# Seguridad en Redes TCP/IP e Inalámbricas

Miguel Angel Astor Romero

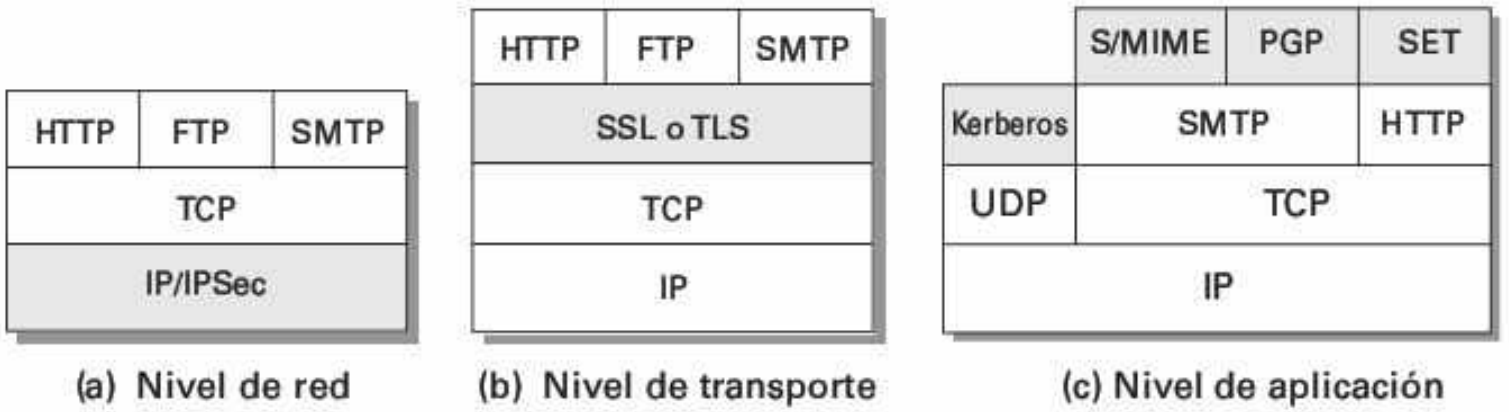
26 de julio de 2019

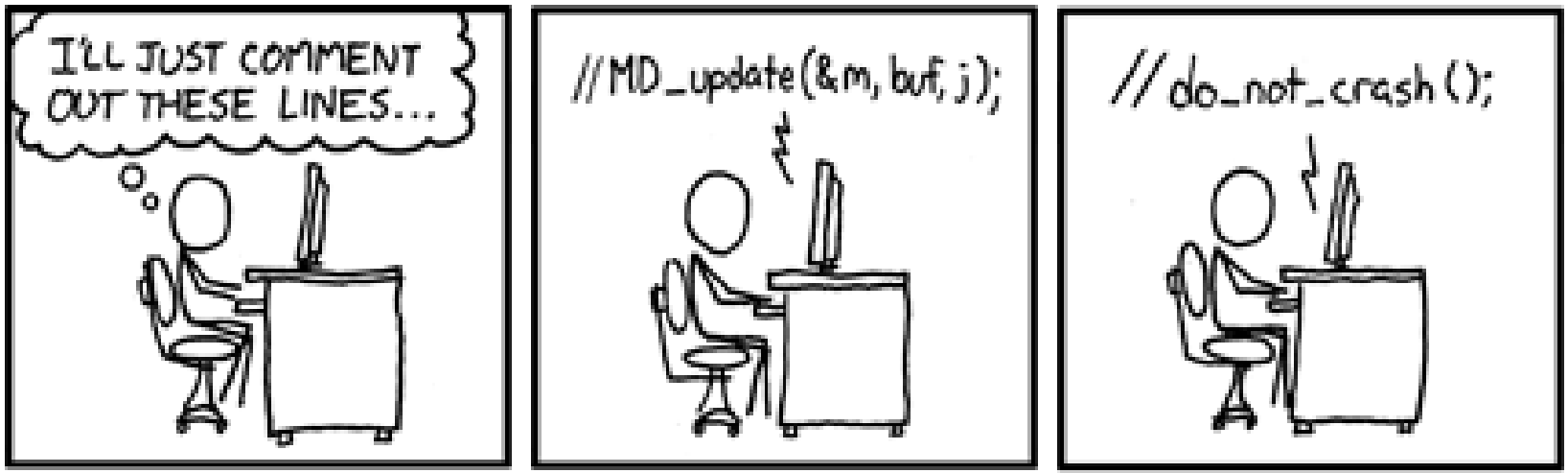
# Agenda

# Amenazas a la Seguridad en la Web

| | Amenazas | Contramedidas |
|---|---|---|
| Integridad | Modificación de datos, memoria o mensajes. | Sumas de verificación. |
| Confidencialidad | Escucha y/o robo de información o parámetros de red. | Criptografía. |
| Disponibilidad | Interrupción, o caida del servicio. | Múltiples. |
| Autenticación | Suplantación de identidad. Falsificación de datos. | PKI. |

# Herramientas de Seguridad en la Pila TCP/IP




I'LL JUST COMMENT OUT THESE LINES...
//MD_update(&m, buf, j);
//do_not_crash();

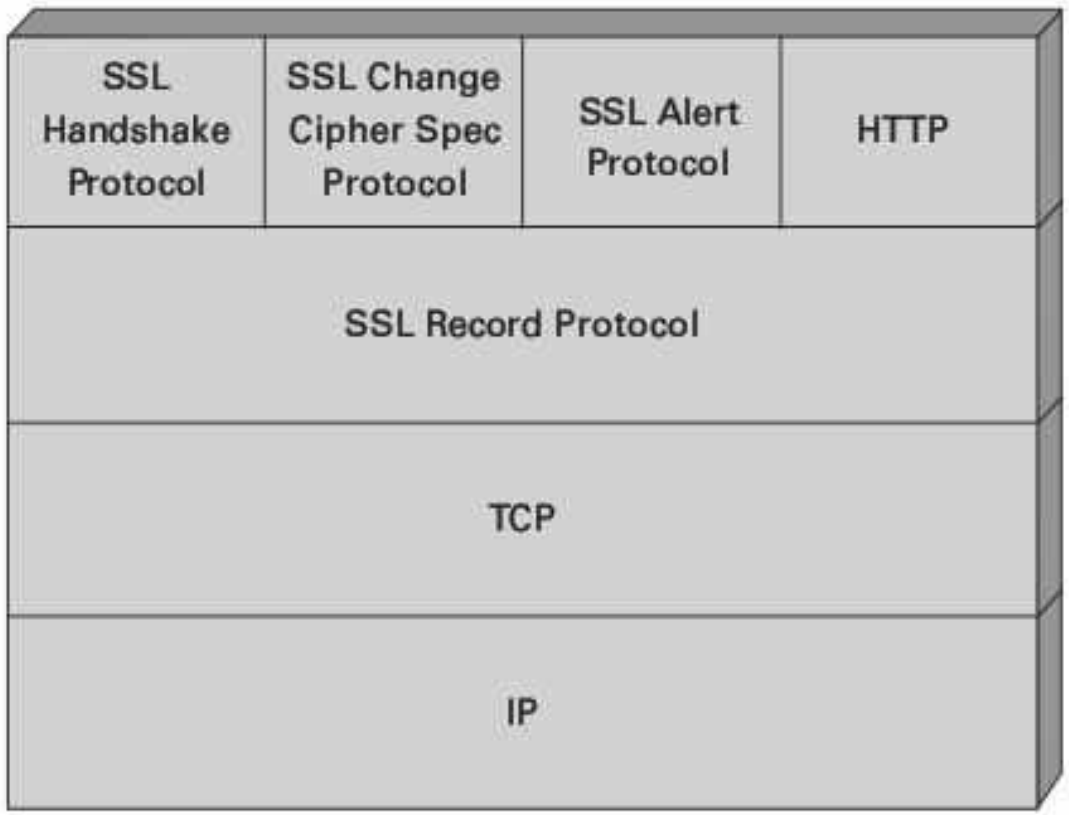
# Historia de SSL - Secure Sockets Layer

Taher Elgamal.

