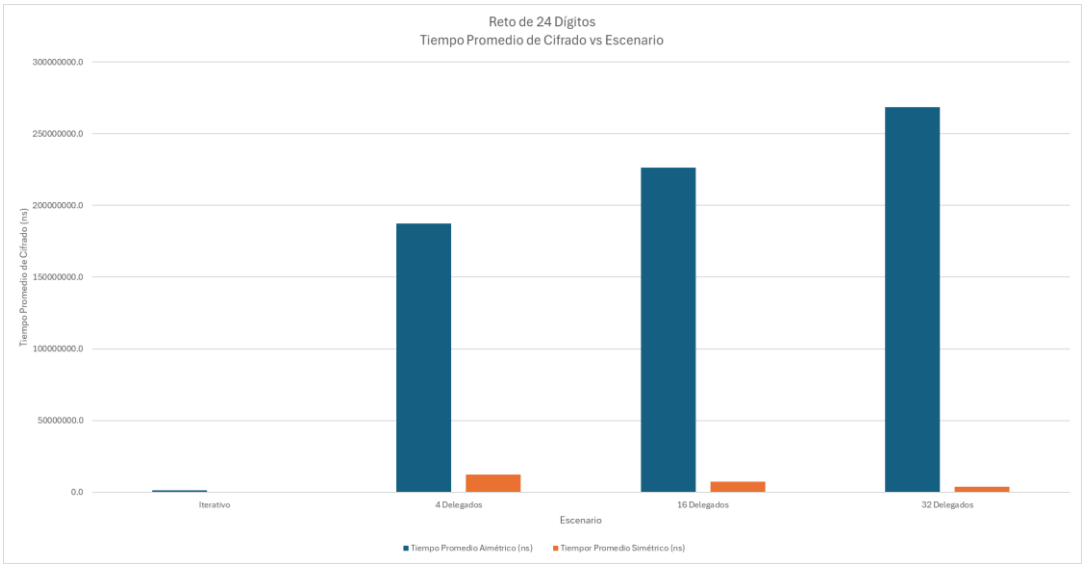

Figura 9. Gráfica de los tiempos promedio de cifrado asimétrico y simétrico por escenario de ejecución para el reto de 24 dígitos.

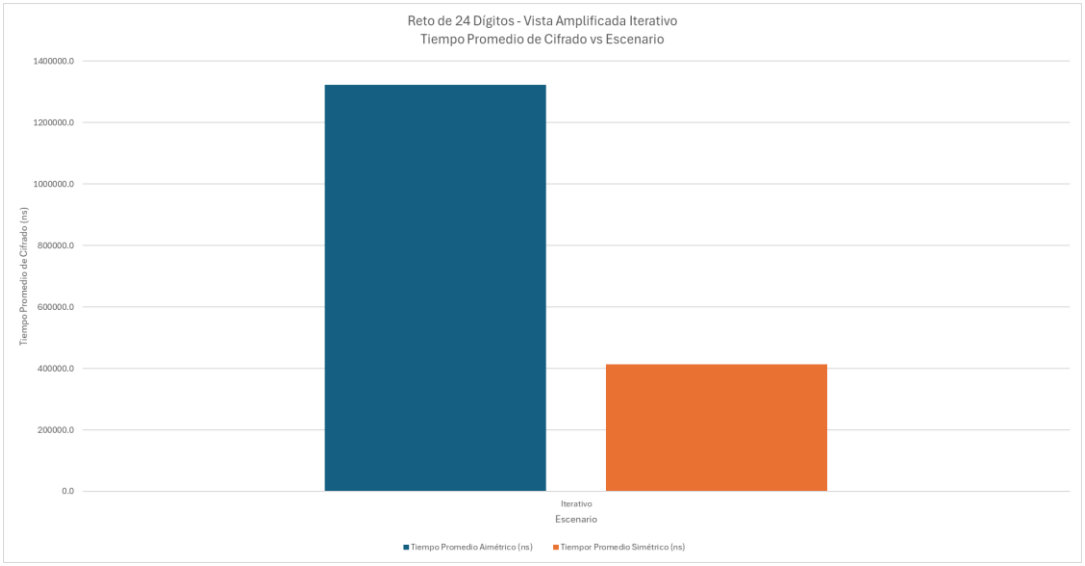

Figura 10. Gráfica que muestra la vista amplificada de los tiempos promedio de cifrado asimétrico y simétrico del escenario de ejecución iterativo para el reto de 24 dígitos.

