# Caso 2 - Infraestructura computacional

## Reporte

Pedro Lobato - 202012490
Sebastian Ospino - 201913643

# Índice general


Chapter 1 **Respuesta** Page 2

1.1 Pregunta 1 2

1.2 Pregunta 2 2

Chapter 2 **Graficas** Page 3

2.1 Caso 4 3

2.2 Caso 16 4

2.3 Caso 32 6

Chapter 3 **Tablas** Page 8

3.1 Caso 4 8

3.2 Caso 16 8

3.3 Caso 32 9

Chapter 4 **Resultados** Page 12

Chapter 5 **Informacion** Page 13

# Capítulo 1

# Respuesta

## 1.1. Pregunta 1

En el protocolo descrito el cliente conoce la llave pública del servidor (K_w+). ¿Cuál es la manera común de enviar estas llaves para comunicaciones con servidores web?: A través de un certificado digital. Para esto, se hace uso de una PKI (Public Key Infrastructure). La PKI cuenta con una entidad central, una entidad certificadora que asume el rol de tercero confiable. Esta entidad crea certificados digitales para los clientes que lo solicitan (en este caso los servidores). Se espera que una entidad certificadora garantice la identidad de sus clientes y la relación de una identidad con una llave pública. Para que un cliente y un servidor puedan establecer una conexión de comunicación segura (en el contexto de la arquitectura cliente servidor), debe poder llevarse a cabo, de manera exitosa, lo que se conoce como un handshake (la parte inicial del protocolo SSL o TLS), que no es más que una serie de mensajes que son enviados entre el cliente y el servidor para establecer los algoritmos y las llaves de encriptado, autenticación e integridad de la información. La primera parte de este handshake consiste en el cliente enviándole un mensaje de inicialización al servidor, la segunda parte consiste en el servidor respondiéndole al cliente con su certificado digital, el cual incluye la llave pública del servidor. Posteriormente suceden muchas cosas más pero no vienen al caso de esta pregunta.

