Informe caso de estudio 3

Canales seguros

Integrantes

Daniel Felipe Díaz Moreno – 202210773 – d.diazm

Sara Sofía Cárdenas Rodríguez - 202214907 – ss.cardenas

Sara Benavides Mora – 202022464 – s.benavidesm

Infraestructura Computacional

Sandra Julieta Rueda Rodríguez

Departamento de Ingeniería de Sistemas y Computación

Universidad de Los Andes

Bogotá, Colombia

1. RESPUESTAS A LAS PREGUNTAS

1.1. En el protocolo descrito el cliente conoce la llave pública del servidor (K_w+). ¿Cuál es el método comúnmente usado para obtener estas llaves públicas para comunicarse con servidores web?

El método comúnmente usado para que el cliente conozca la llave pública del servidor (K_w+) y pueda comunicarse con servidores web se da a través del intercambio de certificados SSL/TLS durante el proceso de handshake. Este proceso se basa en la infraestructura de clave pública (PKI), donde los servidores web tienen un certificado digital que contiene su llave pública, firmado por una autoridad de certificación (CA) confiable. Cuando un cliente se conecta a un servidor, el servidor envía su certificado al cliente, que contiene la llave pública del servidor. El cliente puede verificar la autenticidad del certificado utilizando la cadena de certificados hasta una CA de confianza, y luego utiliza la llave pública del servidor para establecer una comunicación segura (Tanenbaum, 2003).

1.2. ¿Por qué es necesario cifrar G y P con la llave privada?

Es necesario cifrar G y P con la llave privada del servidor para garantizar la integridad y autenticidad de estos valores al momento de realizar el intercambio Diffie-Hellman. Cuando el servidor cifra G y P con su llave privada, cualquier cliente puede descifrar estos valores utilizando la llave pública del servidor para asegurarse de que no han sido modificados durante la transmisión. En consecuencia, se previenen ataques de manipulación en el intercambio de parámetros Diffie-Hellman, asegurando que el cliente obtenga los valores originales y confiables de G y P directamente del servidor. Además, la firma digital de estos valores también garantiza que provienen del servidor auténtico, evitando ataques de suplantación (Stallings, 2017).

1.3. El protocolo Diffie-Hellman garantiza "Forward Secrecy", presente un caso en el contexto del sistema Banner de la Universidad donde sería útil tener esta garantía, justifique su respuesta (por qué es útil en ese caso).

El concepto de "Forward Secrecy" en el contexto del protocolo Diffie-Hellman implica que incluso si las claves privadas se ven comprometidas a largo plazo, las conversaciones pasadas y futuras no podrán ser descifradas ya que la llave simétrica cambia y es temporal (Stallings, 2017).

Entonces, en el sistema Banner de la Universidad la funcionalidad de "Forward Secrecy" sería útil en el momento en el que los profesores realicen el cargue de las notas, pues es esencial garantizar que la confidencialidad e integridad de las calificaciones se mantenga incluso si las claves privadas del sistema se comprometen en el futuro. Por ello, si un atacante ingresa al sistema, solo podrá desencriptar y alterar las comunicaciones que se cifraron con la llave simétrica temporal, es decir, las calificaciones de hace pocos días u horas, mientras que el resto de los mensajes que podrían hacer referencia a las calificaciones pasadas de los estudiantes de dicho docente no podrán ser desencriptados, al no conocer las llaves temporales que lo hicieron. Esto asegura que los cambios percibidos por profesores y alumnos sean mínimos y puedan ser más fácilmente corregidos, a comparación de si el sistema no tuviera esta característica.