Inicialmente, como se observa en la figura 3, se resalta que en el escenario iterativo para el reto de 24 dígitos se encontró que el tiempo promedio de la primera solicitud asimétrica era un dato anómalo debido a que su valor era de un orden de magnitud más grande que el segundo tiempo mayor. Asimismo, tras remover el primer dato se encontró, como se muestra en la figura 4, que los tiempos de las solicitudes asimétricas 2 a 5 eran de igual manera muy diferentes a los datos de las solicitudes 6 en adelante, llegando a estar a más 200,000 nanosegundos por encima del tiempo de la sexta solicitud. Consecuentemente, dado que dichos 5 datos sesgarían el promedio y harían la medida menos representativa del caso usual, se procedió a remover los datos anómalos y se tomó, para el caso iterativo, el promedio de las solicitudes 6 en adelante. Note que, conceptualmente, el comportamiento anómalo de estos datos se debe a que la librería de cifrado optimiza por debajo el proceso

de cifrado asimétrico cuando se va a utilizar varias veces la llave, lo que la librería implementa guardando la instancia de la llave después del primer uso.

Por otra parte, se resalta que las figuras 6, 7 y 8 muestran que los tiempos promedio por solicitud para los escenarios de ejecución con 4, 16 y 32 delegados respectivamente se mantienen estables sin datos anómalos de discrepancia significativa como en el caso anterior. Por lo tanto, para dichos escenarios se tomó el promedio de todas las solicitudes sin descartar datos.

Luego, partiendo entonces de los promedios mencionados, se realizó la tabla con los tiempos promedio de cifrado asimétrico y simétrico para cada tipo de escenario con el reto de 24 dígitos, los cuales se plasmaron en la tabla 1. Posteriormente, a partir de dichos datos se realizaron las gráficas de las figuras 9 y 10 en las cuales se comparan los tiempos de cifrado asimétrico y simétrico de cada escenario. Como se observa en ambas figuras, en cada caso se encontró, como se esperaba de acuerdo con la teoría, que el cifrado asimétrico fue significativamente más demorado que el cifrado simétrico, lo cual tiene sentido debido a que el cifrado simétrico se realiza por bloques. Adicionalmente, se encontró que los tiempos de cifrado fueron mucho más bajos en el escenario iterativo, lo cual es de esperarse debido a que en ese escenario el procesador solo está dedicando sus recursos a ejecutar una solicitud, mientras que en los escenarios de delegados debe repartirlos en varias solicitudes concurrentes. Finalmente, se observa que en la figura 9 se evidencia que el tiempo de cifrado asimétrico en el caso de delgados aumentó con el número de delegados mientras que el de cifrado simétrico disminuyó.

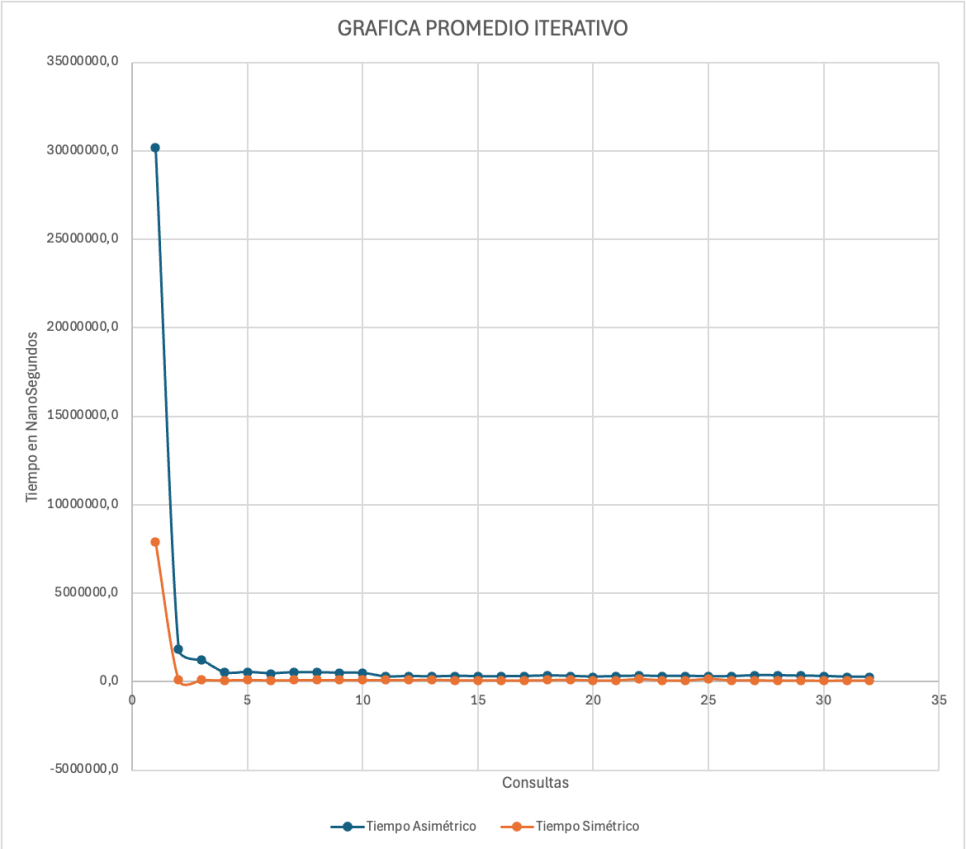

Figura 11. Gráfica ilustrando el carácter anómalo de las primeras 3 solicitudes del modo iterativo para el reto de 32 dígitos.