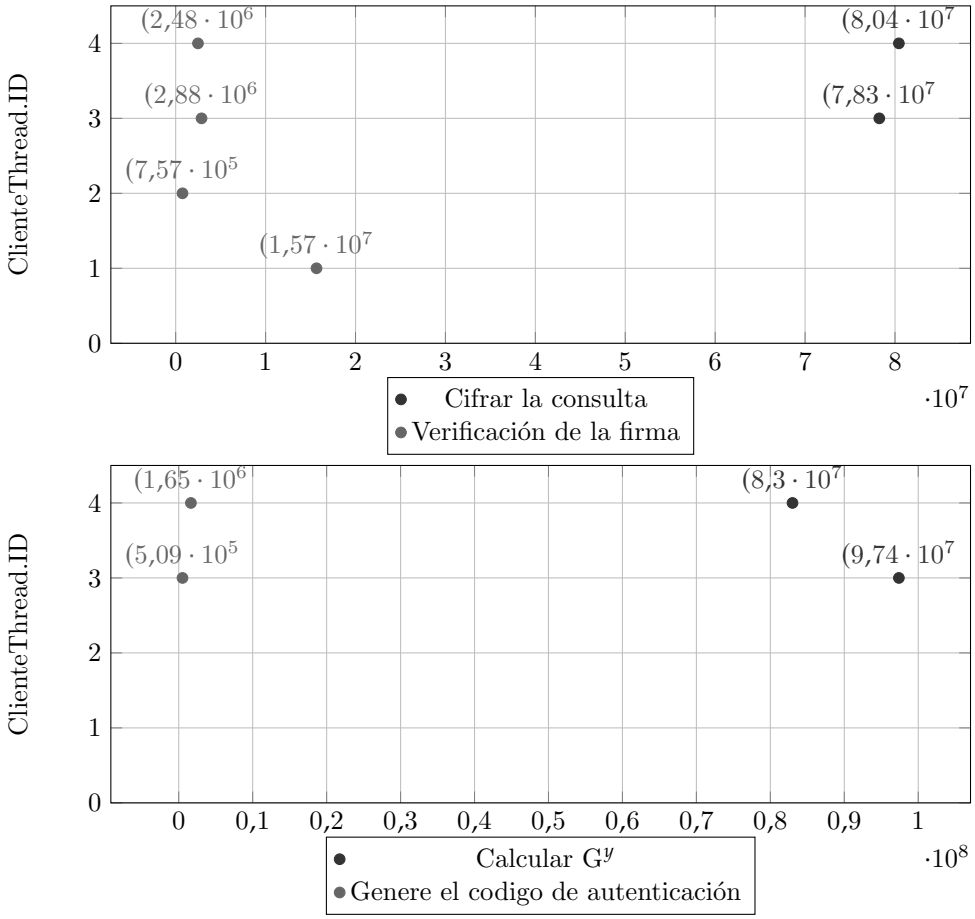
## 1.2. Pregunta 2

El protocolo Diffie-Hellman garantiza "Forward Secrecy", explique en qué consiste esta garantía: "Forward Secrecy" es una característica del protocolo SSL/TLS que impide que un atacante pueda descifrar los datos de sesiones históricas o futuras si es capaz de robar las claves privadas utilizadas en una sesión concreta. Esto se consigue utilizando claves de sesión únicas que se generan con frecuencia y automáticamente. Más específicamente, el protocolo SSL/TLS utiliza el protocolo Diffie-Hellman para establecer llaves maestras secretas de encriptado simétrico por cada conexión o sesión establecida entre un cliente y un servidor. Esto lo logra al volver efímero el número entero secreto que ayuda a la generación segura de la llave maestra, es decir, se generan por cada sesión o conexión de una manera aleatoria y no queda registro de estos una vez se haya generado la llave maestra. Entonces, si un atacante lograse obtener, de alguna manera, este número entero secreto, solo podría desencriptar la información que se transmitió en dicha sesión.

