

# Criptografía Aplicada Básica Para Pentesters

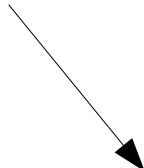
Jul 2014 v1.0

Ing. Aarón Mizrachi  
CISA | ITILv3F

# Que es la criptografía?

Técnicas para codificar los datos, destinadas a alterar el contenido para volverlo no legible a terceros.

Hola, este es un mensaje de texto...



```
000011D0  A0 3B 22 44 80 18 98 80 30 60 5B 80 E3 86 71 7F 77 B7 13 FD  ;"D....0 [...q.w...
000011E4  4F 45 4B 5C A3 96 3C 5A 48 1C 87 78 0C 4A D7 A5 D7 67 70 9C  0EK\...<ZH..x.J...gp.
000011F8  51 80 B6 80 8D 64 F6 CD 15 35 9C 41 B3 F2 0F 77 52 29 C5 90  Q....d...5.A...wR)..
0000120C  E7 51 8C C3 09 D7 EA 3A 90 1D 68 52 38 56 9B 60 D2 A9 80 63  .Q.....:..hR8V.~...c
00001220  7C DE 62 04 45 BB 04 D6 33 AB 7E 11 26 10 4D A1 F3 99 65 F6  |.b.E...3.~.&.M...e.
00001234  25 8A 2E 1D A8 95 96 EB 14 1B 2B 26 F8 93 D4 4F 00 66 E2 84  %.....+&...0.f..
00001248  DC 84 05 D1 B8 02 65 BC 57 E6 31 D9 37 BD 02 C0 80 00 87 EA  .....e.W.1.7.....
0000125C  EC C1 B9 2C 16 A4 F7 87 16 D9 F6 32 46 03 EE 8F E8 47 66 D0  ....,.....2F....Gf.
00001270  F4 33 89 B5 F4 32 E8 98 18 CD 24 16 B9 0E BE 5A E3 91 F7 3A  .3...2....$....Z...:
00001284  47 97 3F 24 02 B6 66 2D E5 37 D2 D8 D4 66 47 9A AB DB E2 A1  G.?$.f-.7...fG....
00001298  84 B5 03 DA AE 67 D2 49 E5 88 EE D6 64 D7 AB D7 85 16 85 E8  ....g.I....d.....
000012AC  C8 39 BD 48 55 9E 98 DF 47 5C 98 51 25 49 82 ED B7 49 1A 34  .9.HU...G\Q%I...I.4
000012C0  8F 22 52 CA 86 50 A5 1B 13 28 6D 99 59 62 39 33 FD 17 1E 0A  ."R..P...(m.Yb93....
000012D4  CA D3 B5 87 A5 B6 ED 2F D9 F6 E8 D6 8D EA B2 C1 F0 54 BF 5D  ...../.....T.]
--- cryptocontainer.img -- 0x12E7/0xA00000-----
```

# Objetivos

- **Confidencialidad:** Garantiza que solo las personas autorizadas puedan leer la información.
- **Integridad:** Garantiza que la información no haya sido alterada.
- **Autenticación:** Garantiza que el documento ha sido elaborado por quien el documento dice.
- **Vinculación o no repudio:** Se evita que el dueño del cifrado rechace a futuro su autoría.

# Como inicio todo?

- Se requería enviar mensajes de manera que si alguien lo interceptase, **no pudiese descifrarlo en un tiempo razonable.**
- Los primeros cifrados eran muy básicos, y con una estrategia sencilla se podían romper. Sin embargo, en su epoca cumplian su función:

**Mantener la información segura mientras fuese secreto.**

# Temporalidad del cifrado

- En épocas de guerra, era suficiente con que **el enemigo no supiese** de los planes del otro **hasta que dichos planes fuesen ejecutados**.
- Por ejemplo: Una orden de comando donde indica a un submarino moverse a nuevo rumbo. El cifrado es solo útil durante la duración de la operación.