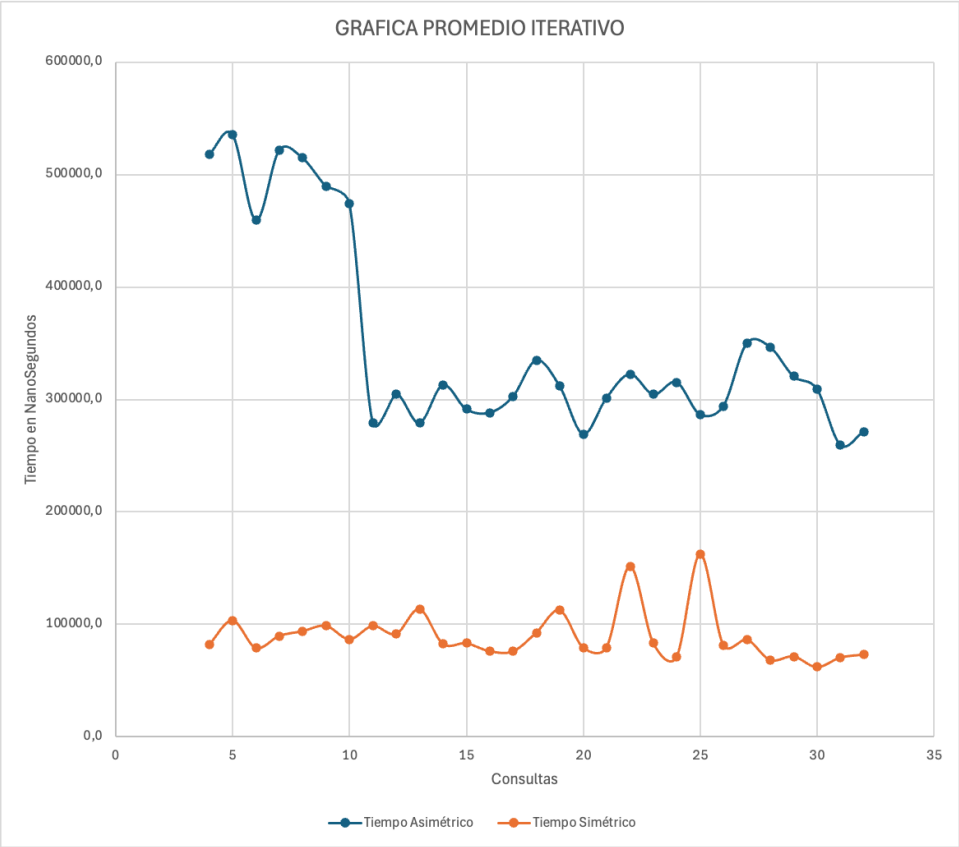

Figura 12. Gráfica ilustrando el carácter anómalo de las solicitudes 3 a 10 del modo iterativo para el reto de 32 dígitos.