- Inventado por Taher Elgamal, Paul Kocher, Phil Karlton y Alan Freir en Netscape Communications
- Versiones:
  - SSL 1.0 No publicada.
  - SSL 2.0 Publicada en 1995.
  - SSL 3.0 Publicada en 1996.
- Adoptado por la IETF en 1999 y renombrado como *Transport Layer Security*.
- Versión más reciente en el RFC 8446.

# El Otro Logro de Netscape


# Pila de Protocolos SSL



# Conceptos

## Conexión

- Protocolo de transporte.
- Relación Par-a-Par.
- Transitoria.
- Asociada a una sesión.

# Conceptos

## Conexión

- Protocolo de transporte.
- Relación Par-a-Par.
- Transitoria.
- Asociada a una sesión.

## Sesión

- Asociación cliente-servidor.
- Creadas por el protocolo *Handshake*.
- Definen los parámetros criptográficos reutilizables entre conexiones.

# Conceptos

## Conexión

- Protocolo de transporte.
- Relación Par-a-Par.
- Transitoria.
- Asociada a una sesión.

## Sesión

- Asociación cliente-servidor.
- Creadas por el protocolo *Handshake*.
- Definen los parámetros criptográficos reutilizables entre conexiones.

## Las sesiones poseen varios estados

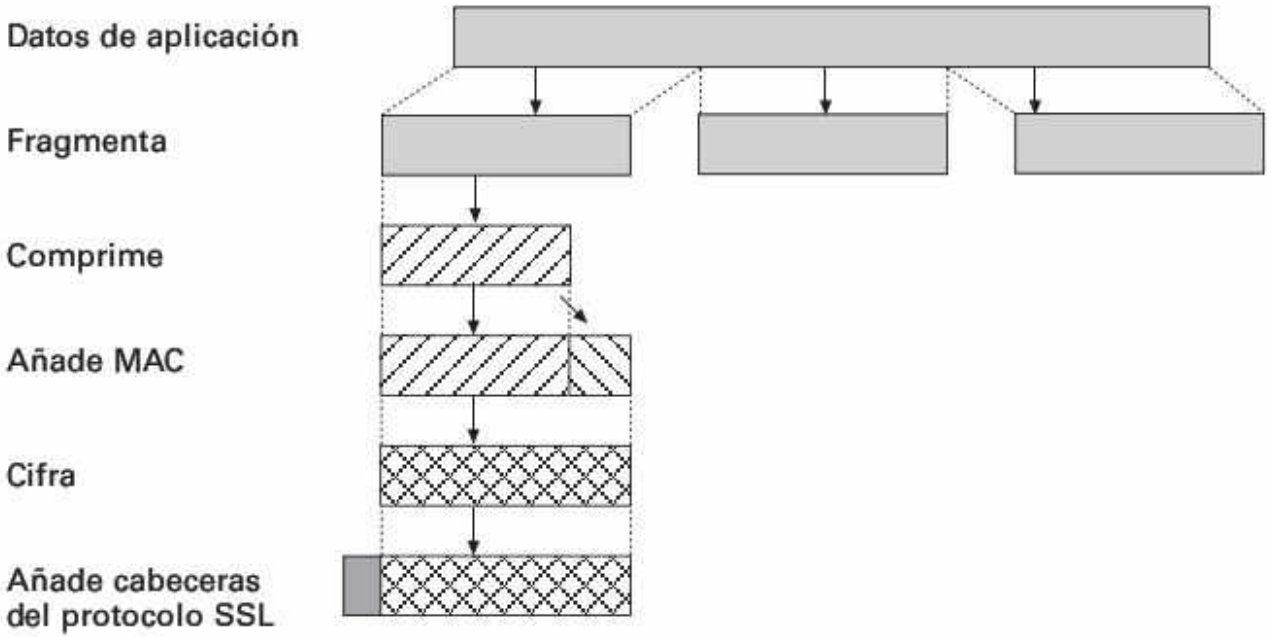
**Operativo** parámteros aptos para enviar o recibir.

**Pendiente** parámetros en espera de pasar a operativos.

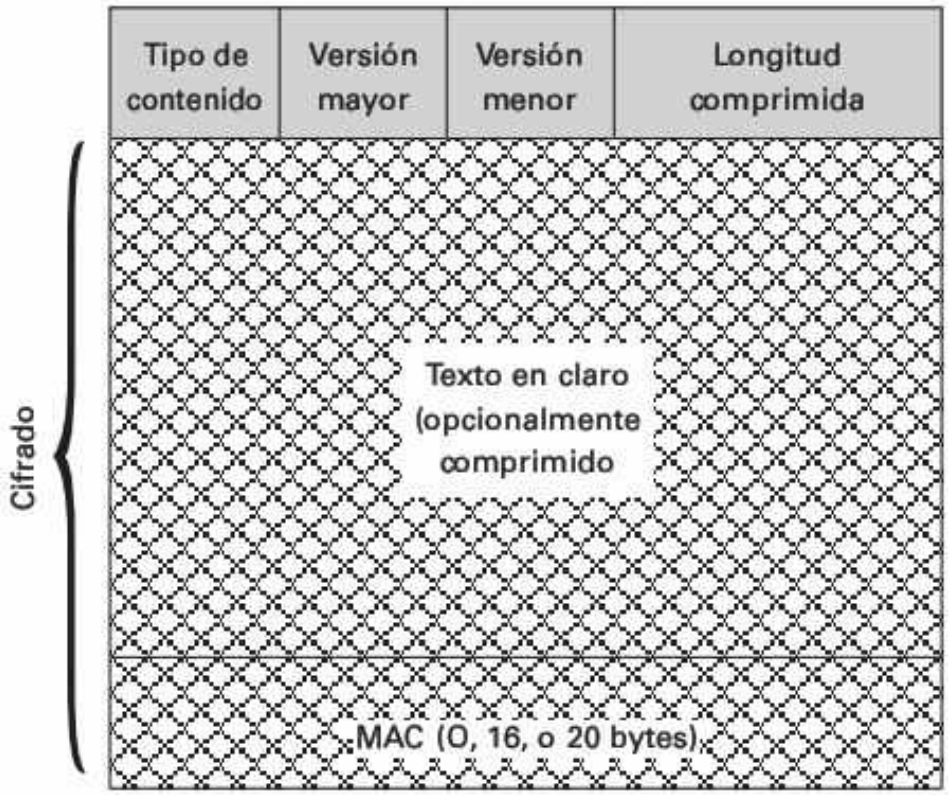
# Protocolo Record

- Protocolo de transporte base de TLS.
- Provee:
  - Confidencialidad.
  - Integridad de datos.
  - Fragmentación.
  - Compresión.
- Se usa para transmitir datos de aplicación o de otros protocolos de TLS.

# Operación del Protocolo Record



# Formato del Mensaje del Protocolo Record



# Protocolo Change Cipher Spec

- Se utiliza para cambiar los parámetros de cifrado durante el *handshake*.
- Consiste en un único byte que siempre tiene el valor 1 (uno).


(a) Protocolo *Change Cipher Spec*

# Protocolo Alert

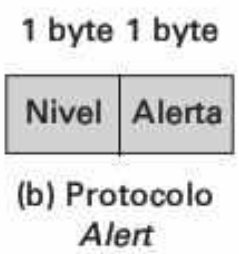
- Se usa para transmitir alertas a la entidad par.
  - Aviso notificación general o de error no fatal.
  - Fatal cancela la sesión inmediatamente.

## Alertas Fatales

- Mensaje inesperado.
- MAC erroneo.
- Falla de descompresión.
- Falla de negociación.

## Avisos

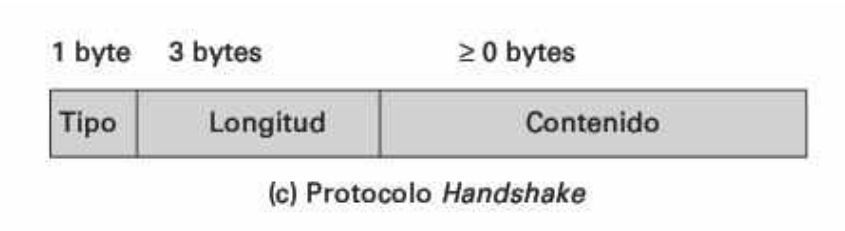
- Notificación de cierre.
- Errores de certificado:
  - No hay certificado.
  - Certificado erroneo.
  - Certificado revocado.