# Es seguro?

- El cifrado **nunca es 100% seguro**. Su fortaleza siempre ha sido calculada en torno a la capacidad computacional de la época.
- La máquina enigma era considerado segura en su época. Hasta que lograron romperla.
- DES 56bit era considerado seguro en los años 90.
- RSA-1024 era considerado bastante seguro a principios del siglo XXI

# Cual es la diferencia entre codificador y cifrado?

Un **codificador** transforma el texto y es reversible mediante un decodificador sin clave.

Un **cifrado** requiere de una clave para cifrar el texto y una clave para descifrar. \*

# Codificaciones famosas

Las codificaciones se utilizan para transformar el texto en un formato distinto.

- **Datos Binarios:** Datos en el cual cada caracter ocupa cualquier posición desde 0 a 255.
- **Base 64:** Se utiliza para transformar datos binarios en un conjunto de datos representados por caracteres del alfabeto, más números, más +, / e =. Esto es muy utilizado en protocolos como SMTP.
- **Hexadecimal:** Como los datos binarios no son completamente representables en un monitor, muchos optan por mostrarlos en formato hexadecimal. Con dos caracteres del conjunto 0-9A-F, que representa su valor binario (00=0,FF=255)



# Operadores de cifrado Básicos

- XOR(A,B) (opera bit a bit), donde A y B son 1 bit. Si B es 1, el resultado es !A (el contrario de A), y si B es 0, entonces el resultado es A.

Por ejemplo:

- XOR(0,1) = 1

$$\text{XOR}(1,1) = 0$$

- XOR(0,0) = 0

$$\text{XOR}(1,0) = 1$$

# Operadores de cifrado Básicos

- SUM(A,B) (opera como una suma con limite),  
Por ejemplo, si la suma es de 8bit, el limite superior del resultado es 255, si la suma es de 16bit, el limite superior del resultado es 65535.

Por ejemplo (suma de 8 bit)

- $SUM8(255,1) = 0$
- $SUM8(254,3) = 1$

**A través de este operador se creó el cifrado CESAR. / ROT13.  
Es considerado INSEGURO por si mismo.**

# Operadores de cifrado Básicos

- S-BOX(A,B[]) (Substitución), donde B es una matriz que mapea cada valor posible de A a otro valor biunivoco.

La substitución puede ocurrir con matrices de n-bit. Por lo tanto, puede existir un S-BOX de 8bit, 16bit, etc... (incluso de 1bit)

# Operadores de cifrado Básicos

- P-BOX(A,B[]) (Caja de Permutación), La caja de permutación puede tener un tamaño variable, y puede referirse a permutar bit a bit, o a permutar conjuntos de bits.
  - Actúa intercambiando las posiciones, por ejemplo, una caja de permutación válida (B[]) para una palabra de 4 letras, puede ser: B[2,1,4,3], y entonces, si introducimos la palabra “hola”, el resultado será: “ohal”

# Tipos de cifrado

- **Cifrado de una sola vía** (texto -> **HASH**) \* (*este cifrado no admite clave para descifrarlo*) \*
- **Cifrado Simétrico** (texto+clave -> “texto cifrado”)
  - Para descifrar: “texto cifrado”+clave -> texto
- **Cifrado Asimétrico** (texto+clave publica -> “texto cifrado”)
  - Para descifrar: “texto cifrado”+”clave privada” -> texto
  - \* La llave pública no es capaz de hacer el proceso inverso.

# Hashes más populares

- **MD5 (128bit)**: Usado para almacenar claves, sin embargo es considerado inseguro para firmar documentos, ya que dos documentos podrían arrojar el mismo hash.
- **SHA-1(160bit)**: Se han planteado teorías donde se dice que SHA-1 es débil, pero a diferencia de MD5 no se conocen ataques prácticos.
- **SHA-2(224,256,512bit)**: Se recomienda por su fortaleza.

# El hash como medio para almacenar la clave

- Usualmente vemos que la clave se almacena en MD5 o SHA-1. Sin embargo, esto es inseguro.
- Existe gente que ha invertido tiempo y esfuerzo en construir “tablas rainbow” para estos algoritmos.
- Dichas tablas permiten traducir el hash en una clave.