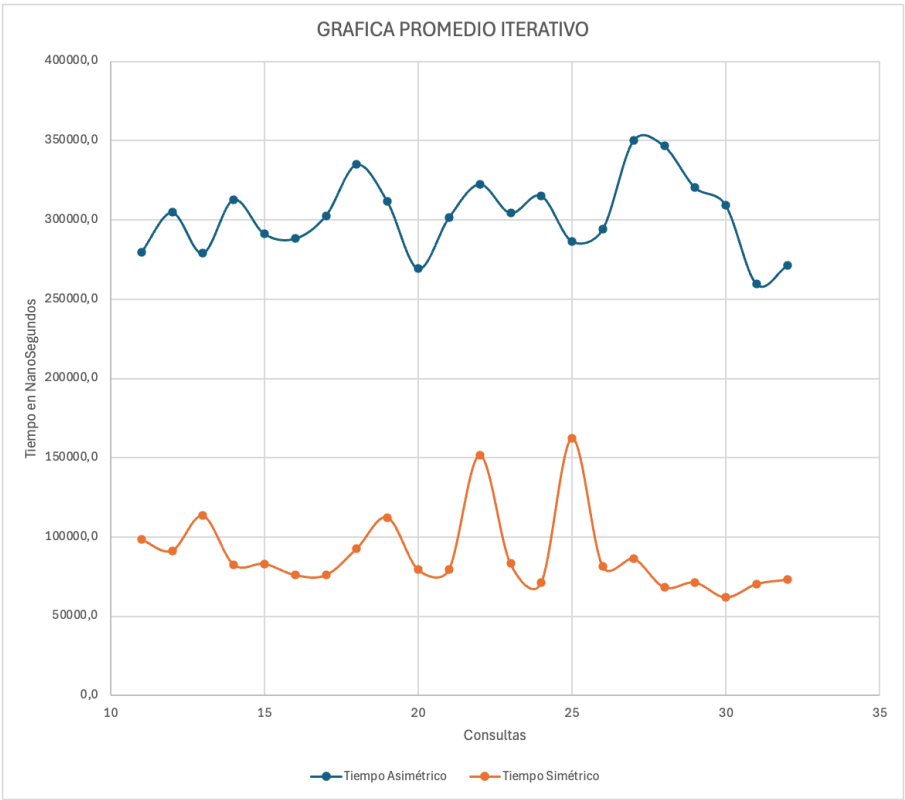

Figura 12. Gráfica ilustrando que de la solicitud 10 en adelante del modo iterativo se estabilizan los tiempos de cifrado para el reto de 32 dígitos.

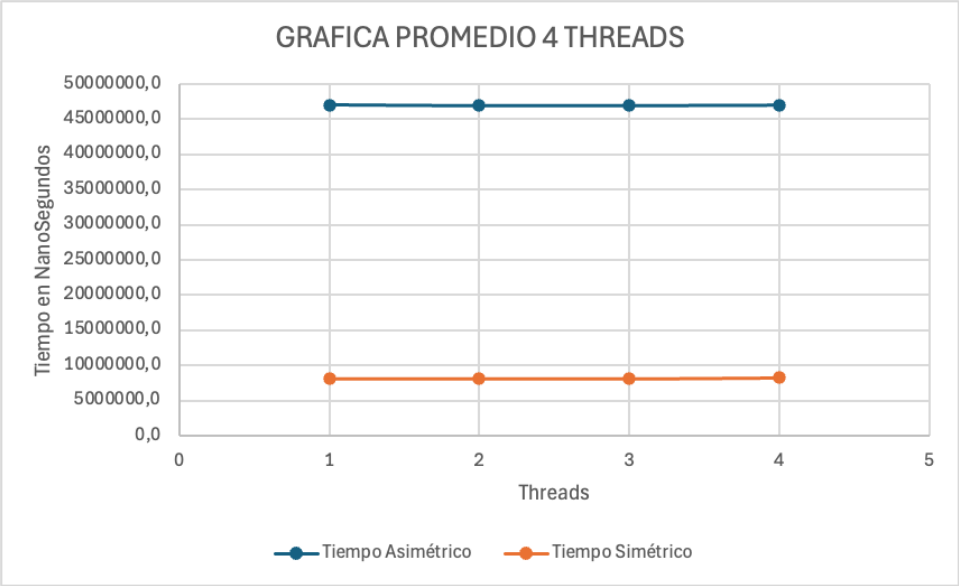

Figura 13. Gráfica de los tiempos promedio de las solicitudes del modo de 4 delegados para el reto de 32 dígitos.