(b) Protocolo *Alert*

# Protocolo Handshake

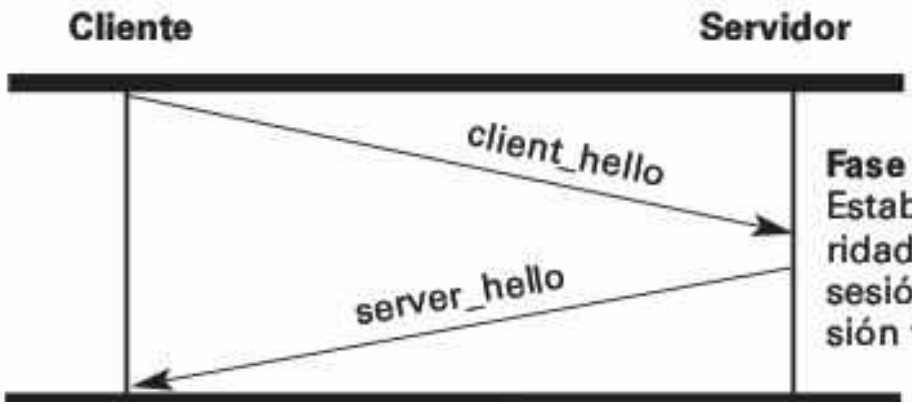
- Toda sesión SSL comienza por un handshake.
- Permite autenticación unidireccional o mutua.
- Realiza la negociación de los parámetros de sesión.




(c) Protocolo *Handshake*

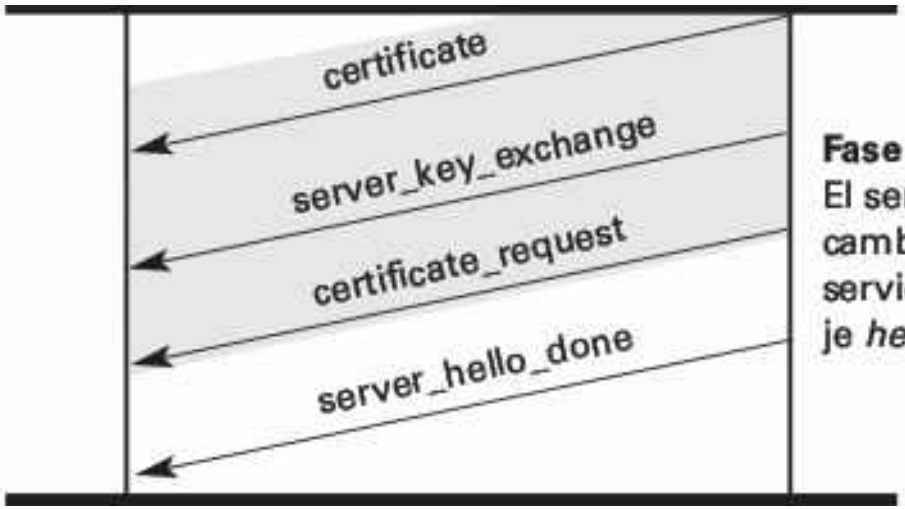
# Protocolo Handshake - Fase 1 - Saludo Inicial

- Establece la versión del protocolo, ID de sesión, suite de cifrado, algoritmo de compresión y números aleatorios iniciales.



# Protocolo Handshake - Fase 2 - Autenticación del Servidor

- Se realiza el intercambio de claves.
- La autenticación del servidor es opcional.




**Fase 2**
El servidor puede enviar un certificado, intercambio de clave y solicitud de certificado. El servidor señala el final de la fase del mensaje *hello*.

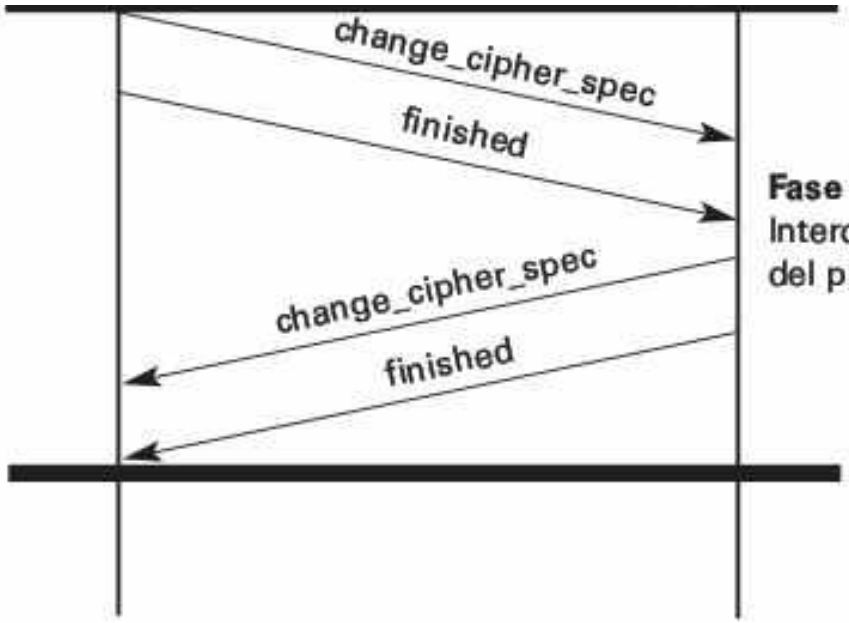
# Protocolo Handshake - Fase 3 - Autenticación del Cliente

- Completamente opcional.


**Fase 3**
El cliente envía certificado, si se le solicita, el intercambio de clave, y puede que envíe la verificación de certificado.

# Protocolo Handshake - Fase 4 - Intercambio de Suite de Cifrado

- Intercambio de suite de cifrado y establecimiento de la sesión.




**Fase 4**
Intercambio de *suite* de cifrado y finalización del protocolo *Handshake*.

# Creación de la Clave Maestra Compartida

- Se comparte un pre-secreto mediante uno de los siguientes algoritmos:

  RSA si se realizó autenticación con certificados.

  Diffie-Hellman fijo, efímero o anónimo.

  Fortezza para uso con tarjetas inteligentes.

$$
\begin{aligned}
master\_secret = \quad & MD5(pre\_master\_secret \| SHA('A' \| pre\_master\_secret \| \\
& ClientHello.random \| ServerHello.random)) \| \\
& MD5((pre\_master\_secret \| SHA('BB' \| pre\_master\_secret \| \\
& ClientHello.random \| ServerHello.random)) \| \\
& MD5(pre\_master\_secret \| SHA('CCC' \| pre\_master\_secret \| \\
& ClientHello.random \| ServerHello.random))
\end{aligned}
$$

