Seguridad en Redes de Computadora

Principios de la seguridad en redes

- **Confidencialidad**
  - Sólo el emisor y el receptor deben poder entender el contenido de los mensajes intercambiados. Para lograr esto el emisor encripta mensajes y el receptor los desencripta.
- **Integridad**
  - Tanto el emisor como el receptor desean garantizar que los mensajes intercambiados no fueron manipulados en tránsito.
- **Autenticación**
  - Tanto el emisor como el receptor buscan confirmar la identidad de su interlocutor.
- **Disponibilidad**
  - Los servicios deben ser accesibles y estar disponibles para los usuarios.

Actores maliciosos

Acciones maliciosas que pueden llevarse acabo:

1. Interceptar mensajes (sniffing)
2. Inyectar mensajes en la conexión
3. Impersonar otros actores (e.g. spoofing de direcciones en los paquetes)

4. Tomar control de la conexión eliminando al emisor o al receptor y adoptando dicho rol (hijacking)
5. Denegar servicios, prohibiendo que otros usuarios legítimos utilicen el servicio afectado.

Criptografía

Terminología

- m : mensaje de texto plano
- $K_A(m)$: texto cifrado(encriptado con la clave K_A)
-

Tipos de ataques

1. Ataque de texto cifrado:
    a. Supone que el atacante puede obtener y analizar textos cifrados.
    b. Existen dos enfoques
        i. Fuerza bruta: explorar **todo** el espacio de claves.
        ii. Análisis estadístico
2. Ataque de texto plano conocido:
    a. Supone que el atacante dispone del texto plano correspondiente a cierto texto cifrado.
3. Ataque de texto plano elegido:
    a. Supone que el atacante puede obtener el texto cifrado de un texto plano a elección (oráculo de encripción)

Criptografía simétrica: DES

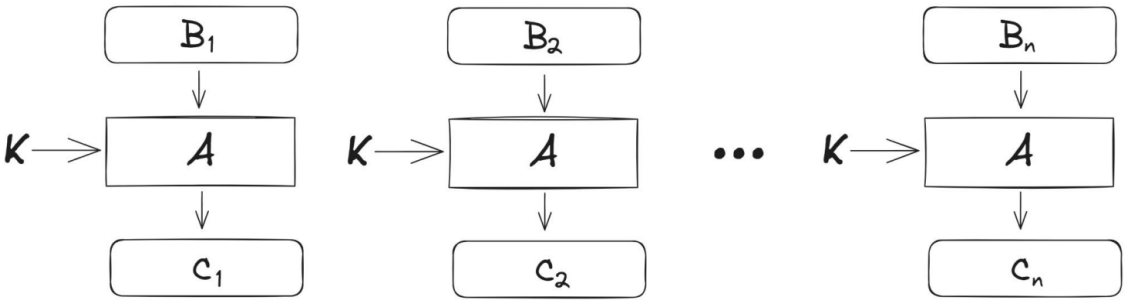
DES: Data Encryption Standard

Clave simétrica de 56 bits; toma textos planos de 64 bits. Además, cifra a partir de bloques (block cipher).

Block ciphers: modo de operación

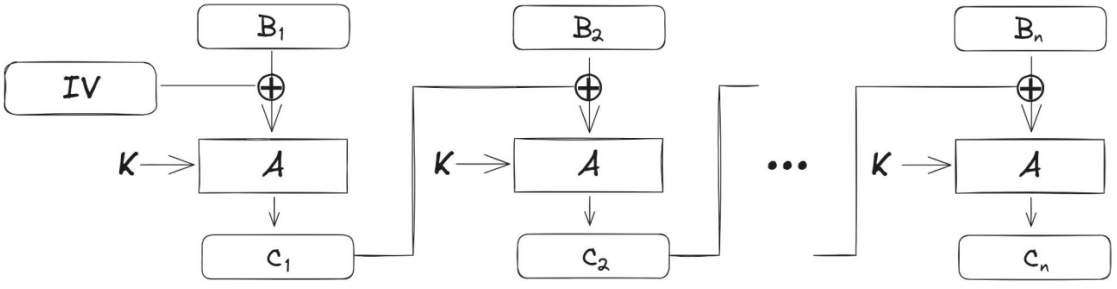
Electronic Codebook

Supongamos un block cipher A con clave K y un texto plano confirmado por n bloques $B_1, ..., B_n,$. El modo ECB (electronic codebook) encripta cada bloque independientemente de los otros:



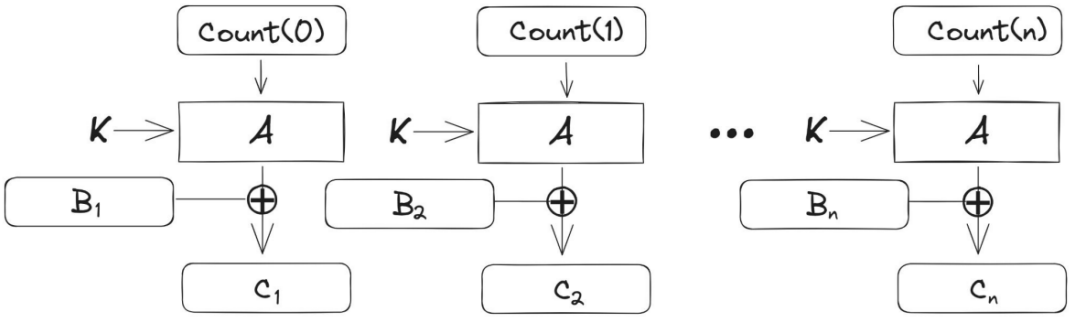
Cipher Block Chaining

El modo CBC “encadena” los bloques cifrados. Cada bloque se cifra a partir del XOR entre el cifrado de bloque anterior y el bloque actual.



CTR (Counter)

El modo CTR produce un stream de bits cifrados. Se cifra sucesivamente el valor de una secuencia incremental y se calcula el XOR del valor cifrado obtenido con el bloque correspondiente.



Criptografía simétrica: DES

DES: Data Encryption Standard

Consiste en una clave simétrica de 56 bits y toma textos planos de 64 bits. Se basa en el cifrado por bloques (block cipher). Sin embargo, es un método de criptografía muy vulnerable. En 1999, se logró quebrar una clave por fuerza bruta en menos de un día.

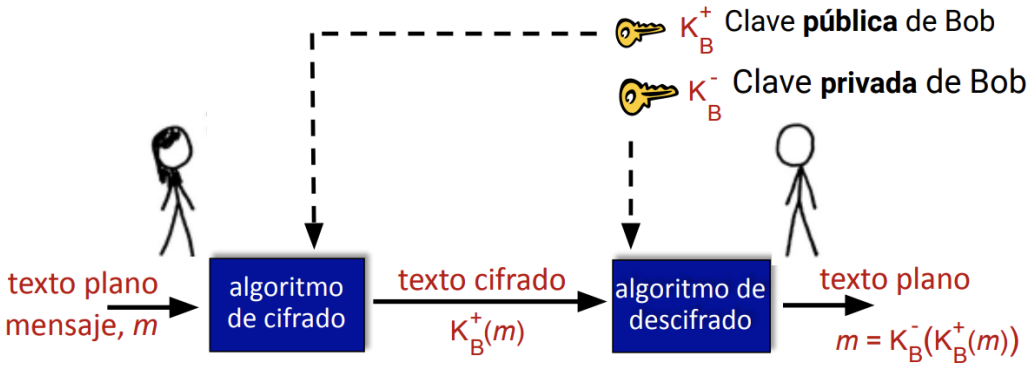
Se puede robustecer a partir de 3DES: cifrar tres veces con tres claves diferentes.

AES: Advanced Encryption Standard

En el 2001 reemplazó a DES y se convirtió en el standard del NIST. Procesa datos en bloques de 128 bits y emplea claves de 128, 192 o 256 bits.

Criptografía de clave pública

Mientras que la criptografía simétrica necesita que los interlocutores conozcan una clave compartida, la criptografía de clave pública adopta un enfoque muy diferente. Los interlocutores ya no comparten una clave secreta. En cambio, la clave de cifrado es pública (conocida por todos), mientras que la clave de descifrado es privada (es decir que solo es conocida por el receptor de los mensajes cifrados)



Los requisitos son:

1. $K_B^+(.)\ Y\ K_B^-(.)\ tales\ que:$
   $K_B^-(K_B^+(m)) = m$

2. Dada la clave pública K_B^+, debe ser computacionalmente inviable derivar las clave privada K_B^-

RSA: Concepto preliminares

Un mensajes es una secuencia de bits. Dicha secuencia puede representarse unívocamente como un número entero. Luego, en RSA, cifrar un mensaje equivale a cifrar un número.