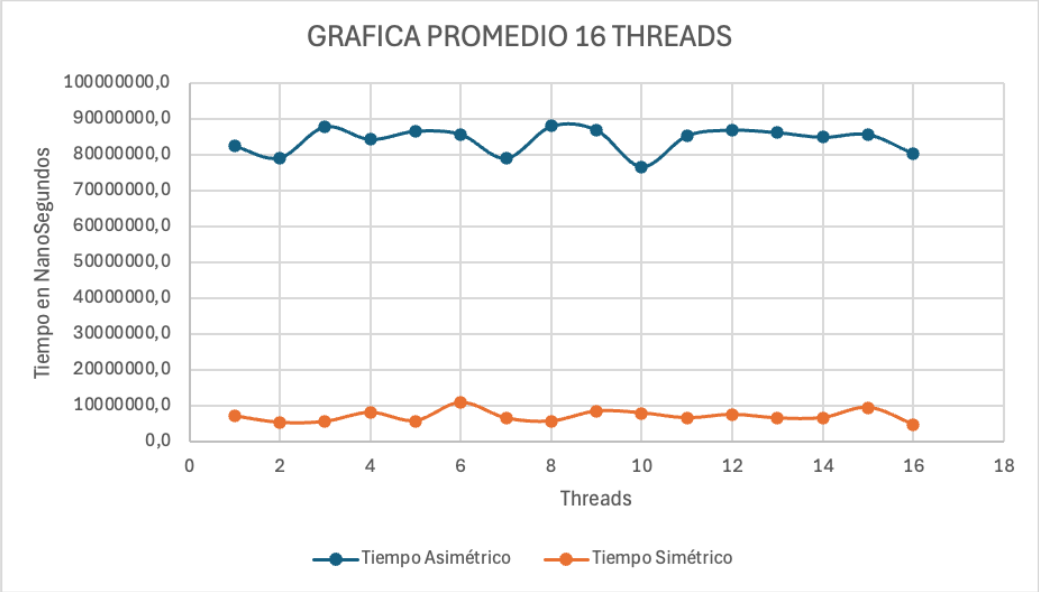

Figura 14. Gráfica de los tiempos promedio de las solicitudes del modo de 16 delegados para el reto de 32 dígitos.

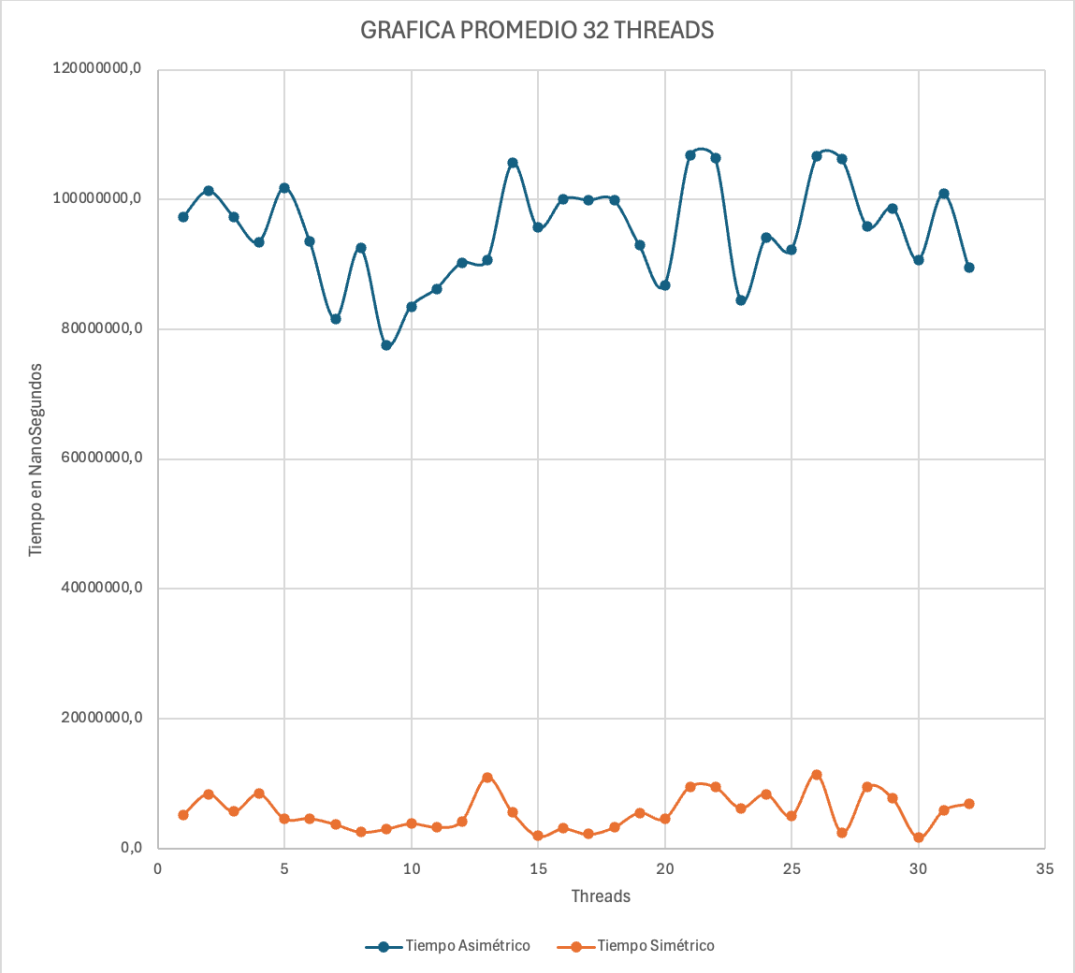

Figura 14. Gráfica de los tiempos promedio de las solicitudes del modo de 32 delegados para el reto de 32 dígitos.