# Tarjetas Fortezza


Crypto Museum
cryptomuseum.com

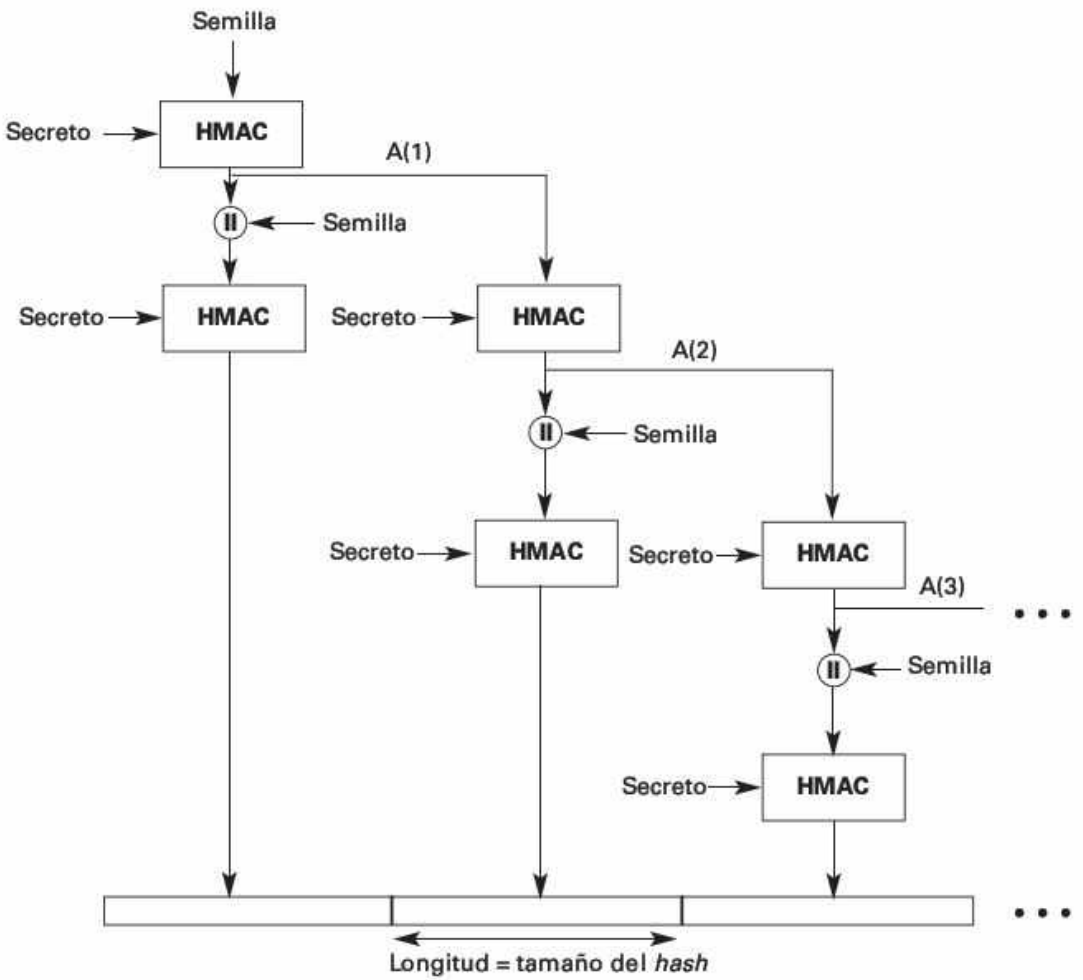
# Generación de Parámetros Criptográficos

- Después de la generación del secreto maestro se deben generar 6 parámetros, en este orden:

  Client write MAC secret clave HMAC del cliente.
  Server write MAC secret clave HMAC del servidor.
  Client write key clave secreta para el cliente.
  Server write key clave secreta para el servidor.
  Client IV vector de inicialización del cliente.
  Server IV vector de inicialización del servidor.

- Estos parámetros se generan con un generador de números aleatorios llamado PRF.

# Función Pseudoaleatoria PRF



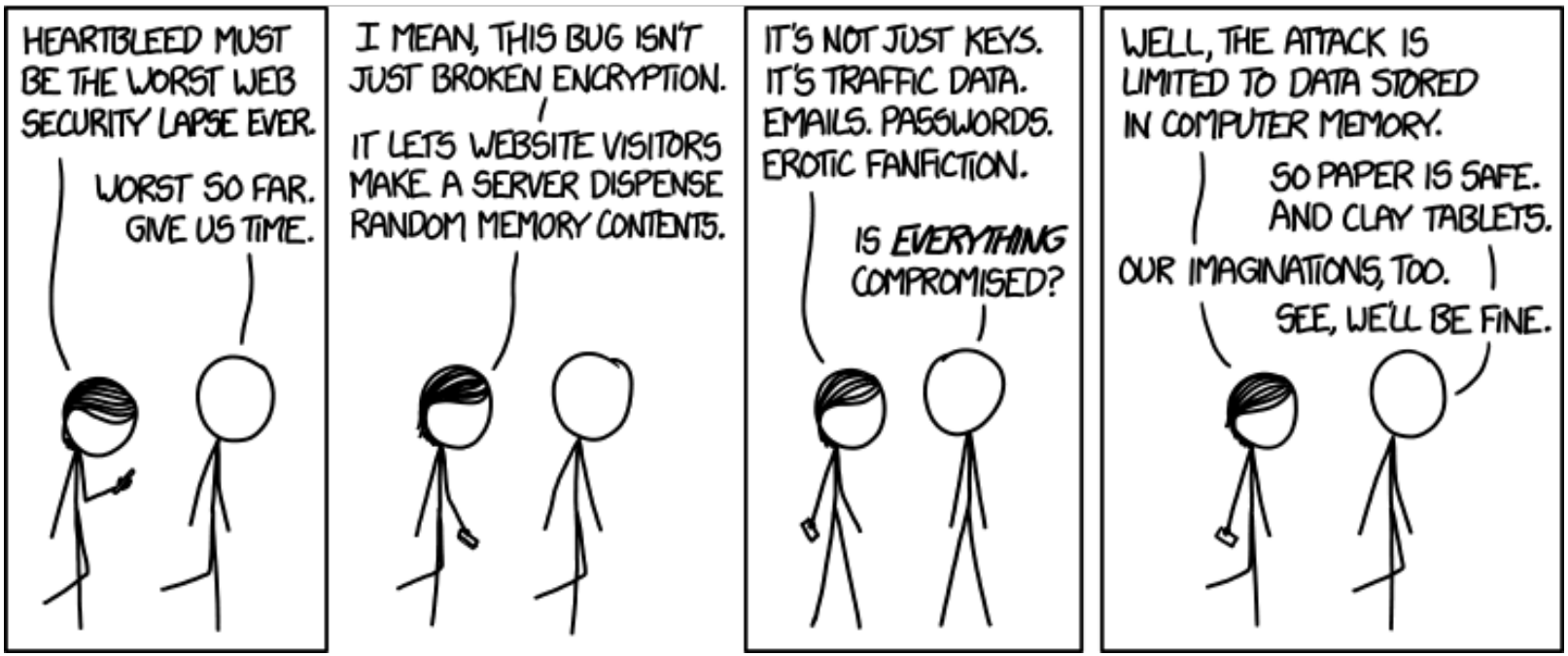
# HTTPS

- Extensión de HTTP que utiliza SSL/TLS como transporte.
- Soporta autenticación unidireccional o mutua.