# El hash como medio para almacenar la clave

- Solución: Salted Hash
- Que ocurre si hacemos la siguiente función:
  - MD5(HOLA)=c6f00988430dbc8e83a7bc7ab5256346
  - MD5(HOLAf3948hft), donde HOLA es el texto a codificar, y f3948hft un valor generado aleatoriamente que transmitiremos junto con la clave MD5. ej.
    - c1500bf02b5be8b0edd9f8ab7c317a99+f3948hft
  - Resultará que al ser aleatorio, no habrá nadie quien haya guardado esa combinación.
  - Pueden hacer google a ambos hashes.



Your search - **c1500bf02b5be8b0edd9f8ab7c317a99** - did not match any documents.

Suggestions:

- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.

---

About 105 results (0.36 seconds)

Tip: Search for **English** results only. You can specify your search language in [Preferences](#)

---

md5.znaet.org - **c6f00988430dbc8e83a7bc7ab5256346**

[md5.znaet.org/md5/c6f00988430dbc8e83a7bc7ab5256346](http://md5.znaet.org/md5/c6f00988430dbc8e83a7bc7ab5256346) ▾

Mar 9, 2014 - **c6f00988430dbc8e83a7bc7ab5256346** ... wrongkey: 34d877cd  
112e8d8a e0669914 64a429f5 md5: **c6f00988430dbc8e83a7bc7ab5256346** ...

HOLA хеш-код - **c6f00988430dbc8e83a7bc7ab5256346**

[md5.znaet.org/md5/c6f00988430dbc8e83a7bc7ab5256346](#)

# Tipos de cifrados Simétricos

- De Bloque: El más común y conocido, el cifrado de bloque. El algoritmo aplica sobre un bloque de n-bits (ej. AES256, bloques de 256bit).
- De Stream: Cifra bit a bit a medida que va saliendo. Es útil para transmitir información sobre medios con pérdida y ruido. Usualmente el cifrado por stream se usa mediante un CSPRNG (Generador de números random criptograficamente seguro). Sin embargo, también se puede usar cifrados de bloque para generar los siguientes bits

# Cifrados Simetricos Conocidos

- **Serpent** – 128bit, 256bit – El mas seguro, sin embargo, no fue seleccionado como AES ya que no era lo suficientemente rápido como rinjdael.
- **AES** (Rinjdael) – 128bit, 256bit – Suficientemente seguro, y acelerado por hardware.
- **Twofish (128bit)** – Tiene problemas en el caso de que se conozca el texto transmitido, se puede obtener la clave (con cierto esfuerzo).
- **Blowfish (64bit)** – Tiene problemas con claves débiles que hace que el tráfico no se vea con entropía.
- **3DES (3\*64bit)** – Utilizado por mayormente por Microsoft.
- **DES-56bit** – Utilizado antiguamente en internet. Su espacio de claves es reducido y se considera inseguro.
- **RC4** – Utilizado por WEP, es considerado inseguro.

# Cifrados Simétricos

- Los cifrados simétricos utilizan los operadores básicos de cifrado de forma concatenada.
- La disposición de dichos operadores y el número de iteraciones crean la seguridad del cifrado. Mientras mas iteraciones, el cifrado en teoría será mas robusto. Sin embargo, tambien tiende a ser mas lento.
- La clave (input) puede ser de 128bit o 256bit dependiendo de la especificación... sin embargo, la clave que introduce el usuario puede tener una cantidad menor de bits (entropía). Por lo tanto es indispensable expandirla a ese tamaño mediante una función HASH

# Cifrado Simétrico Por Bloque – Problemas.

- Que ocurre si cifras dos bloques iguales con la misma clave? El resultado será el mismo bloque cifrado. Si ciframos algo como una imagen, donde los patrones se repiten, dicha repetición de patrones hara que la imagen sea visible incluso después de cifrarla.
- Por lo tanto, se ha inventado una serie de mecanismos o modos de operación que permiten cifrar cada bloque de una manera distinta. (CBC, XTS, ECB, OFB, etc)

# Modos de Operación

- Para disco se recomienda utilizar **XTS**. Cifra dependiendo de la posición en el disco del bloque. Esto lo hace rápido y fácil.
- **CBC** usa vectores de inicialización y el bloque cifrado anterior para cifrar. Por lo tanto, en una lectura de disco sería mas lento. Sin embargo, se aplica mejor para ambientes de red.