RSA: Generación de claves

1. Elegir dos número primos p, q grandes (e.g. de 1024 bits).
2. Calcular $n\ =\ p * q$ y $z\ =\ (p-1) * (q-1)$.
3. Elegir e < n de modo tal que no posea factores comunes con z (e y z se dicen coprimos)
4. Elegir d de modo tal que $e * d - 1$ sea divisible por z.
5. La clave pública es el par $(n, e) =\ K_B^+$ y la clave privada es el par $(n, d) = K_B^-$

RSA: cifrado y descifrado

Dadas las claves (n,e) y (n,d):

1. Para cifrar el mensaje m (< n), calcular:

$$c = m^e \ mod \ n$$

2. Para descifrar el patrón de bits c, calcular

$$m = \ c^d \ mod \ n$$

Otra propiedad importante es:

$$K_B^- \ (K_B^+(m)) = \ m \ = \ K_B^+ \ (K_B^-(m))$$

Emplear primero la clave pública y luego la privada sobre un mensaje es lo mismo que emplear primero la clave privada y luego la pública. Esta propiedad es importante para las firmas digitales.

Seguridad de RSA

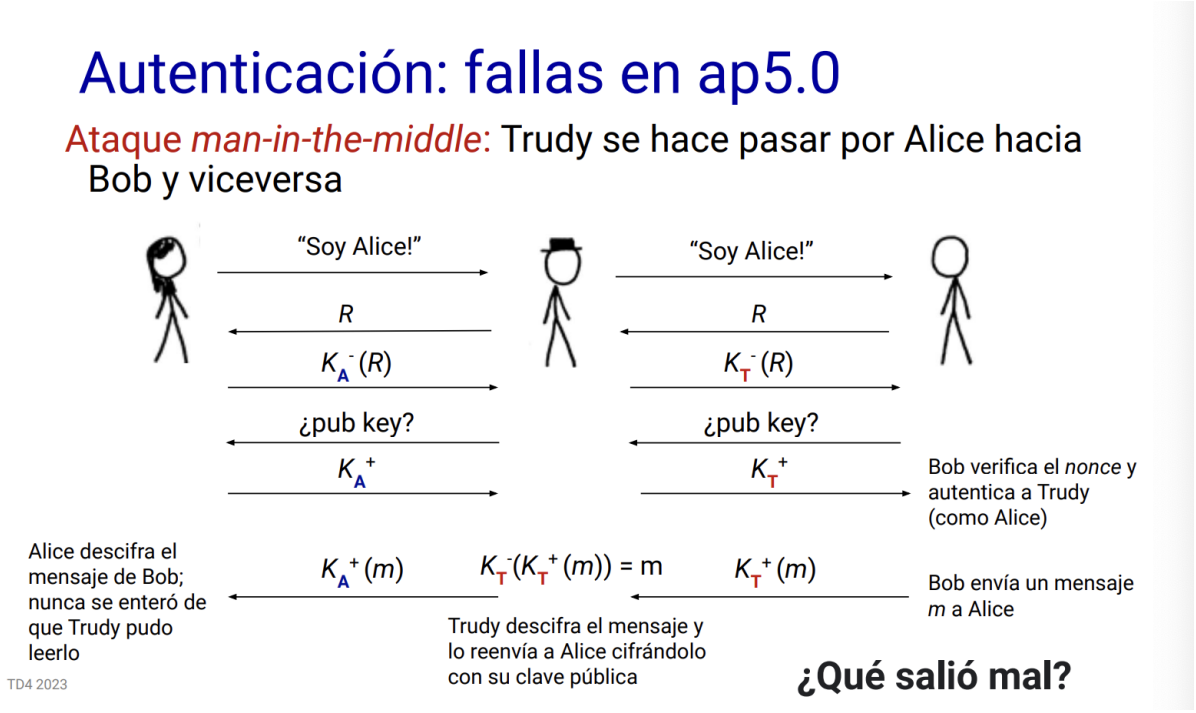
A partir de una clave pública (N, E), ¿cuán difícil es computar d?