2. ESCENARIOS DEL CLIENTE

2.1. Verificar la firma

Acción	Verificar la firma			
Cantidad de clientes	**Tiempo (ms)**	**Tiempo promedio por cliente (ms)**	**Firmas verificadas por segundo**	**Operación**
4	340,4885	85,122125	11,74782702	4/(340,4885*0,001)
16	3895,4054	243,4628375	4,107403045	16/(3895,4054*0,001)
32	15377,6481	480,5515031	2,080942404	32/(15377,6481*0,001)
64	28496,0603	445,2509422	2,245924501	64/(28496,0603*0,001)
128	82719,1210	646,2431328	1,547405224	128/(82719,121*0,001)

2.2. Calcular G^y

Acción	Calcular G^y			
Cantidad de clientes	**Tiempo (ms)**	**Tiempo promedio por cliente (ms)**	**Cálculos por segundo**	**Operación**
4	6,4352	1,6088	621,5813028	4/(6,4352*0,001)
16	35,0447	2,19029375	456,5597651	16/(35,0447*0,001)
32	1867,6261	58,36331563	17,13405055	32/(1867,6261*0,001)
64	4286,5638	66,97755938	14,93037383	64/(4286,5638*0,001)
128	15680,9887	122,5077242	8,162750605	128/(15680,9887*0,001)

2.3. Cifrar la consulta

Acción	Cifrar la consulta			
Cantidad de clientes	**Tiempo (ms)**	**Tiempo promedio por cliente (ms)**	**Consultas cifradas por segundo**	**Operación**
4	4,1249	1,031225	969,7204781	4/(4,1249*0,001)
16	54,6600	3,41625	292,7186242	16/(54,66*0,001)
32	94,8519	2,964121875	337,3680443	32/(94,8519*0,001)
64	420,7748	6,57460625	152,1003634	64/(420,7748*0,001)
128	1941,3927	15,16713047	65,93204971	128/(1941,3927*0,001)

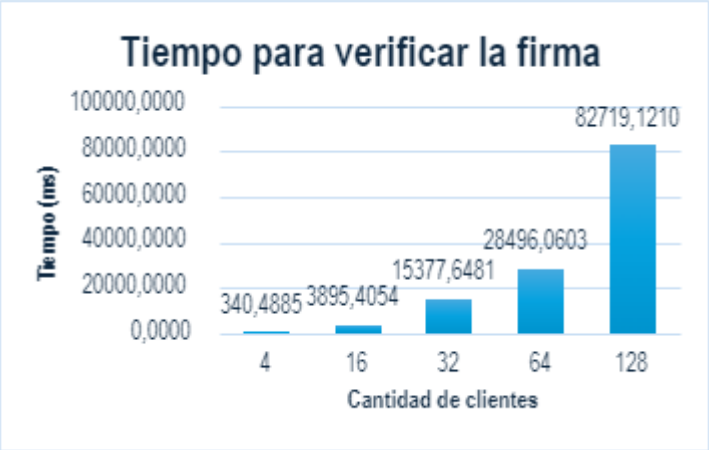
2.4. Generar el código de autenticación

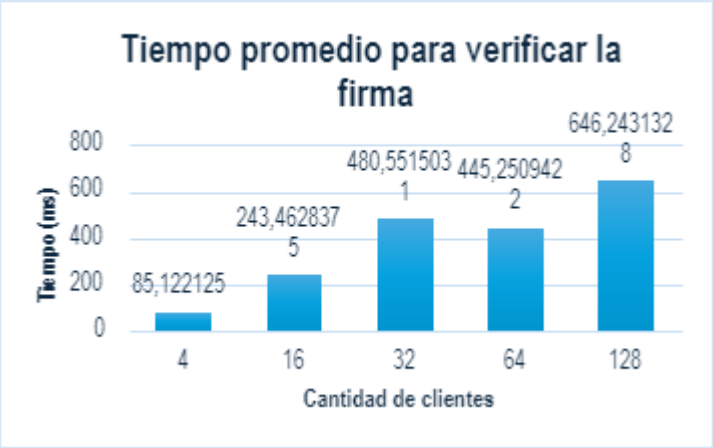
Acción	Generar el código de autenticación