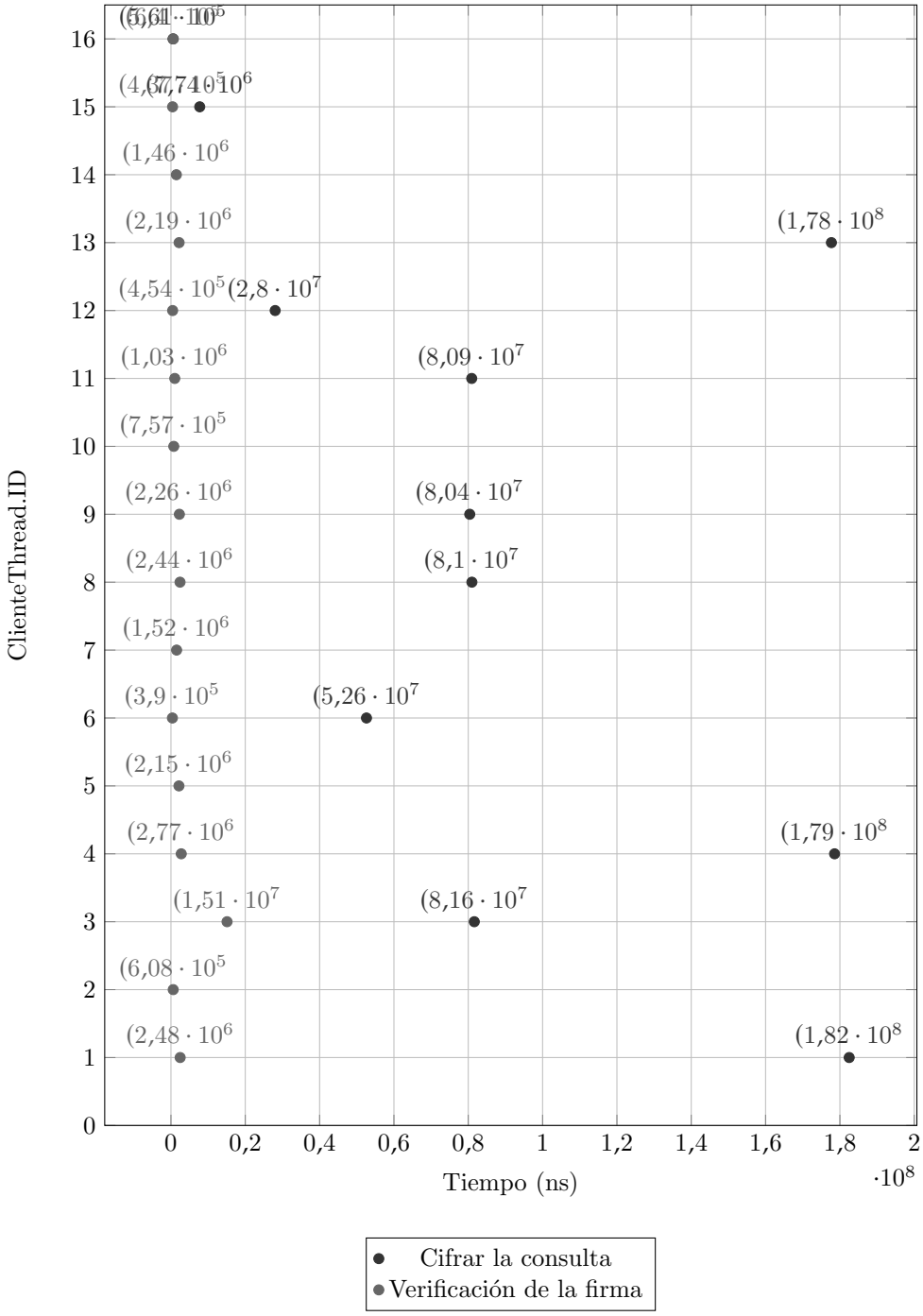
# Capítulo 2

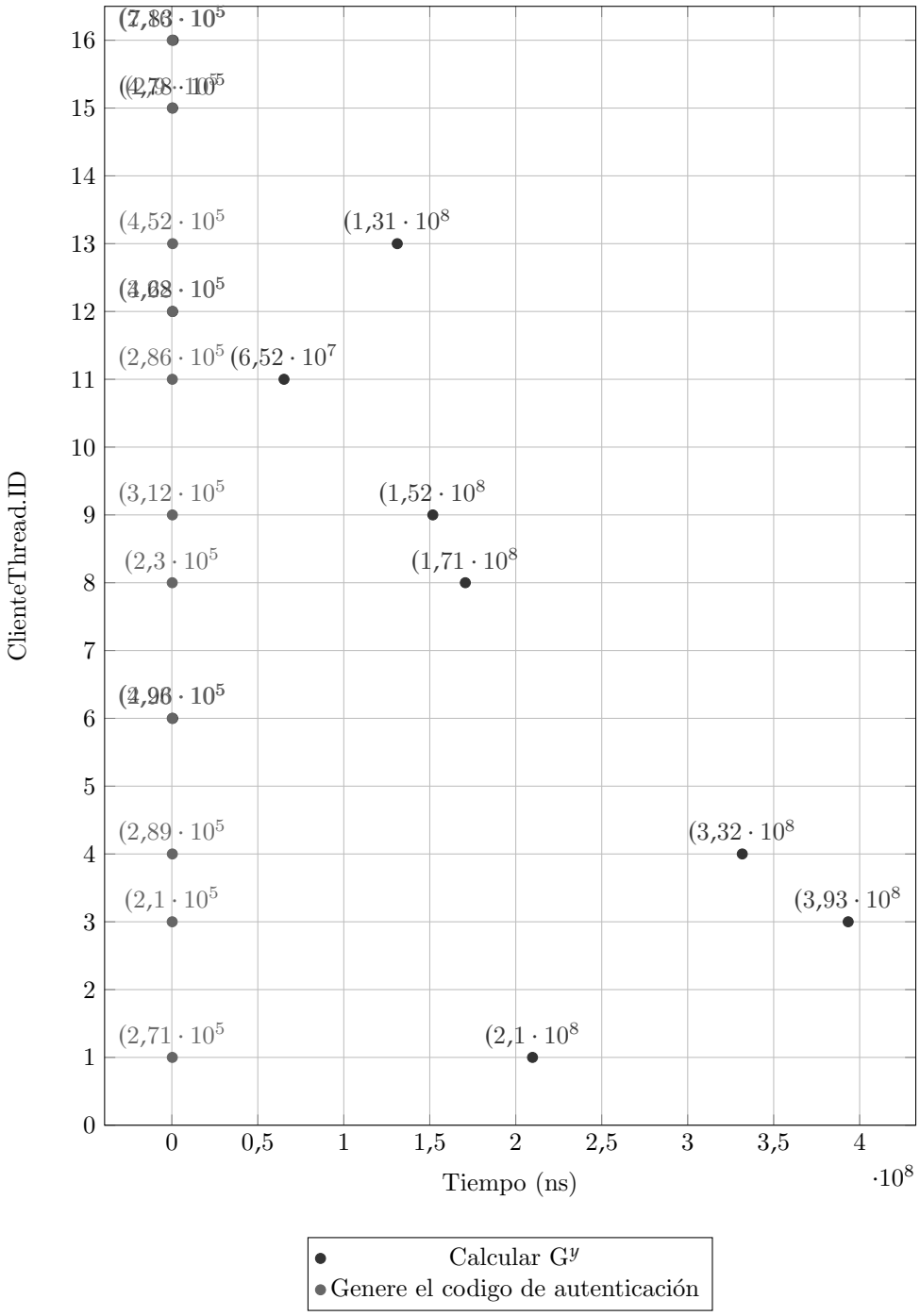
# Graficas

## 2.1. Caso 4

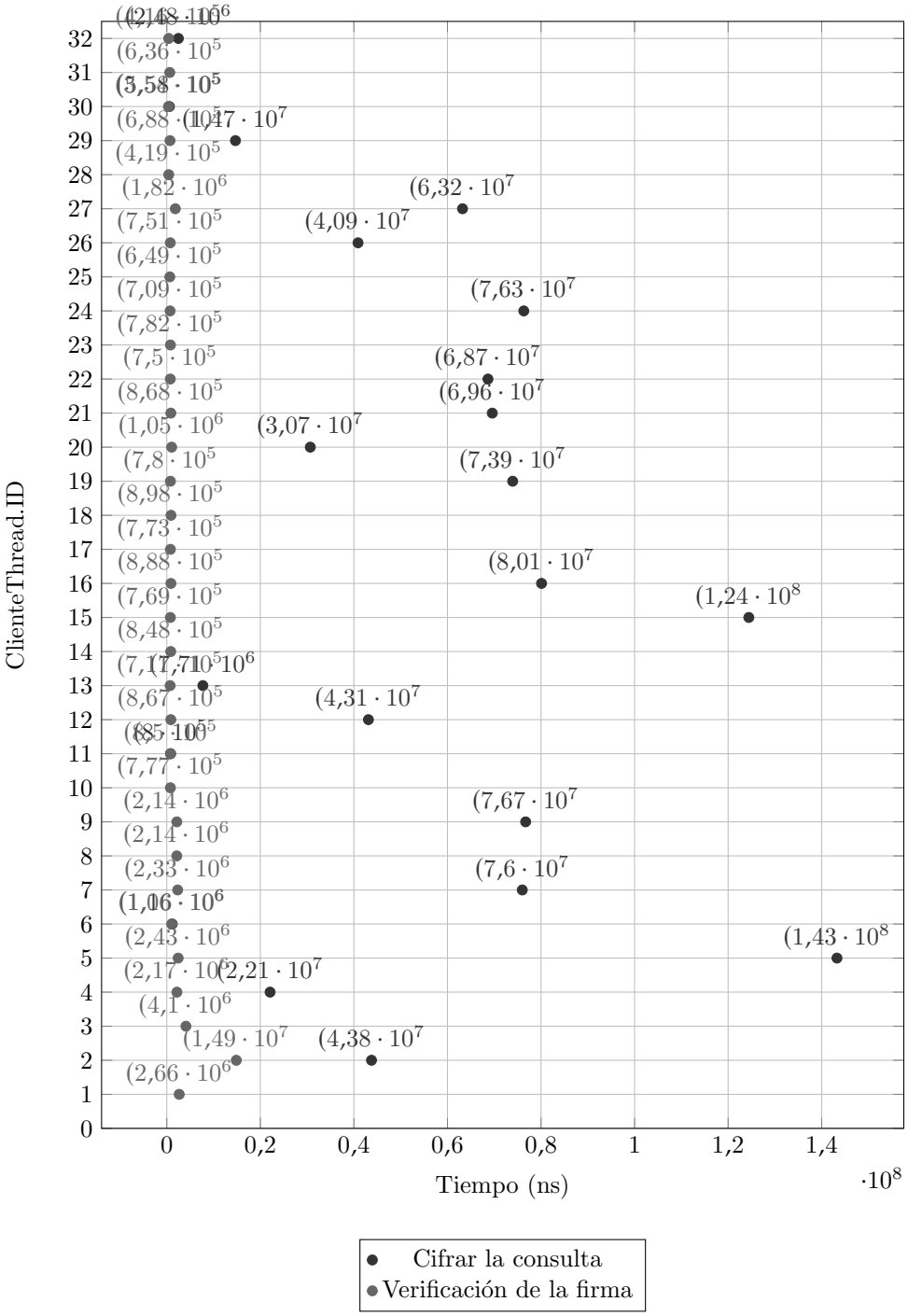


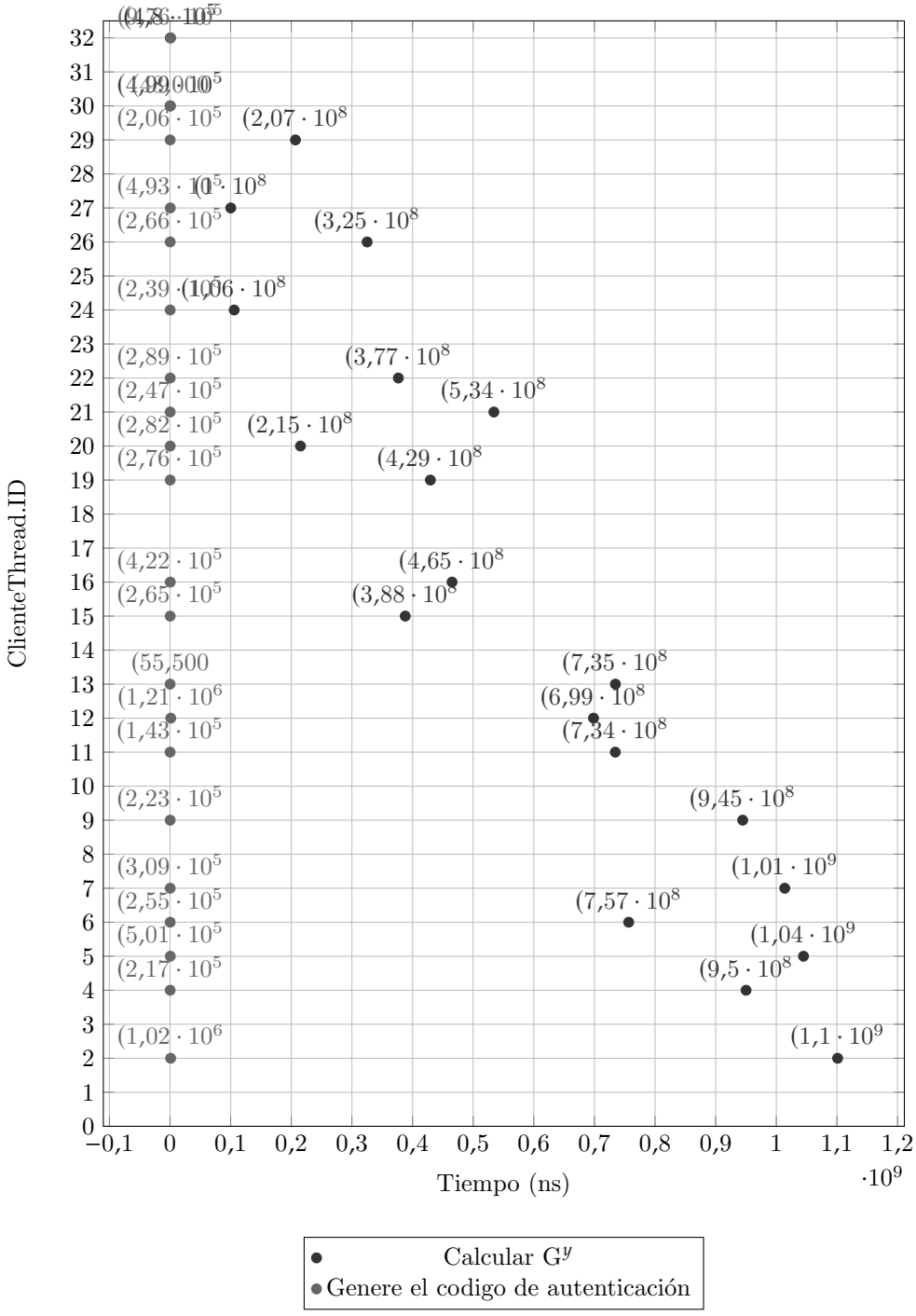
## 2.2. Caso 16


ClienteThread.ID
Tiempo (ns)
$\cdot 10^8$
0
0,2
0,4
0,6
0,8
1
1,2
1,4
1,6
1,8
2
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
$(1{,}78 \cdot 10^8)$
$(1{,}79 \cdot 10^8)$
$(1{,}82 \cdot 10^8)$
$(8{,}09 \cdot 10^7)$
$(8{,}04 \cdot 10^7)$
$(8{,}1 \cdot 10^7)$
$(8{,}16 \cdot 10^7)$
$(5{,}26 \cdot 10^7)$
$(2{,}8 \cdot 10^7)$
$(1{,}51 \cdot 10^7)$
$(4{,}54 \cdot 10^5$
$(1{,}03 \cdot 10^6$
$(7{,}57 \cdot 10^5$
$(2{,}26 \cdot 10^6$
$(2{,}44 \cdot 10^6$
$(1{,}52 \cdot 10^6$
$(3{,}9 \cdot 10^5$