Necesitamos factorizar N, es decir, encontrar sus factores primos. Esto es un problema computacionalmente difícil. No existen hasta ahora algoritmos tradicionales de tiempo polinomial para factorizar números suficientemente grandes. Mientras que los semiprimos (producto de dos primos) con factores de tamaño similar suelen ser más difíciles de factorizar (porque tienen factores grandes).

En la práctica, la exponenciación (modular) en RSA es costosa a comparación de los algoritmos simétricos (como AES) que permiten encriptar considerablemente más rápido. En la práctica, la criptografía de clave pública se utiliza para establecer una conexión segura y derivar una clave
compartida de sesión para cifrar los datos posteriores con un método simétrico.

Autenticación

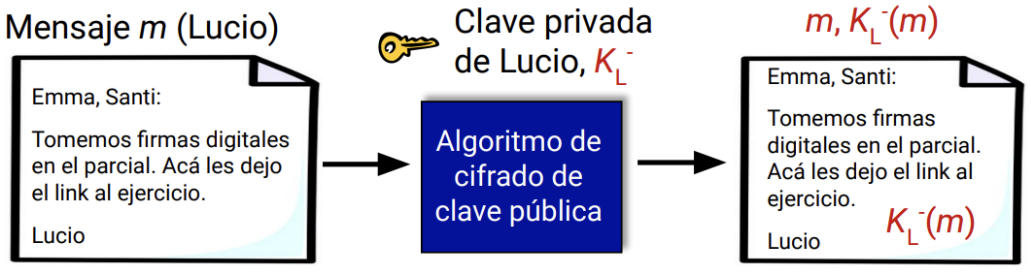
Un nonce es un valor numérico utilizado una única vez (R).



Integridad

Firmas digitales

Son mecanismo criptográficos análogos a las firmas en papel. El emisor firma digitalmente un documento de su autoría. Es verificable y no falsificable, es decir, el receptor puede probar que sólo el emisor del documento produjo la firma que lo acompaña. El esquema básico se compone en firmar el mensaje m cifrándolo con la clave privada K.



Supongamos que Emma recibe el mensaje m con firma $K_L^-(m)$. Emma verifica la firma aplicando la clave pública K_L^+ de Lucio a la firma digital y asegurando que

$K_L^+(K_L^-(m)) = m$. Si esto vale, quienquiera que haya firmado m debe haber empleado la clave privada de Lucio.

Luego, Valen verifica que:

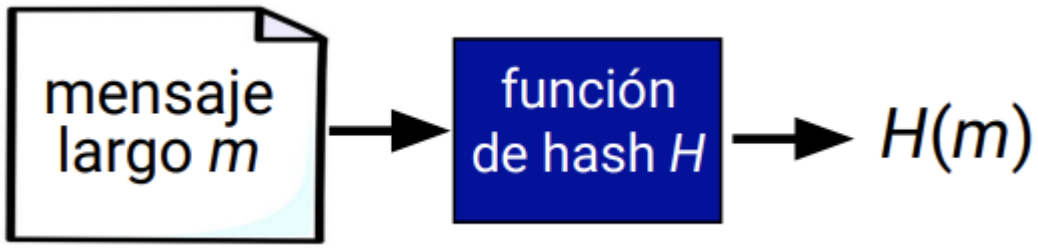
- Lucio firmó m
- Ninguna otra persona firmó m
- Lucio firmó m y no otro mensaje m' (integridad)

No se rechaza:

- Valen puede demostrar que fue Lucio quien firmó el mensaje m

Resúmenes de mensajes (digests)

Es computacionalmente costoso cifrar mensajes largos con criptografía de clave pública. Entonces, como solución se establece como objetivo poder obtener resúmenes de nuestros mensajes que sean de longitud fija y fáciles de computar. Para calcular estos digests de mensajes utilizamos **funciones de hash** criptográficas:



Algunas propiedades de las funciones de hash criptográficas:

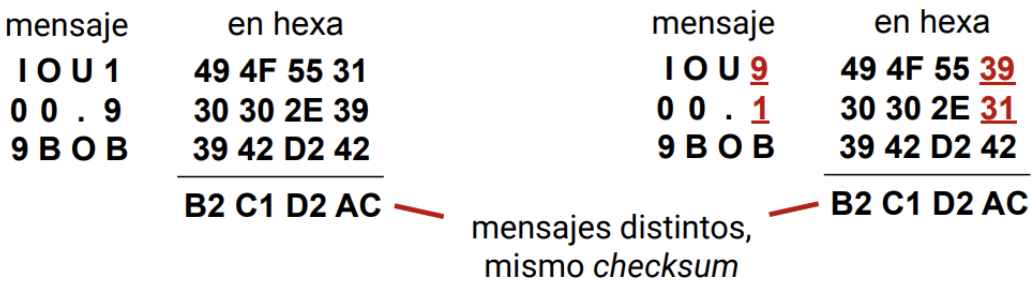
- No son inyectivas (”muchos a uno”)
- Produce una salida (digest) de longitud fija (e.g. 256 bits)
- Dado un digest x, es computacionalmente inviable encontrar un m tal que $H(m) = x$ (i.e. resistente a la preimagen)

Ejemplo (malo): checksum de Internet

El algoritmo de checksum de Internet posee algunas de estas propiedades:

- Genera digests de longitud fija. (16 bits)
- Es “muchos a uno”

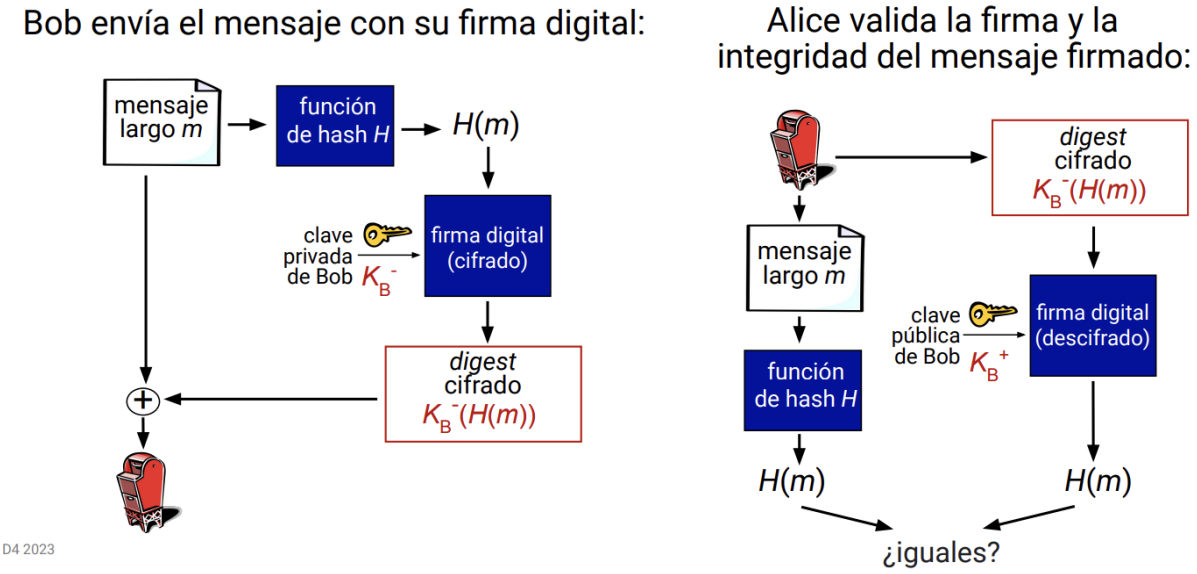
Sin embargo, dado un mensaje con cierto digest, es fácil encontrar otro mensaje con el mismo digest (no es resistente a colisiones)



Algunas funciones de hash populares

1. MD5 (RFC 1321)
    a. Calcula digests de 128 bits en un proceso de cuatro rounds.
    b. Si bien es vulnerable, continúa en uso (por ejemplo, para verificar integridad en las transferencias de archivos).
    c. En 2013, se descubrió un ataque para generar colisiones de MD5 que corre en menos de un segundo en una computadora estándar.
2. SHA-1 es otra popular (también vulnerable)
    a. Estandarizada en EEUU por NIST.
    b. Produce digests de 160 bits.
    c. SHA-2 y SHA-3 (sus sucesoras) se consideran seguras al día de hoy.