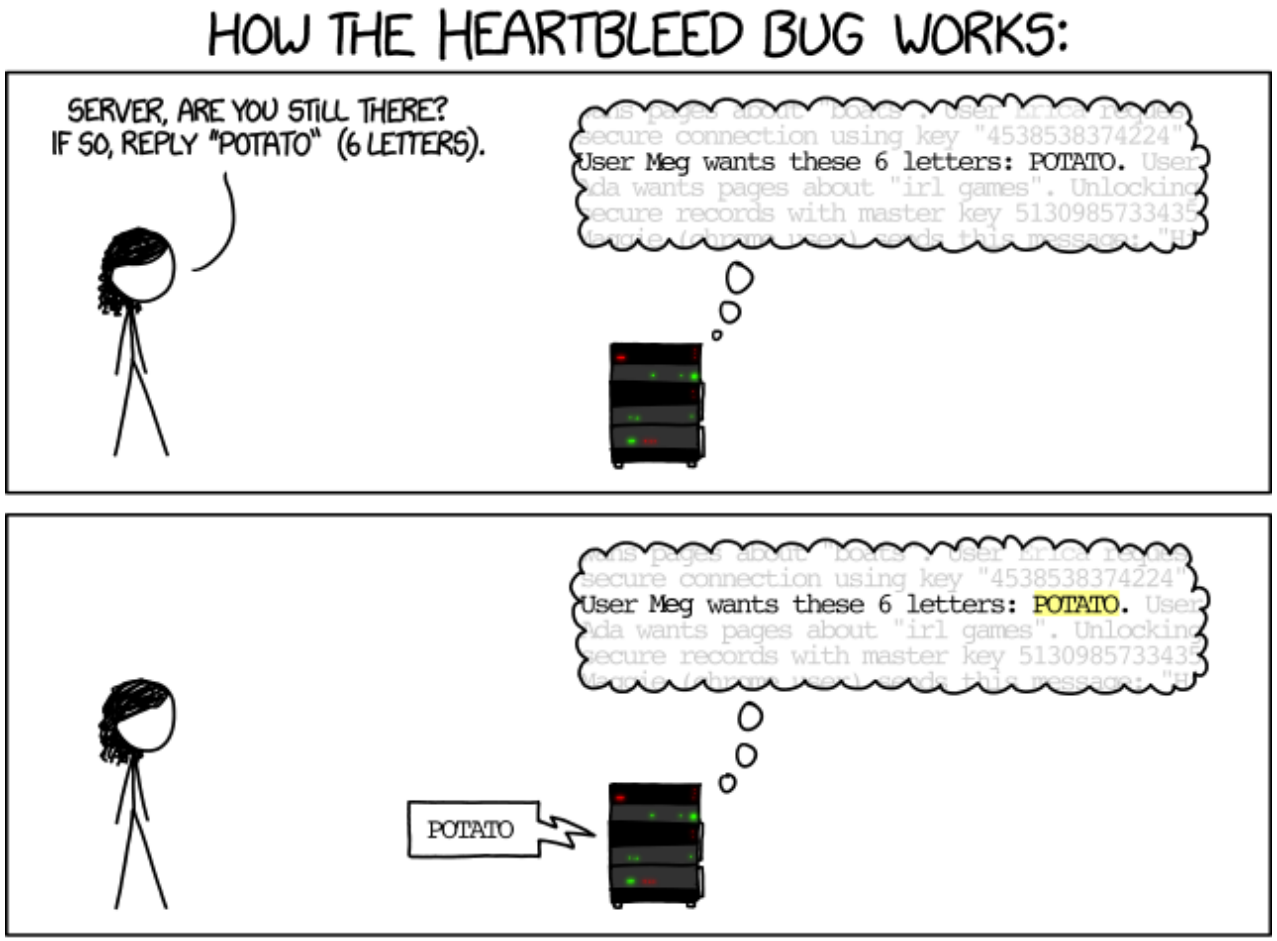
# Extensión Heartbeat de TLS

- Definida en el RFC 6520: Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension.
- Especificación de un protocolo *keep-alive* opcional.
- El protocolo es básicamente un protocolo *echo*.
- Fue susceptible a una falla severa descubierta en el año 2014.

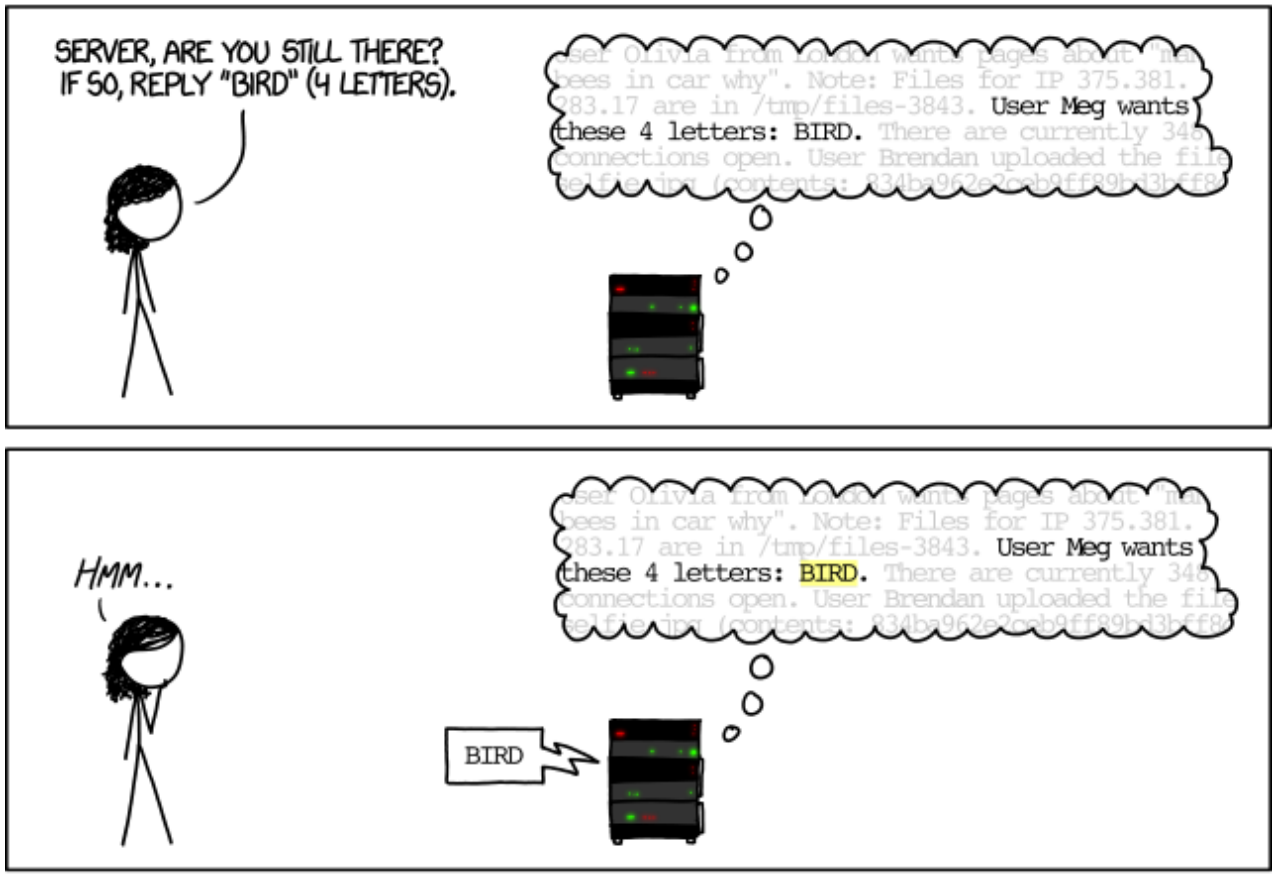
# Heartbleed


HEARTBLEED MUST BE THE WORST WEB SECURITY LAPSE EVER.
WORST SO FAR. GIVE US TIME.
I MEAN, THIS BUG ISN'T JUST BROKEN ENCRYPTION.
IT LETS WEBSITE VISITORS MAKE A SERVER DISPENSE RANDOM MEMORY CONTENTS.
IT'S NOT JUST KEYS. IT'S TRAFFIC DATA. EMAILS. PASSWORDS. EROTIC FANFICTION.
IS EVERYTHING COMPROMISED?
WELL, THE ATTACK IS LIMITED TO DATA STORED IN COMPUTER MEMORY.
SO PAPER IS SAFE. AND CLAY TABLETS.
OUR IMAGINATIONS, TOO.
SEE, WE'LL BE FINE.