$(2{,}15 \cdot 10^6$
$(2{,}77 \cdot 10^6$
$(6{,}08 \cdot 10^5$
$(2{,}48 \cdot 10^6$
$(1{,}46 \cdot 10^6$
$(2{,}19 \cdot 10^6$
Cifrar la consulta
Verificación de la firma


ClienteThread.ID
Tiempo (ns)
$\cdot 10^8$
$(7{,}83 \cdot 10^5)$
$(4{,}52 \cdot 10^5)$
$(1{,}31 \cdot 10^8)$
$(2{,}86 \cdot 10^5)$
$(6{,}52 \cdot 10^7)$
$(3{,}12 \cdot 10^5)$
$(1{,}52 \cdot 10^8)$
$(2{,}3 \cdot 10^5)$
$(1{,}71 \cdot 10^8)$
$(2{,}89 \cdot 10^5)$
$(3{,}32 \cdot 10^8)$
$(2{,}1 \cdot 10^5)$
$(3{,}93 \cdot 10^8)$
$(2{,}71 \cdot 10^5)$
$(2{,}1 \cdot 10^8)$
Calcular $G^y$
Genere el codigo de autenticación

## 2.3. Caso 32


ClienteThread.ID
Tiempo (ns)
$\cdot 10^8$
0
0,2
0,4
0,6
0,8
1
1,2
1,4
Cifrar la consulta
Verificación de la firma