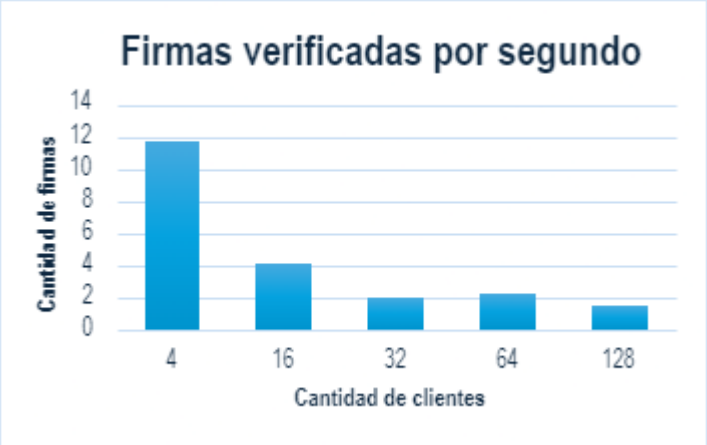
Cantidad de clientes	Tiempo (ms)	Tiempo promedio por cliente (ms)	Códigos generados por segundo	Operación
4	1,4091	0,352275	2838,691363	4/(1,4091*0,001)
16	3,7021	0,23138125	4321,87137	16/(3,7021*0,001)
32	4,8255	0,150796875	6631,437157	32/(4,8255*0,001)
64	6,6311	0,103610938	9651,490703	64/(6,6311*0,001)
128	10,5359	0,082311719	12148,93839	128/(10,5359*0,001)