Firma digital: digest firmado

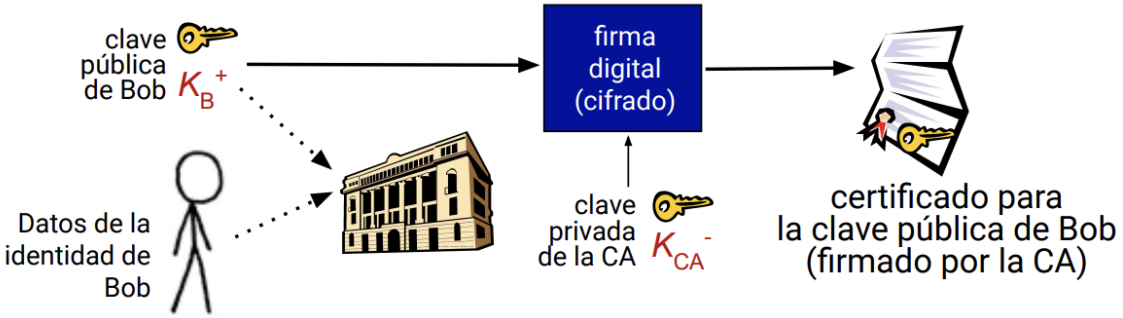


Corrección del protocolo ap5.0

Para corregir el protocolo ap5.0, necesitamos verificar las claves públicas. Si Bob pudiera comprobar que la clave pública de Alice es auténtica, el ataque man-in-the-middle de Trudy no tendría efecto. Es por eso que existen las Autoridades de Certificación (CAs).

Autoridades de Certificación (CAs)

Las autoridades de certificación (CAs) vinculan una clave pública con una entidad (persona física, sitio web, router, etc.) Cuando una entidad E desea certificar su clave pública, la CA debe verificar la identidad del solicitante. La CA genera un certificado vinculando la identidad de E con su clave pública. Dicho certificado está firmado digitalmente por la CA.



Entonces, cuando Alice quiere utilizar la clave pública de Bob, obtiene el certificado de Bob, utiliza la clave pública de la CA para validar el certificado y extrae el certificado de la clave pública.

Protocolos seguros en Internet: TLS

Transport Layer Security

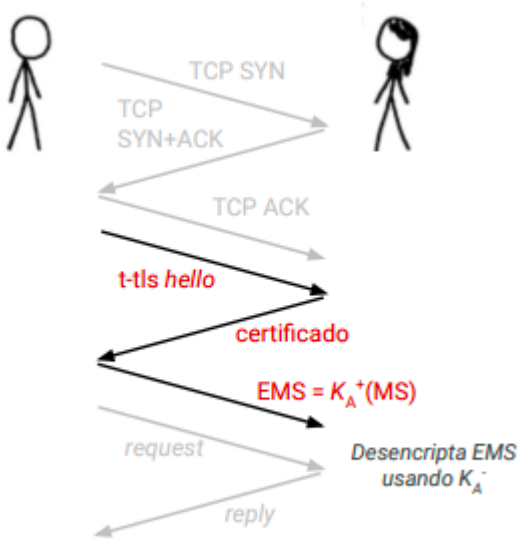
Protocolo seguro que opera sobre la capa de transporte del stack TCP/IP. Soportado por prácticamente todos los navegadores y servidores we: HTTPS (Puerto 443). TLS ofrece **confidencialidad** (vía cifrado simétrico), **integridad** (vía hashes criptográficos) y **autenticación** (vía criptografía de clave pública).

Veamos que necesitamos en el protocolo mediante una simplificación del mismo: t-tls (Toy TLS). Ya estudiamos las distintas componentes:

- Handshaking: los interlocutores emplean sus certificados y claves privadas para autenticarse mutuamente; luego intercambian o generan un **secreto** compartido por ambos.
- Derivación de claves: los interlocutores generan una serie de claves a partir del secreto compartido.
- Transferencia de datos: el stream de datos se divide en registros (records) acompañados de digests y posteriormente encriptados.
- Cierre de conexión: Mensajes de control especiales para cerrar de forma segura la conexión.

Durante el handshake de T-TLS:

1. Bob establece una conexión TCP con Alice.
2. Bob comprueba la identidad de Aalice mediante su certificado.
3. Bob genera un secreto (Master Secret, MS) y se lo envía a Alice cifrado con su clave pública (de aquí se derivarán luego todas las claves de sesión)
4. Se necesitan 3 RTTs antes de que los interlocutores puedan intercambiar datos.