ClienteThread.ID
Tiempo (ns)
$\cdot 10^9$
Calcular $G^y$
Genere el codigo de autenticación

# Capítulo 3

# Tablas

## 3.1. Caso 4

| ClienteThread.ID | Tiempo (ns) |
|---|---|
| 4.0 | 80430200 |
| **ClienteThread.ID** | **Tiempo (ns)** |
| 3.0 | 97381801 |
| **ClienteThread.ID** | **Tiempo (ns)** |
| 4.0 | 1648300 |
| **ClienteThread.ID** | **Tiempo (ns)** |
| 1.0 | 15661699 |
| 4.0 | 2478199 |
| 2.0 | 757300 |

## 3.2. Caso 16

| ClienteThread.ID | Tiempo (ns) |
|---|---|
| 12.0 | 28044301 |
| 15.0 | 7741099 |
| 11.0 | 80925599 |
| 9.0 | 80397000 |
| 1.0 | 182498500 |
| 3.0 | 81621900 |
| 13.0 | 177715101 |
| 4.0 | 178566100 |
| 8.0 | 80975700 |
| 16.0 | 561300 |
| **ClienteThread.ID** | **Tiempo (ns)** |
| 4.0 | 331737501 |
| 13.0 | 131101299 |
| 9.0 | 151765900 |
| 11.0 | 65237500 |
| 1.0 | 209752700 |
| 3.0 | 393328600 |
| 6.0 | 495800 |
| 12.0 | 467700 |
| 15.0 | 478200 |
| 16.0 | 783400 |

| ClienteThread.ID | Tiempo (ns) |
|---|---|
| 1.0 | 270701 |
| 4.0 | 288600 |
| 8.0 | 229901 |
| 11.0 | 285799 |
| 9.0 | 311801 |
| 3.0 | 210300 |
| 6.0 | 223000 |
| 12.0 | 321701 |
| 15.0 | 289901 |
| 16.0 | 215499 |
| **ClienteThread.ID** | **Tiempo (ns)** |
| 3.0 | 15093499 |
| 1.0 | 2476499 |
| 8.0 | 2440801 |
| 9.0 | 2264600 |
| 13.0 | 2190400 |
| 5.0 | 2147401 |
| 11.0 | 1026300 |
| 10.0 | 756600 |
| 2.0 | 607601 |
| 6.0 | 390000 |
| 12.0 | 453600 |
| 15.0 | 437201 |
| 14.0 | 1458300 |
| 16.0 | 639899 |
| 7.0 | 1523300 |