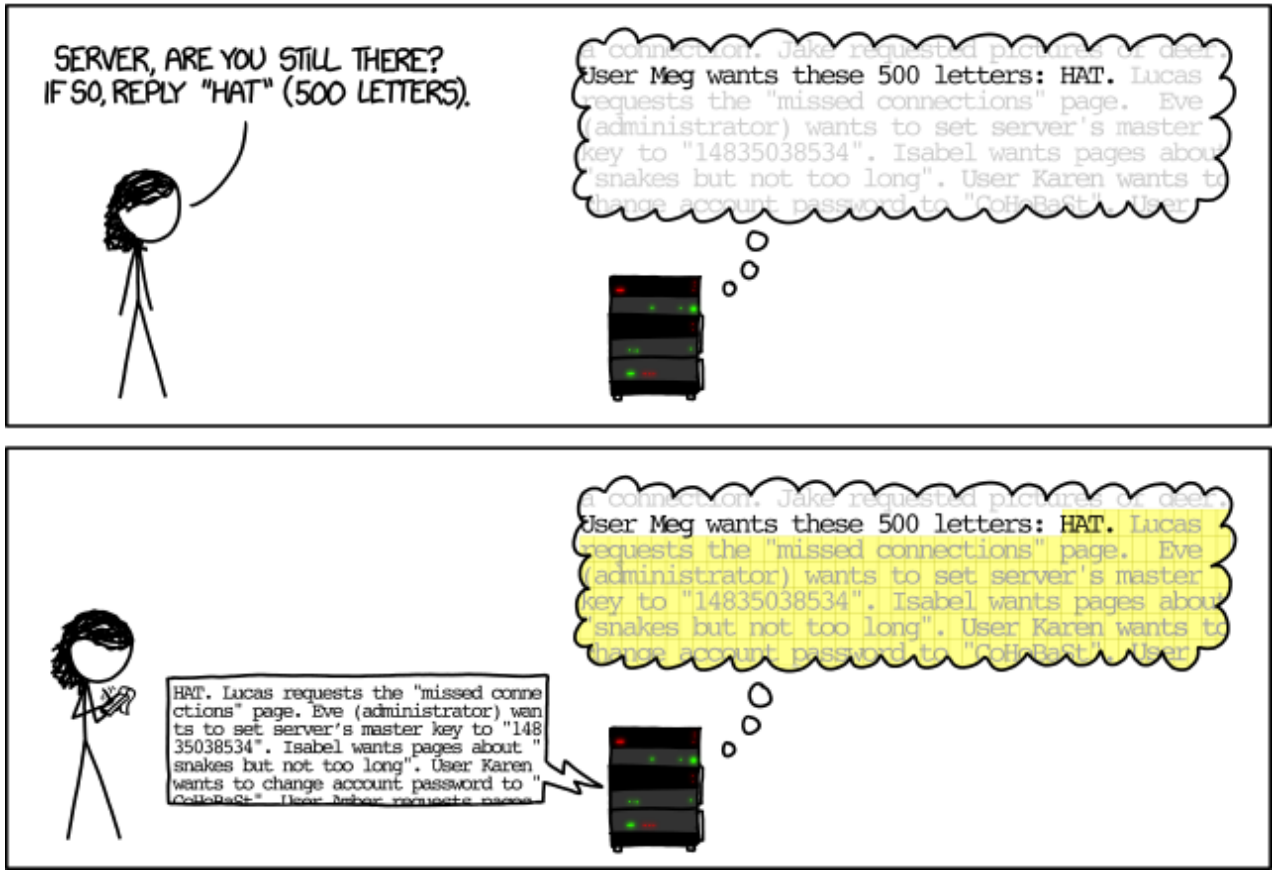
# Como Funciona Heartbleed - 1/3



# Como Funciona Heartbleed - 2/3



# Como Funciona Heartbleed - 3/3



# Goto Fail

Error descubierto en el 2014 en la implementación de TLS de Apple.

```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                 uint8_t *signature, UInt16 signatureLen)
{
  OSStatus        err;
  ...

  if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
  if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;
  if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
    goto fail;
  ...

fail:
  SSLFreeBuffer(&signedHashes);
  SSLFreeBuffer(&hashCtx);
  return err;
}
```

# Implementaciones

## OpenSSL


- En desarrollo desde el 2008.
- Usada en distros de Linux.

# Implementaciones

## OpenSSL

OpenSSL
Cryptography and SSL/TLS Toolkit

- En desarrollo desde el 2008.
- Usada en distros de Linux.

## LibreSSL

- *Fork* de OpenSSL.
- Usada en los sistemas BSD.

# Implementaciones

## OpenSSL

OpenSSL
Cryptography and SSL/TLS Toolkit

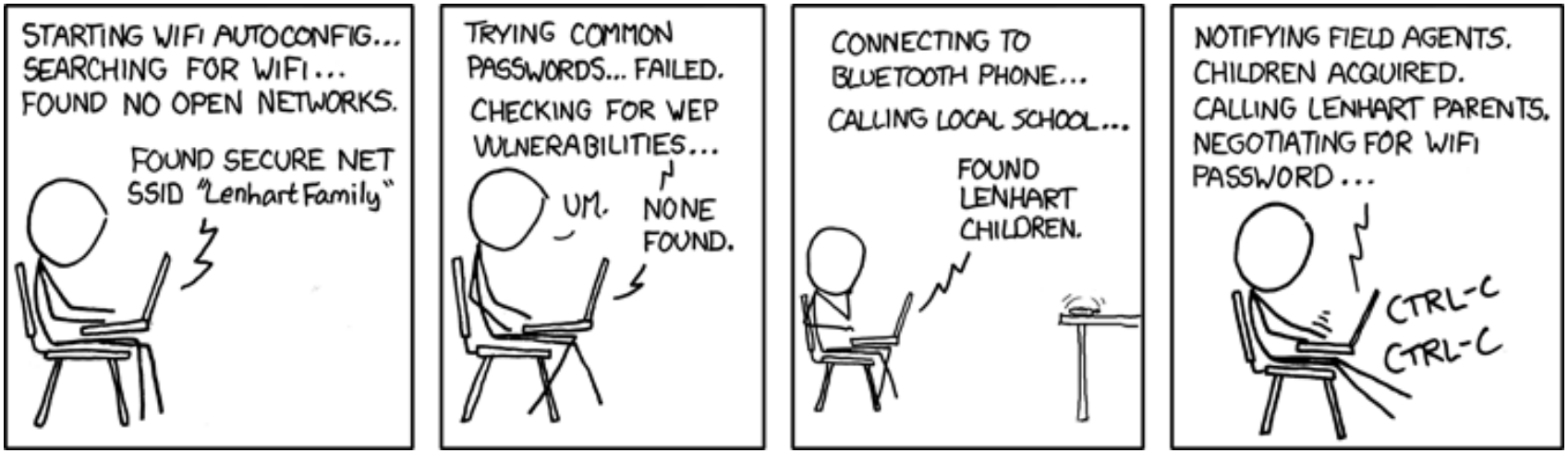
- En desarrollo desde el 2008.
- Usada en distros de Linux.

## LibreSSL

- *Fork* de OpenSSL.
- Usada en los sistemas BSD.

## Otras implementaciones

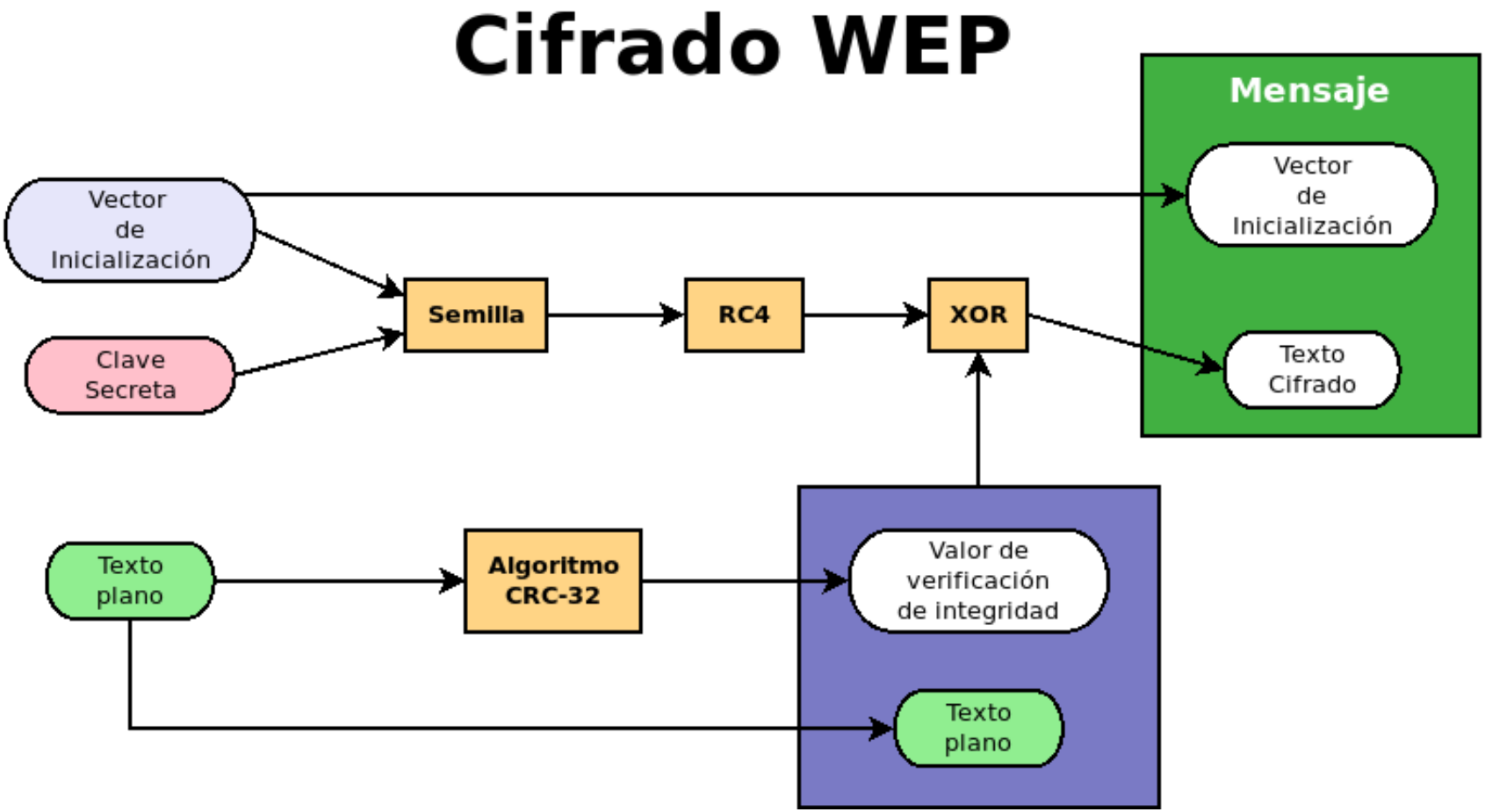
- Apple's SSL en iOS.
- BoringSSL y Tink de Goole. *Forks* de OpenSSL.