Se generan claves distintas para cada propósito: dos para cifrado de datos y dos para los mecanismos de integridad. Es mucho más seguro que tener una única clave.

- K_c: clave de cifrado de datos cliente → servidor
- M_c: clave HMAC para datos clientes → servidor
- K_s: clave de cifrado de datos servidor → cliente
- M_s: clave HMAC para datos servidor → cliente.

Hash-based Message Authentication Code (HMAC) es un digest calculado a partir de una función de hash y una clave. Estas claves se generan mediante la función KDF (Key Derivation Function), la cual utiliza el master secret y datos aleatorios.

Cifrado de datos

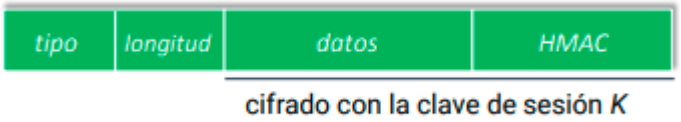
TCP ofrece un stream de bytes a los protocolos de aplicación. El stream se fragmenta en una secuencia de registros (records). Cada registro incluye una HMAC (calculado a partir de la clave de sesión M). Cada registro es cifrado con la clave simétrica K y luego se entrega a socket TCP subyacente:

No es posible ejecutar ataques sobre el stream de datos:

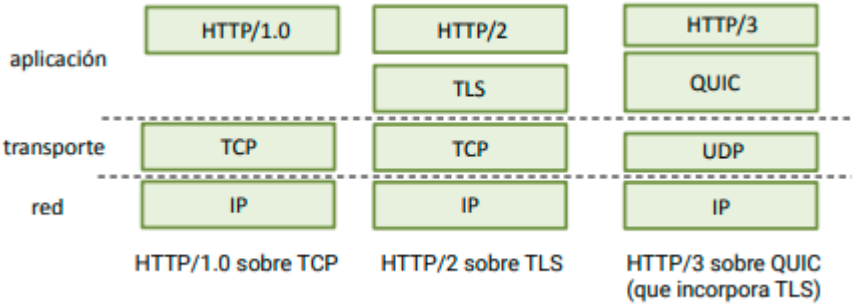
TLS emplea número de secuencia y nonces. Los números de secuencia son “implícitos”. Mientras que los interlocutores mantienen cuentas separadas de los mensajes transmitidos y recibidos; ese valor se incorpora dentro del cálculo del HMAC. Los nonce son valores aleatorios elegidos durante el handshake que impactan en la derivación de claves de sesión.

Cierre de conexión TLS

TLS contempla tipos de registro e incluye uno para notificar el cierre de la sesión. Si bien el tipo va en texto plano, queda protegida su integridad por el HMAC del registro.