# Cifrado Asimétrico (Teoría)

- El cifrado asimétrico se caracteriza por cifrar con una clave (pública) y descifrar con otra (privada).
- Por lo tanto, es costumbre publicar la llave pública.
- Conseguir la llave pública a partir de la privada es casi imposible.

# Algoritmos de cifrado asimétrico

- **RSA** – Se basa en números primos ( $P \cdot Q$ ), la llave privada es el primo  $P$ , y la pública es  $P \cdot Q$ , si  $P$  y  $Q$  son primos lo suficientemente grandes. Será imposible conseguir  $P$  y  $Q$  por separado. Sirve para **firmar y cifrar** documentos.
- **DSA** – Curvas elípticas – Sirve para **firmar** documentos. Se requieren claves mas pequeñas.
- **El Gamal** – Se basa en el problema del logaritmo discreto. Sirve para **firmar y cifrar** documentos.



# SSL/TLS

- Los algoritmos de cifrado son heterogéneos y no existía un consenso de como se debía cifrar información entre dos hosts en una red.
- El modelo de cifrado asimétrico es muy bueno, sin embargo, es poco práctico. Las claves de los cifrados asimétricos basados en RSA varían de 768bit a 4096bit (incluso mas si lo desean). Esto lo hace pesado para cifrar grandes cantidades de información.

# SSL/TLS

- El cifrado simétrico en la red por si mismo es inseguro ya que depende de una clave pre compartida. Si uno quisiera crear un sitio web para atender a una cantidad ilimitada de usuarios desconocidos, compartir una sola clave con todo el mundo sería lo mismo que no cifrar nada.
- La solución: **compartir la clave temporal del cifrado simétrico mediante un mecanismo asimétrico, y cifrar los datos de forma simétrica.**

# SSL/TLS

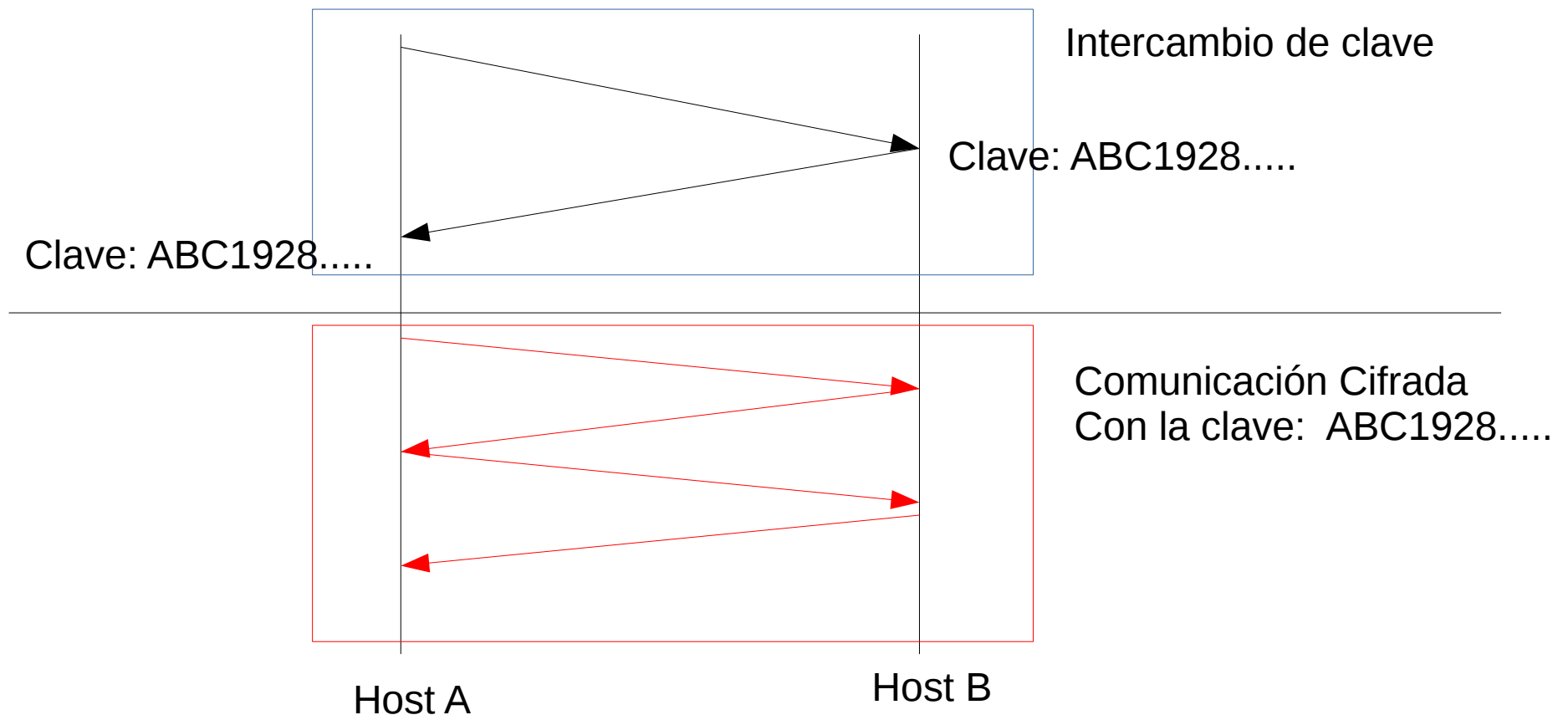
- SSL define una serie de protocolos para la transmisión de datos cifrados en canales TCP.
- TLS es un estandar mas nuevo, su versión actual, la 1.2, corrige muchas debilidades de las versiones anteriores.
- TLS Permite iniciar la conexión en la mitad de una conexión TCP
- TLS Permite conexiones UDP con DTLS