STARTING WIFI AUTOCONFIG...
SEARCHING FOR WIFI...
FOUND NO OPEN NETWORKS.
FOUND SECURE NET SSID "Lenhart Family"
TRYING COMMON PASSWORDS... FAILED.
CHECKING FOR WEP VULNERABILITIES...
UM.
NONE FOUND.
CONNECTING TO BLUETOOTH PHONE...
CALLING LOCAL SCHOOL...
FOUND LENHART CHILDREN.
NOTIFYING FIELD AGENTS.
CHILDREN ACQUIRED.
CALLING LENHART PARENTS.
NEGOTIATING FOR WIFI PASSWORD...
CTRL-C
CTRL-C

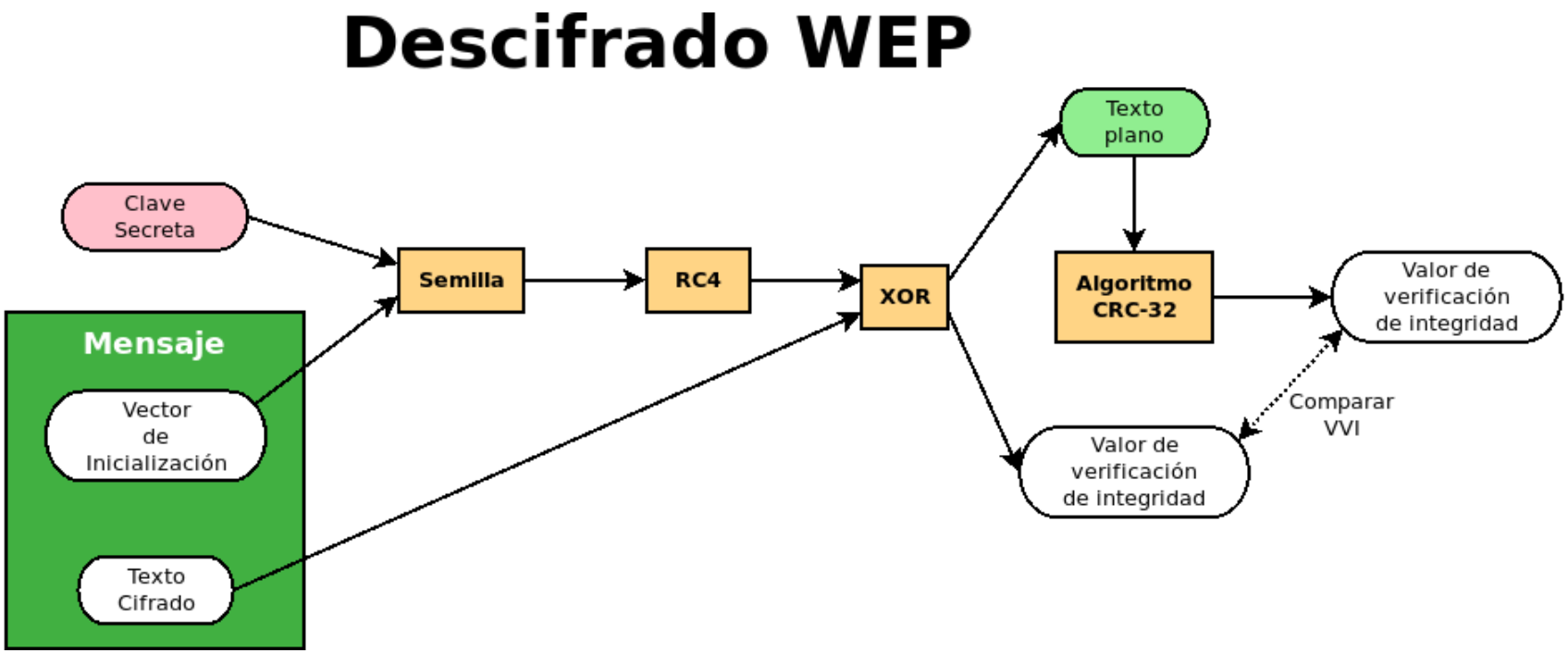
# WEP

- Siglas de *Wired Equivalent Privacy*.
- Primer protocolo de cifrado y privacidad usado en IEEE 802.11.
- Es muy facil de romper debido a sus malas consideraciones de diseño.
- Cifrado de flujo basado en el algoritmo RC4.

# Cifrado


Cifrado WEP
Mensaje
Vector de Inicialización
Clave Secreta
Semilla
RC4
XOR
Vector de Inicialización
Texto Cifrado
Texto plano
Algoritmo CRC-32
Valor de verificación de integridad
Texto plano

# Descifrado


Descifrado WEP
Clave Secreta
Semilla
RC4
XOR
Texto plano
Algoritmo CRC-32
Valor de verificación de integridad
Comparar VVI
Valor de verificación de integridad
Mensaje
Vector de Inicialización
Texto Cifrado

# ¿Por que WEP es Inseguro?

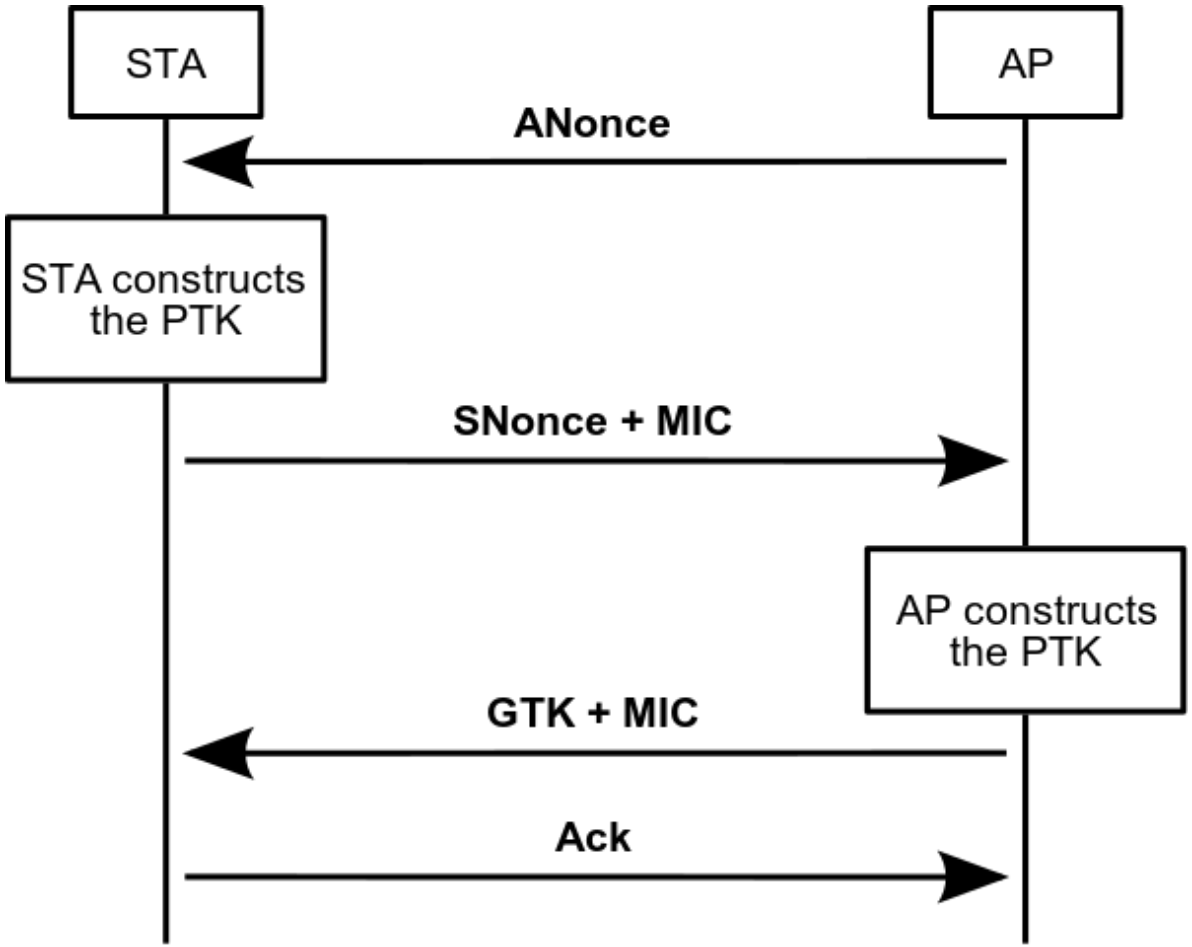
WEP sufre de dos grandes problemas:

1. La contraseña es de tamaño fijo y se reutiliza constantemente.
2. El vector de inicialización se transmite en claro.
   - Esto mitiga todas las ventajas de cambiar el VI constantemente.

# WPA

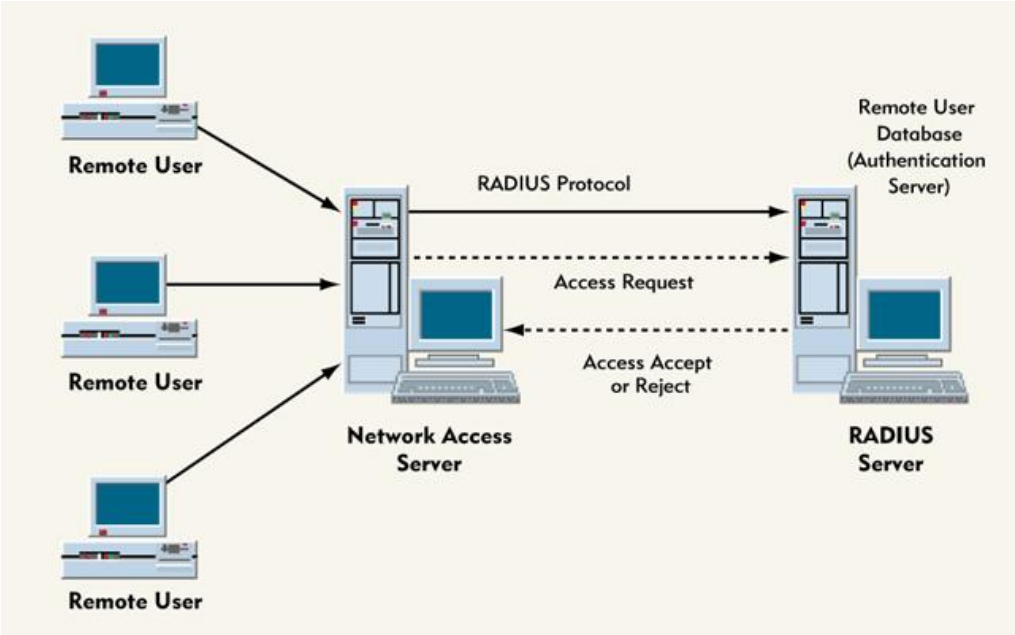
- Siglas de *Wi-Fi Protected Access*.
- Actualmente en su versión 2.
- La versión 3 fue estandarizada en el 2018.
- La versión 2 puede funcionar en dos modos:
  - PSK utiliza una clave compartida.
  - EAP utiliza un servidor de autenticación.

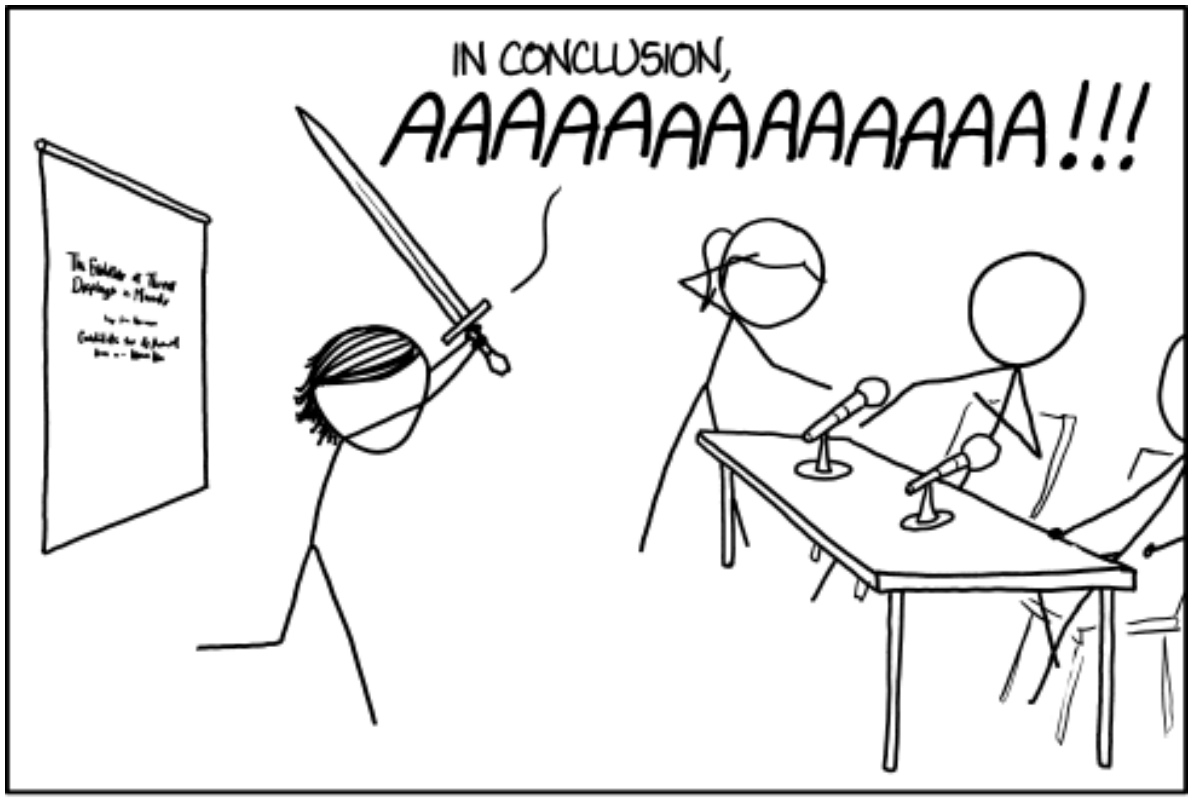
# Funcionamiento de WPA2/PSK



# RADIUS

- Servicio de autenticación, autorización y contabilidad.
- Diseñado en 1991.
- Estándar de Internet gestionado por la IETF.
- Se utiliza como servidor de autenticación en WPA2/EAP.




IN CONCLUSION,
AAAAAAAAAAAAA!!!
THE BEST THESIS DEFENSE IS A GOOD THESIS OFFENSE.

# Conclusiones

- SSL/TLS es la base de la seguridad en la Web.
- La tendencia es a utilizar TLS para toda comunicación en la Web, mediante el protocolo HTTPS.
  - Incluyendo servicios estáticos sobre HTTP.
- WEP nos muestra porque hay que tener mucho cuidado al diseñar mecanismos de seguridad.

# Próxima Clase

- Amenazas y Ataques:
  - Categorías de Ataque.
  - Malware y Tipos de Malware.
  - *Firewalls*.
  - Sistemas de Detección de Intrusos (IDS).
  - IDS basados en minería de datos.

# ¿Preguntas?