3. GRÁFICAS DEL CLIENTE

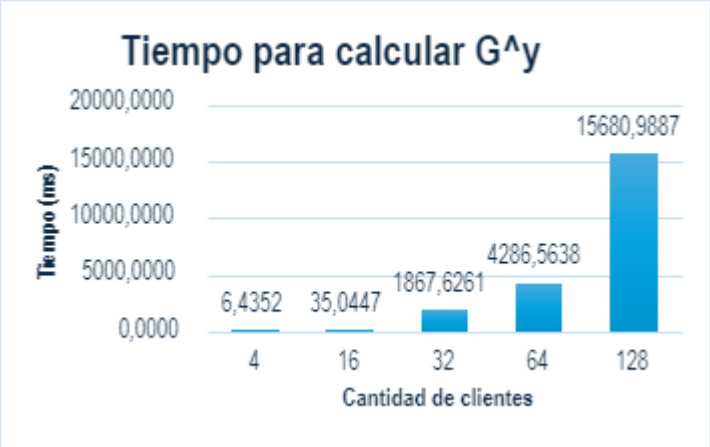
3.1. Verificar la firma

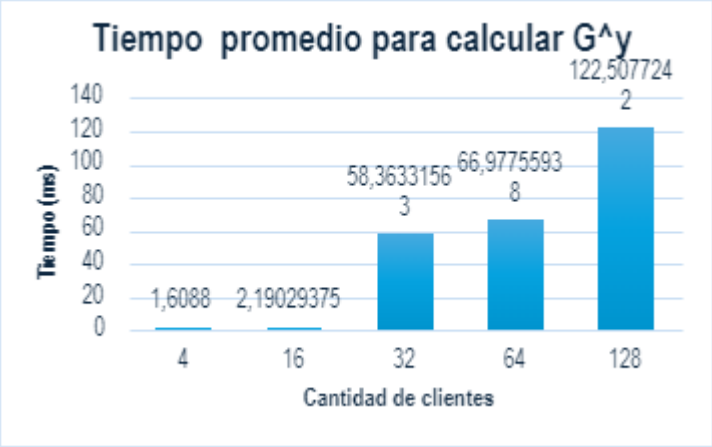


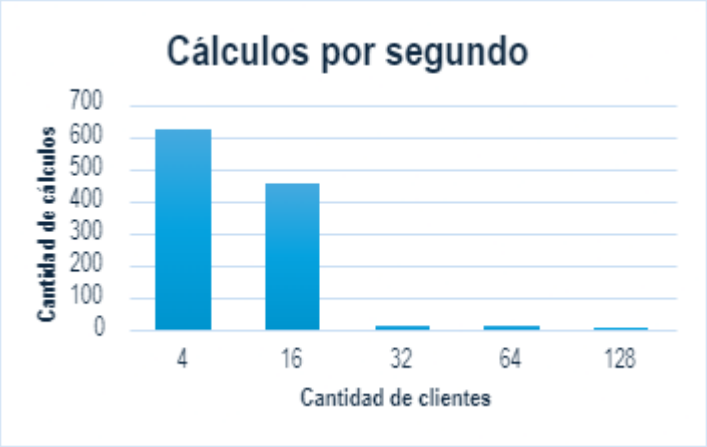




3.2. Calcular G^y

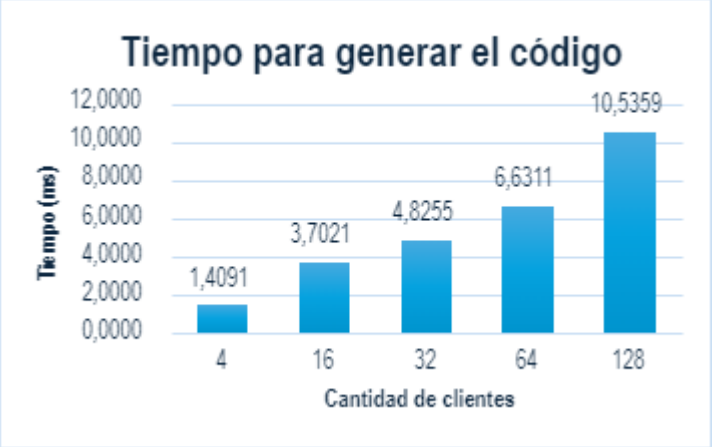






3.3. Cifrar la consulta

3.4. Generar el código de autenticación



4. ESCENARIOS DEL SERVIDOR

4.1. Generar la firma

Acción	Generar la firma			
Cantidad de clientes	**Tiempo (ms)**	**Tiempo promedio por cliente (ms)**	**Firmas generadas por segundo**	**Operación**
4	335,0971	83,774275	11,93683861	4/(335,0971*0,001)
16	3878,7628	242,422675	4,125026671	16/(3878,7628*0,001)
32	15249,5561	476,5486281	2,098421737	32/(15249,5561*0,001)
64	40259,4226	629,0534781	1,589689962	64/(40259,4226*0,001)
128	290186,0539	2267,078546	0,441096318	128/(290186,0539*0,001)

4.2. Descifrar la consulta

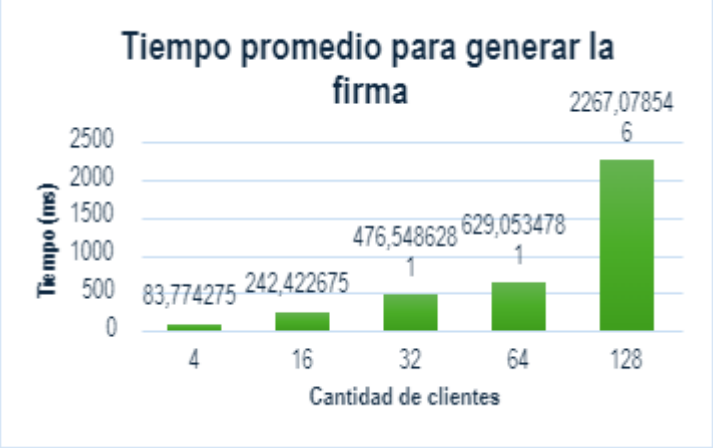
Acción	Descifrar la consulta			
Cantidad de clientes	**Tiempo (ms)**	**Tiempo promedio por cliente (ms)**	**Consultas descifradas por segundo**	**Operación**
4	5,2580	1,3145	760,7455306	4/(5,258*0,001)
16	61,8070	3,8629375	258,8703545	16/(61,807*0,001)
32	280,7760	8,77425	113,969855	32/(280,776*0,001)
64	1035,8882	16,18575313	61,78272906	64/(1035,8882*0,001)
128	3425,3894	26,76085469	37,36801428	128/(3425,3894*0,001)