# SSL/TLS

- Este protocolo define:
  - Intercambio de claves
  - Firmas digitales
  - Cifrado de datos
- La conexión inicia desde el cliente solicitando una conexión cifrada con al menos ciertas características de seguridad.
- El servidor Responde con sus capacidades.
- Comienza el intercambio de claves.
- Comienza la conexión cifrada.

# Intercambio de claves

- **El problema:** Queremos que dos hosts *que no se conocen* se pongan de acuerdo en que clave compartida usar para cifrar sus datos:



# Intercambio de claves

- Como intercambiar la clave en público en un mundo donde todo el mundo escucha?



Como hacer para que a pesar de haber dicho la clave en público nadie se entere?

# Método 1 para intercambio de claves:

- **Algoritmo Diffie Hellman:** Este algoritmo permite que dos participantes se pongan de acuerdo en la clave. Todo ello en presencia de terceros que escuchan. Dichos **terceros no podrán lograr conocer cual fue la clave** en la que se pusieron de acuerdo.

# Diffie-Hellman (DH)

- Diffie-Hellman puede intercambiar claves de forma segura, pero a su vez **es vulnerable a ataques de hombre en el medio. Para evitarlo, se usan firmas digitales** que asegurarán que la transacción no ha sido modificada en el medio.
- El protocolo usado por SSL/TLS se llama DHE (Diffie-Hellman Efimero)



# Diffie-Hellman Efimero (DHE)

## PFS – Perfect Forward Secrecy

- Una vez termina la comunicación cifrada, ambos hosts deben borrar la clave.
- Mediante “Diffie-Hellman Efimero” **es imposible recuperar la clave de cifrado a futuro**, incluso teniendo la comunicación y estando el posesión de los equipos de transmisión.

# Método 2 para intercambio de claves:

- **Intercambio de claves vía RSA:** Al igual que DH, permite que los participantes se pongan de acuerdo en la clave de forma segura.
- A diferencia de DH, **no hay versión efímera.**