RESUMEN RETO 32 DIGITOS			
/	ASIMÉTRICO	SIMÉTRICO	RELACIÓN
Iterativo	350696,3	90426,3	3,9
4 Threads	46983241,6	8148762,6	5,8
16 Threads	84095798,2	7078938,1	11,9
32 Threads	95008687,0	5598652,3	17,0

Tabla 2. Tabla mostrando el tiempo promedio de cifrado para el reto de 32 dígitos en los casos asimétrico y simétrico de cada modo de ejecución.

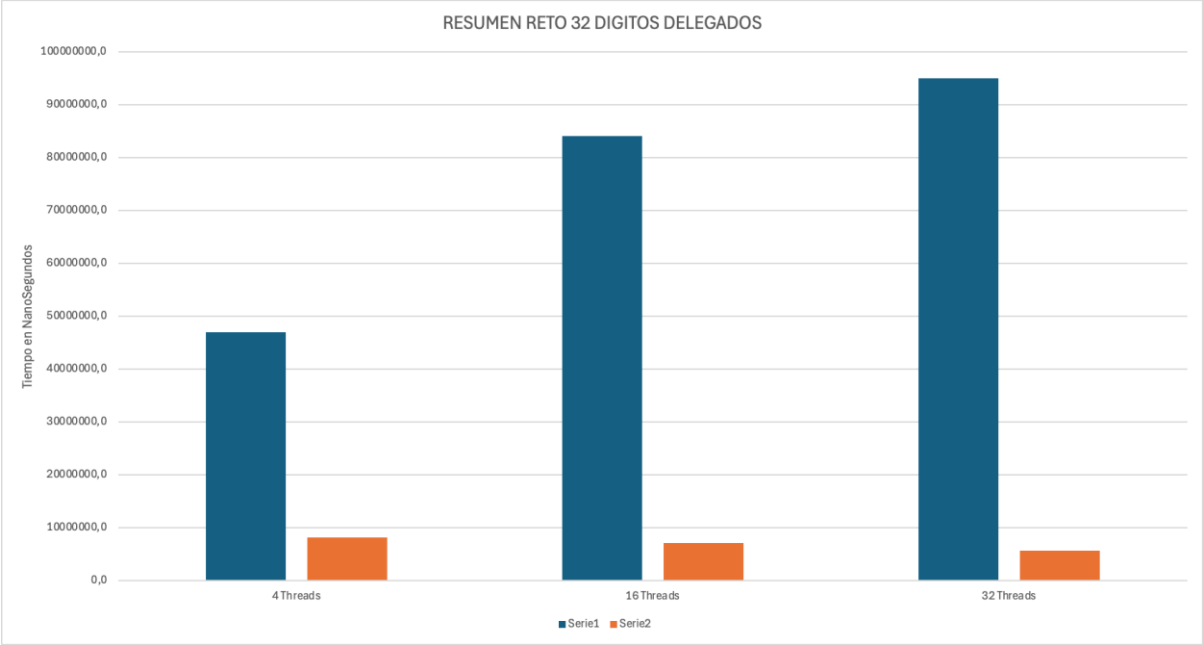

Figura 15. Gráfica de los tiempos promedio de cifrado asimétrico y simétrico por escenario de ejecución para el reto de 32 dígitos (Modo delegados).

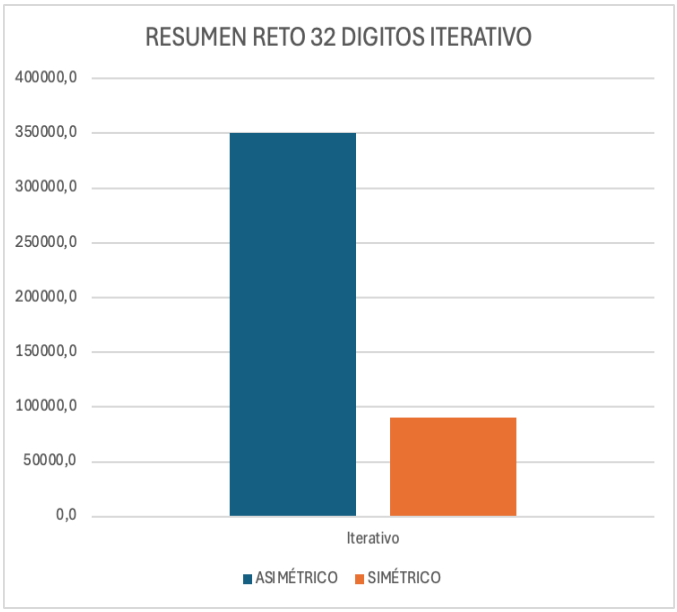

Figura 16. Gráfica de los tiempos promedio de cifrado asimétrico y simétrico por escenario de ejecución para el reto de 32 dígitos (Modo iterativo).