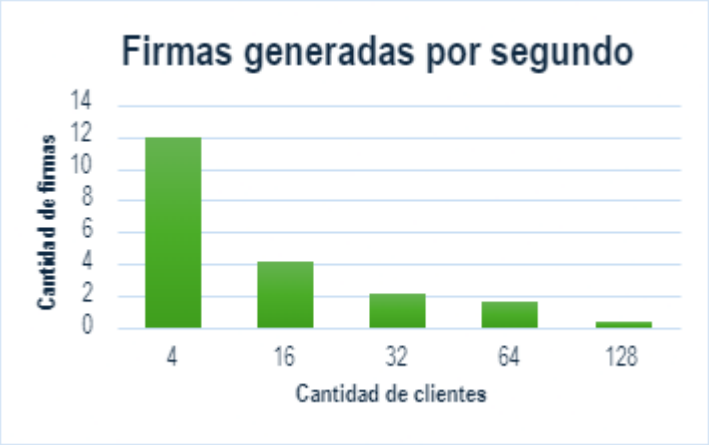
4.3. Verificar el código de autenticación

Acción	Verificar el código de autenticación			
Cantidad de clientes	**Tiempo (ms)**	**Tiempo promedio por cliente (ms)**	**Códigos verificados por segundo**	**Operación**
4	2,2593	0,564825	1770,459877	4/(2,2593*0,001)
16	5,6133	0,35083125	2850,373221	16/(5,6133*0,001)
32	8,8329	0,276028125	3622,819233	32/(8,8329*0,001)
64	11,4006	0,178134375	5613,739628	64/(11,4006*0,001)
128	17,0381	0,133110156	7512,574759	128/(17,0381*0,001)

5. GRÁFICAS DEL SERVIDOR

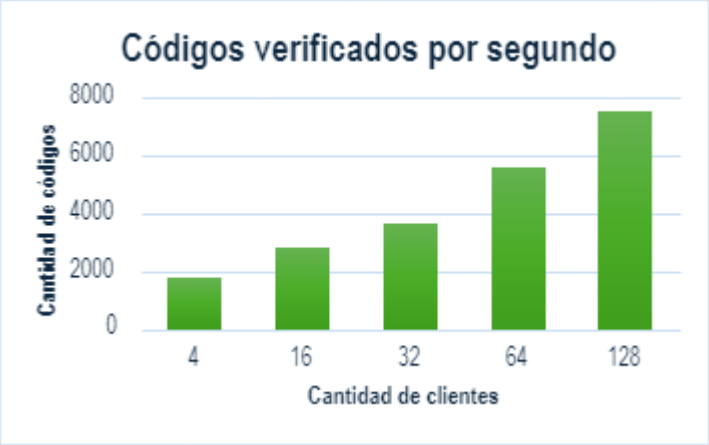
5.1. Generar la firma





5.2. Descifrar la consulta

5.3. Verificar el código de autenticación



6. ANÁLISIS DE RESULTADOS

6.1. ANÁLISIS CLIENTE

La verificación de la firma y el cálculo de G^y son operaciones que requieren más recursos de procesamiento y que requieren más tiempo de ejecución para completarse, por ende, se observa que van disminuyendo en eficiencia a medida que el número de clientes aumenta. Por otro lado, la generación del código de autenticación es la operación más eficiente y, a su vez, la que menos se vio afectada por el aumento de carga de clientes porque la cantidad de códigos generados por segundo sigue siendo alta. Ahora bien, el proceso de cifrar la consulta mostró una tendencia en la que se volvió más lenta a medida que se aumentaba el número de clientes, pero, aun así, pudo manejar este aumento en la carga.

6.2. ANÁLISIS SERVIDOR

La generación de la firma y el descifrar la consulta son operaciones que disminuyeron su eficiencia con el aumento de los clientes, mostrando así que necesitan una mayor cantidad de recursos mientras que la verificación del código de autenticación tuvo un comportamiento en el que no se vió altamente afectado por el aumento en la carga de clientes y, por ende, fue la operación más eficiente porque aunque el tiempo promedio de verificación por cliente aumentó ligeramente con la carga, la cantidad de códigos verificados por segundo fue alta.

6.3. ANÁLISIS GENERAL

La implementación mostró que puede ser escalable hasta cierto punto, ya que el tiempo promedio por cliente y la cantidad de operaciones por segundo disminuyen gradualmente según se aumentaba el número de clientes, mostrando que el sistema puede manejar más carga. Sin embargo, otras operaciones como el cifrado y descifrado de la consulta mostraron no ser tan escalables, así que, requieren de ajustes adicionales para poder manejar la carga.