# Intercambio de claves RSA

P0: El cliente elige una clave temporal de cifrado: Ej. d9192e81928

P1: El cliente cifra dicha clave temporal con la llave pública del servidor

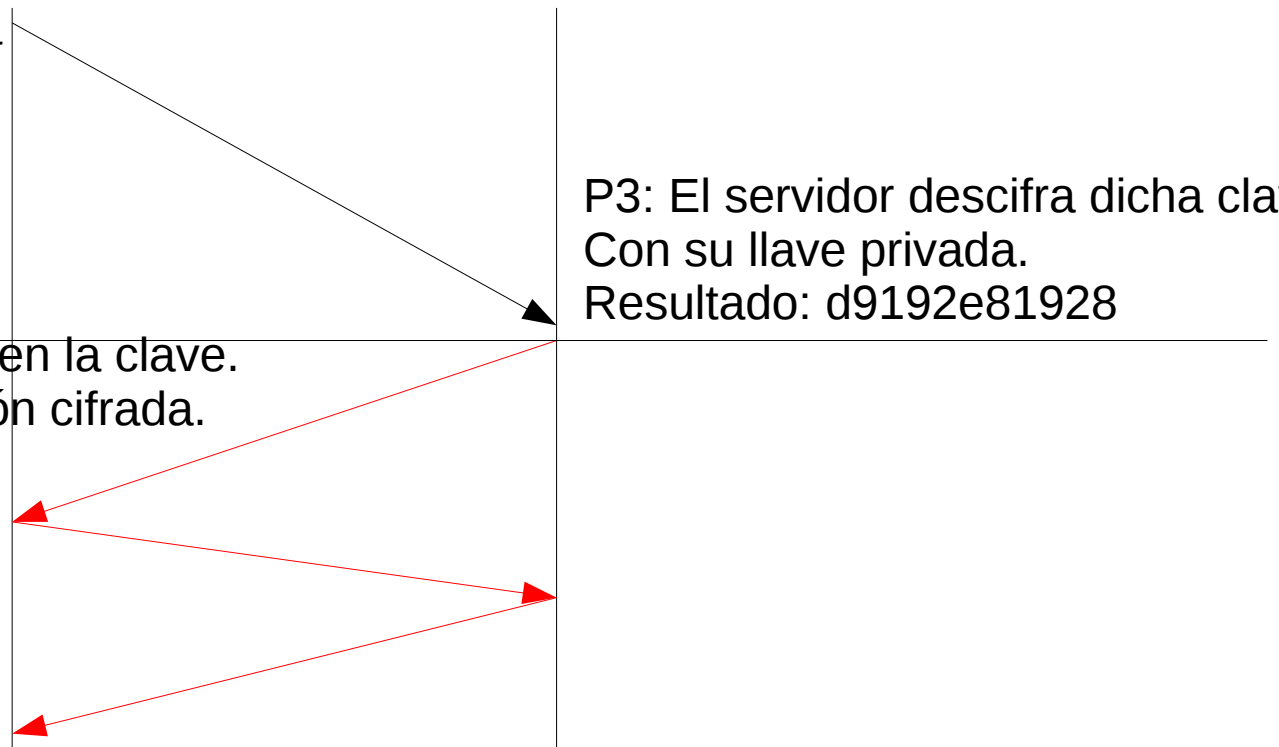
P2: Cliente envía  
Clave cifrada

P3: El servidor descifra dicha clave  
Con su llave privada.  
Resultado: d9192e81928

En este punto ambos conocen la clave.  
Puede iniciar la comunicación cifrada.

Host A  
Cliente

Host B  
Servidor



# Intercambio de claves RSA

Que ocurre si alguien filtrara la clave privada del servidor?

# Intercambio de claves RSA

- **RSA no es efímero**, por lo tanto, cualquiera que grabe la conversación, podrá con ayuda de la clave privada, descifrarla
- **Usos legítimos:**
  - **Legal:** Las autoridades podrían grabar las conversaciones y descifrarlas solo cuando sean autorizados para “escuchar”.
  - **N-IDS/N-IPS:** Estos descifran las conversaciones para analizar el tráfico y buscar ataques.
- **Usos ilegítimos:**
  - **Robo de Información:** Un atacante podría robar la clave privada y escuchar las conversaciones
  - **Espionaje:** Alguien podría recibir la clave privada y escuchar las conversaciones.

# Intercambio de claves DHE

Que ocurre si el servidor no es tan efimero y guarda las claves temporales.

Que ocurre si el servidor envia dichas claves temporales a un tercero en secreto?

# Intercambio de claves

El intercambio de claves **solo será seguro si confiamos en el host remoto.**