En un principio, como se observa en la figura 1, se resalta que en el escenario iterativo para el reto de 32 dígitos se encontró que el tiempo promedio de la primera solicitud asimétrica era un dato anómalo porque su valor era de un orden de magnitud mayor que el segundo tiempo mayor. Asimismo, tras remover el primer dato, se encontró, como se muestra en la figura 13, que los tiempos de las solicitudes asimétricas 2 a 10 eran diferentes a los datos de las solicitudes 6, llegando a estar a más 200.000 nanosegundos por encima del tiempo de la sexta solicitud. Como esos 7 datos sesgarían el promedio y serían menos

representativas del caso usual, se retiró de los datos anómalos y se tomó, para el caso iterativo, el promedio de las solicitudes 10 en adelante. Note que, conceptualmente, el comportamiento anómalo de estos datos se debe a que la librería de cifrado optimiza por debajo el proceso de cifrado asimétrico cuando se va a utilizar varias veces la llave, lo que la librería implementa guardando la instancia de la llave después del primer uso.

Por otra parte, se resalta que las figuras 12, 13 y 14 muestran que los tiempos promedio por solicitud para los escenarios de ejecución con 4, 16 y 32 delegados respectivamente se mantienen estables sin datos anómalos de discrepancia significativa como en el caso anterior. Por lo tanto, para dichos escenarios se tomó el promedio de todas las solicitudes sin descartar datos.

Luego, partiendo de los promedios mencionados, se realizó la tabla con los tiempos promedio de cifrado asimétrico y simétrico para cada escenario con el reto de 32 dígitos, plasmados en la tabla 2. Después, a partir de esos datos se realizaron las gráficas de las figuras 15 y 16 en las que se comparan los tiempos de cifrado asimétrico y simétrico de cada escenario. Como se observa en ambas figuras, en cada caso se encontró, como se esperaba, que el cifrado asimétrico fue significativamente más demorado que el cifrado simétrico, lo que tiene sentido porque el cifrado simétrico se realiza por bloques. Adicionalmente, se encontró que los tiempos de cifrado fueron mucho más bajos en el escenario iterativo, lo cual es de esperarse debido a que en ese escenario el procesador solo está dedicando sus recursos a ejecutar una solicitud, mientras que en los escenarios de delegados debe repartirlos en varias solicitudes concurrentes. Finalmente, se observa que en la figura 15 se evidencia que el tiempo de cifrado asimétrico en el caso de delgados aumentó con el número de delegados mientras que el de cifrado simétrico disminuyó.

Finalmente, cabe destacar que al observar las gráficas de barras que muestran los tiempos promedio de cifrado para los retos de 24 y 32 dígitos, se observa que los tiempos de cifrado del reto de 24 dígitos son superiores a los del reto de 32 dígitos. Esto parece contradecir la teoría, ya que se esperaría que cifrar datos más grandes requiera más tiempo. Sin embargo, esta discrepancia se debe al uso de diferentes equipos para la recolección de datos. Para los tiempos del reto de 24 dígitos, se utilizó una máquina que tiene 8 años, mientras que para el reto de 32 dígitos se usó una máquina mucho más reciente, de apenas 1 año y medio. Por lo tanto, la diferencia en los tiempos de cifrado se puede atribuir a la diferencia en el hardware: la máquina más moderna y eficiente usada para el reto de 32 dígitos explica por qué sus tiempos de cifrado son menores a pesar del tamaño mayor del reto.

5. **Cálculos y Tiempos.**

Procesador M2 Pro, velocidad 3,6GHZ
Ver Tabla2

ITERATIVO.

Tener cuidado con la interpretación de los datos para una ejecución iterativa. Son muy pocas iteraciones para poder asegurar la velocidad calculada. Hace falta un Servidor que reciba más de 32 Clientes para corroborar estos resultados.

$$\text{Tiempo asimétrico en segundos} = \frac{350696{,}3}{1 * 10^9} s$$

$$\textit{Tiempo simétrico en segundos} = \frac{90426{,}3}{1 * 10^9} s$$

$$\text{Velocidad de cifrado asimétrico} = \frac{32\, reto}{\frac{350696{,}3}{1 * 10^9} s} = \frac{32 * 10^9}{350696{,}3} \frac{retos}{segundo}$$

$$\approx 91247 \frac{retos}{segundo}$$

$$= \frac{32 * 10^9}{350696{,}3} \frac{retos}{segundo} * \frac{1}{3{,}6 * 10^9} \frac{\text{segundos}}{\text{ciclo}} \approx 2{,}5 * 10^{-5} \frac{retos}{ciclo}$$

$$\text{Velocidad de cifrado simétrico} = \frac{32\, reto}{\frac{90426{,}3}{1 * 10^9} s} = \frac{32 * 10^9}{90426{,}3} \frac{retos}{segundo} \approx 353879 \frac{retos}{segundo}$$

$$= \frac{32 * 10^9}{90426{,}3} \frac{retos}{segundo} * \frac{1}{3{,}6 * 10^9} \frac{\text{segundos}}{\text{ciclo}} \approx 9{,}9 * 10^{-5} \frac{retos}{ciclo}$$