## 3.3. Caso 32

| ClienteThread.ID | Tiempo (ns) |
|---|---|
| 27.0 | 63221500 |
| 16.0 | 80112900 |
| 4.0 | 22078500 |
| 29.0 | 14681800 |
| 21.0 | 69579999 |
| 22.0 | 68658100 |
| 26.0 | 40884800 |
| 5.0 | 143260600 |
| 7.0 | 75995801 |
| 19.0 | 73940199 |
| 12.0 | 43101400 |
| 24.0 | 76316501 |
| 2.0 | 43763099 |
| 15.0 | 124421800 |
| 9.0 | 76712399 |
| 20.0 | 30678499 |
| 11.0 | 800001 |
| 6.0 | 1164800 |
| 13.0 | 7711300 |
| 30.0 | 557800 |

| ClienteThread.ID | Tiempo (ns) |
|---|---|
| 5.0 | 1044837000 |
| 15.0 | 387965600 |
| 16.0 | 465203300 |
| 22.0 | 376600900 |
| 19.0 | 429428700 |
| 12.0 | 698516801 |
| 27.0 | 100170501 |
| 29.0 | 206891299 |
| 24.0 | 105803000 |
| 2.0 | 1101102601 |
| 9.0 | 944558700 |
| 26.0 | 325129400 |
| 21.0 | 534206801 |
| 4.0 | 950098000 |
| 6.0 | 756500500 |
| 20.0 | 215163900 |
| 32.0 | 480301 |
| 11.0 | 734393300 |
| 13.0 | 734641000 |
| 30.0 | 499300 |
| **ClienteThread.ID** | **Tiempo (ns)** |
| 15.0 | 265099 |
| 16.0 | 421899 |
| 27.0 | 492599 |
| 7.0 | 309000 |
| 19.0 | 276100 |
| 24.0 | 239400 |
| 9.0 | 222901 |
| 22.0 | 289001 |
| 21.0 | 246700 |
| 12.0 | 1211500 |
| 2.0 | 1016700 |
| 26.0 | 266401 |
| 29.0 | 205601 |
| 4.0 | 216700 |
| 20.0 | 282400 |
| 32.0 | 976200 |
| 11.0 | 143400 |
| 6.0 | 255000 |
| 13.0 | 55600 |
| 30.0 | 48001 |

| ClienteThread.ID | Tiempo (ns) |
|---|---|
| 1.0 | 2661699 |
| 3.0 | 4095300 |
| 5.0 | 2431600 |
| 7.0 | 2327900 |
| 8.0 | 2135000 |
| 4.0 | 2165100 |
| 9.0 | 2144000 |
| 6.0 | 1064699 |
| 11.0 | 850000 |
| 13.0 | 710599 |
| 14.0 | 847900 |
| 12.0 | 866500 |
| 10.0 | 776900 |
| 17.0 | 773000 |
| 21.0 | 867901 |
| 16.0 | 887600 |
| 19.0 | 780301 |
| 15.0 | 768599 |
| 22.0 | 749900 |
| 26.0 | 751000 |
| 18.0 | 898001 |
| 20.0 | 1048900 |
| 23.0 | 781801 |
| 31.0 | 636200 |
| 29.0 | 687900 |
| 24.0 | 708700 |
| 27.0 | 1823600 |
| 25.0 | 648499 |
| 28.0 | 419001 |
| 32.0 | 416099 |
| 30.0 | 354101 |

# Capítulo 4

# Resultados

El proceso de verificado y generacion del codigo hmac es, sin duda, lo que menos tiempo requiere para ser resuelto. Por otra parte, los calculos tanto de $G^y$

# Capítulo 5

# Informacion

Velocidad de procesador: 2.60GHz

Ciclos Por Segundo: 2600 Millones

Promedio Tiempo Cifrado Segundos: $0.09*10^9$

Promedio Tiempo Verificacion Firma: $0.002*10^9$

Promedio Tiempo HMAC: $0.00025*10^9$

Cantidad de operaciones: CPS/Promedio

Cifrados: $\frac{260}{9}=28.\bar{8}$

Verificacion: 1300

HMAC: 10400