Por otro lado, el sistema tuvo un buen desempeño porque las operaciones críticas como la verificación y generación de la firma, el cálculo de G^y y la verificación del código de autenticación pudieron manejar una cantidad razonable de clientes por segundo. Este comportamiento es destacable ya que la generación y verificación de firmas usando RSA son operaciones que exigen más recursos que las de cifrado y descifrado simétrico. Ahora bien, las operaciones de cifrado y descifrado de la consulta fueron más sensibles al aumento de carga y

si el número de clientes aumenta mucho, podrían convertirse en cuellos de botella y empeorar el desempeño de la implementación. Esto se debe a que se usó el algoritmo de cifrado simétrico AES en modo CBC el cual, es especialmente eficiente cuando se usa con bloques pequeños de datos.

Como se usó el algoritmo de código de autenticación HMACSHA256 para garantizar la integridad de los datos transmitidos entre el cliente y el servidor, se pudo observar un mejor rendimiento para generar y verificar los códigos de autenticación porque este algoritmo es más eficiente en términos de recursos en comparación con RSA y, aunque el tiempo si aumentó a mayo carga, el impacto generado fue menor al que tuvieron las operaciones de la firma digital.

7. ESPECIFICACIONES DE LA MÁQUINA UTILIZADA

Usando los datos obtenidos anteriormente que se mostraron en las tablas, se procede a realizar los cálculos para estimar cuántas consultas puede cifrar la máquina en la que se realizaron las pruebas, cuántos códigos de autenticación puede calcular y cuántas verificaciones de firma hace por segundo. Para hacer los cálculos se tuvo en cuenta que el procesador de la máquina es un Intel(R) Core(TM) i7-10750H con una velocidad de 2.60GHz y que los tiempos del sistema se dieron en ms. Cada sección muestra la fórmula que se utilizará para los cálculos y posteriormente se muestra el proceso realizado para cada uno de los casos. Los resultados se dan en segundos.

7.1. Estimación de la cantidad de consultas que puede cifrar la máquina

$$Consultas\ cifradas\ (seg) = \frac{número\ de\ clientes}{tiempo\ en\ ms * 0.001} = \frac{número\ de\ clientes}{tiempo\ en\ s}$$

$$Para\ 4\ clientes = \frac{4}{4.12 * 0.001} = 969.72$$

$$Para\ 16\ clientes = \frac{16}{54.66 * 0.001} = 292.71$$

$$Para\ 32\ clientes = \frac{32}{94.85 * 0.001} = 337.36$$

7.2. Estimación de la cantidad de códigos de autenticación que puede calcular

$$Códigos\ de\ autenticación\ (seg) = \frac{número\ de\ clientes}{tiempo\ en\ ms * 0.001} = \frac{número\ de\ clientes}{tiempo\ en\ s}$$

$$Para\ 4\ clientes = \frac{4}{1.40 * 0.001} = 2838.69$$

$$Para\ 16\ clientes = \frac{16}{3.70 * 0.001} = 4321.87$$

$$Para\ 32\ clientes = \frac{32}{4.82 * 0.001} = 6631.43$$

7.3. Estimación de la cantidad de verificaciones de firmas por segundo

$$Verificaciones\ de\ firma\ (seg)\ = \frac{número\ de\ clientes}{tiempo\ en\ ms * \ 0.001} = \frac{número\ de\ clientes}{tiempo\ en\ s}$$

$$Para\ 4\ clientes = \frac{4}{340.48 * 0.001} = 11.74$$

$$Para\ 16\ clientes = \frac{16}{3895.40 * 0.001} = 4.10$$

$$Para\ 32\ clientes = \frac{32}{15377.64 * 0.001} = 2.08$$

De esta manera, puede evidenciarse que la operación más rápida de las tres es calcular códigos de verificación, seguida de cifrar consultas, siendo la operación más costosa la de verificar una firma. Esto es acorde a la teoría.

8. REFERENCIAS

Stallings, W. (2017). *Cryptography and Network Security Principles and Practice* (7.ª ed.).

Tanenbaum, A. (2003). *Computer Networks* (4.ª ed.).