4 THREADS.

$$\text{Tiempo asimétrico en segundos} = \frac{46983241{,}6}{1 * 10^9} s$$

$$\textit{Tiempo simétrico en segundos} = \frac{8148762{,}6}{1 * 10^9} s$$

$$\text{Velocidad de cifrado asimétrico} = \frac{1\, reto}{\frac{46983241{,}6}{1 * 10^9} s} = \frac{1 * 10^9}{46983241{,}6} \frac{retos}{segundo}$$

$$\approx 21 \frac{retos}{segundo}$$

$$= \frac{1 * 10^9}{46983241{,}6} \frac{retos}{segundo} * \frac{1}{3{,}6 * 10^9} \frac{\text{segundos}}{\text{ciclo}} \approx 5{,}9 * 10^{-9} \frac{retos}{ciclo}$$

$$\text{Velocidad de cifrado simétrico} = \frac{1\, reto}{\frac{8148762{,}6}{1 * 10^9} s} = \frac{1 * 10^9}{8148762{,}6} \frac{retos}{segundo}$$

$$\approx 123 \frac{retos}{segundo}$$

$$= \frac{1 * 10^9}{8148762{,}6} \frac{retos}{segundo} * \frac{1}{3{,}6 * 10^9} \frac{\text{segundos}}{\text{ciclo}} \approx 3{,}4 * 10^{-8} \frac{retos}{ciclo}$$

16 THREADS.

$$\text{Tiempo asimétrico en segundos} = \frac{84095798{,}2}{1 * 10^9} s$$

$$\textit{Tiempo simétrico en segundos} = \frac{7078938{,}1}{1 * 10^9} s$$

$$\text{Velocidad de cifrado asimétrico} = \frac{1\,reto}{\frac{84095798{,}2}{1 * 10^9} s} = \frac{1 * 10^9}{84095798{,}2} \frac{retos}{segundo}$$

$$\approx 12 \frac{retos}{segundo}$$

$$= \frac{1 * 10^9}{84095798{,}2} \frac{retos}{segundo} * \frac{1}{3{,}6 * 10^9} \frac{\text{segundos}}{\text{ciclo}} \approx 3{,}3 * 10^{-9} \frac{retos}{ciclo}$$

$$\text{Velocidad de cifrado simétrico} = \frac{1\,reto}{\frac{7078938{,}1}{1 * 10^9} s} = \frac{1 * 10^9}{7078938{,}1} \frac{retos}{segundo}$$

$$\approx 141 \frac{retos}{segundo}$$

$$= \frac{1 * 10^9}{7078938{,}1} \frac{retos}{segundo} * \frac{1}{3{,}6 * 10^9} \frac{\text{segundos}}{\text{ciclo}} \approx 3{,}9 * 10^{-8} \frac{retos}{ciclo}$$

32 THREADS.

$$\text{Tiempo asimétrico en segundos} = \frac{95008687{,}0}{1 * 10^9} s$$

$$\textit{Tiempo simétrico en segundos} = \frac{5598652{,}3}{1 * 10^9} s$$

$$\text{Velocidad de cifrado asimétrico} = \frac{1\,reto}{\frac{95008687{,}0}{1 * 10^9} s} = \frac{1 * 10^9}{95008687{,}0} \frac{retos}{segundo}$$

$$\approx 11 \frac{retos}{segundo}$$

$$= \frac{1 * 10^9}{95008687{,}0} \frac{retos}{segundo} * \frac{1}{3{,}6 * 10^9} \frac{\text{segundos}}{\text{ciclo}} \approx 2{,}9 * 10^{-9} \frac{retos}{ciclo}$$

$$\text{Velocidad de cifrado simétrico} = \frac{1\,reto}{\frac{5598652{,}3}{1 * 10^9} s} = \frac{1 * 10^9}{5598652{,}3} \frac{retos}{segundo}$$

$$\approx 179 \frac{retos}{segundo}$$

$$= \frac{1 * 10^9}{5598652{,}3} \frac{retos}{segundo} * \frac{1}{3{,}6 * 10^9} \frac{\text{segundos}}{\text{ciclo}} \approx 5{,}0 * 10^{-8} \frac{retos}{ciclo}$$