TLS desde la óptica de HTTP



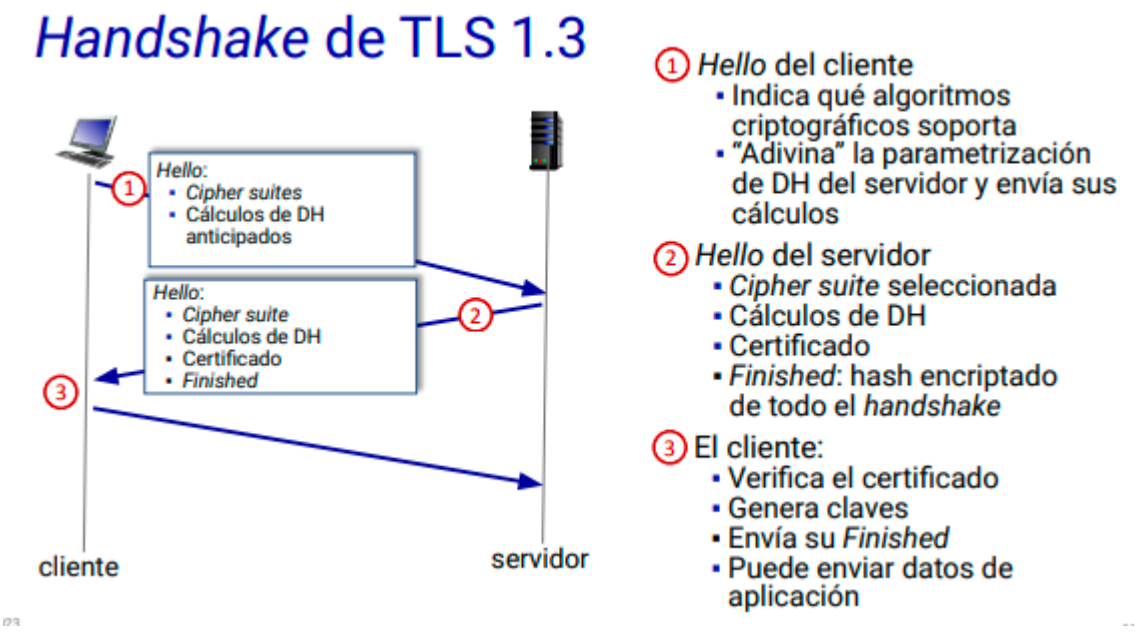
TLS 1.3

Esta versión de TLS surgió con el objetivo de mejorar la performance y la seguridad de las versiones anteriores.

> La cipher suite es una paleta de algoritmos criptográficos para generar claves, cifrar datos y calcular MACs. En el handshake se llega a un acuerdo de cuáles de estos utilizar.

TLS 1.3 recorta considerablemente la suite de cifrado (cipher suite), dejando sólo 5 opciones respecto de las 37 de TLS 1.2. Además, sólo permite el uso de Diffie-

Hellman (DH) para generar claves, dejando de lado la opción RSA (Esto es por que es muy difícil de implementar correctamente). DH es un algortimo criptográfico que permite que los interlocutores generen una clave compartida sin conocimineto previo mutuo. También permite comprimir el handshake a 1 RTT.



Firewalls

Son dispositivos que permiten aislar de Internet la interna de una organización. Además, implementan políticas para permitir ciertos paquetes y descartar otros. Existen tres tipos de firewalls: filtrado sin estado (stateless packet filters), filtrado con estado (stateful packet filters) y gateways de aplicación (application gateways)

Funciones de los firewalls

1. *Prevenir acceso/manipulación de datos privados.*
    a. e.g., un atacante reemplaza la página web de la UTDT con otra cosa (defacement)
2. Permitir acceso autorizado a la red interna
    a. Conjunto de usuarios/hosts autenticados
3. Prevenir ataque DoS (Denial of Service)

a. SYN flooding: un atacante abre muchas conexiones TCP falsas, comprometiendo los recursos del sistema para establecer conexiones legítimas

Filtrado Stateless

El firewall filtra paquete a paquete, tomando decisiones a partir de: direcciones IP origen y destino, puerto origen y destino y protocolo de transporte (TCP/UDP), tipo de mensaje ICMP y flags de TCP.

Un ejemplo podría ser bloquear datagramas entrantes y salientes con puerto de transporte 23 (origen o destino). Es decir que no se permiten conexiones vía telnet (un protocolo inseguro de login remoto).

Otro ejemplo sería bloquear segmentos TCP entrantes sin flag de ACK. Por corolario no se permite que se generen conexiones TCP desde el exterior, pero sí que se generen desde la red interna hacia afuera.

Listas de control de acceso (ACLs)

Son una tabla de reglas que se aplican en orden a los paquetes (match + action à-la OpenFlow)

acción	IP src	IP dst	proto	puerto src	puerto dst	*flags*
allow	222.22/16	fuera de 222.22/16	TCP	> 1023	80	*
allow	fuera de 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	fuera de 222.22/16	UDP	> 1023	53	---
allow	fuera de 222.22/16	222.22/16	UDP	53	> 1023	----
deny	*	*	*	*	*	*

Filtrado stateful

Traza el estado de cada conexión TCP. Registra el establecimiento y el fin de las conexiones y puede determinar si los paquetes intercambiados son consistentes. En la ACL, se indica si se debe verificar el stado de la conexión antes de admitir los paquetes.

acción	IP src	IP dst	proto	puerto src	puerto dst	*flags*	conexión
allow	222.22/16	fuera de 222.22/16	TCP	> 1023	80	*	
allow	fuera de 222.22/16	222.22/16	TCP	80	> 1023	ACK	X