Seguridad en Redes

Hablar de seguridad en las redes es algo muy ambiguo y que abarca un rango amplio de conceptos, puntos de vista, técnicas, paradigmas y demás. Por ello haremos un breve recorrido de los distintos aspectos que podemos encontrar en este amplio tema y nos centraremos en un par de puntos clave en lo referente a seguridad y privacidad en las comunicaciones en redes modernas tipo tcp/ip que podemos encontrar en sitios como el propio hogar, empresas y lugares públicos.
Comenzaremos definiendo algunos conceptos que envuelven los distintos ámbitos de la seguridad en redes y recordaremos las distintas capas que componen el modelo de red, analizando los mecanismos de seguridad que podemos encontrar en cada una de ellas.
Seguidamente analizaremos en profundidad dos mecanismos y protocolos que supusieron un antes y un después en la concepción de redes seguras, como son la implementaciones de protocolos de seguridad en redes inalámbricas como el protocolo WEP e implementaciones de canales seguros de comunicación con SSL/TLS.

1 Introducción a la seguridad

Repasemos cual son las características que forman el concepto de seguridad y que se aceptan como los pilares que sustentan tanto terminos, aplicaciones, herramientas y acciones:

- **Integridad:** La integridad hace referencia a la cualidad de la información para ser correcta y no haber sido modificada, manteniendo sus datos exactamente tal cual fueron generados, sin manipulaciones ni alteraciones por parte de terceros. Esta integridad se pierde cuando la información se modifica o cuando parte de ella se elimina, y una gran garantía para mantenerla intacta puede ser la firma digital.
- **Disponibilidad:** Capacidad de disponer de la información a la que podemos acceder cuando la necesitamos a través de los canales adecuados siguiendo los procesos correctos.
- **Confidencialidad:** cualidad de la información para no ser divulgada a personas o sistemas no autorizados. Se trata básicamente de la propiedad por la que esa información solo resultará accesible con la debida y comprobada autorización.

Nos centraremos en esta última propiedad, la confidencialidad.

1.1 Introducción a la seguridad de redes TCP/IP

En general cuando hablemos de redes nos referiremos a redes informáticas o de computadores basadas en el modelo TPC/IP. Podemos encontrar muchos tipos de redes distintas podemos hacer un par de subgrupos de tipos de redes:

- Según él área que abarca:
    - **LAN** (Local area network o red de área local) redes mas cercanas al usuario, que podemos encontrar en hogares, empresas y lugares públicos.

    - **WLAN** (Wireless Local Area Network) redes inalámbricas que típicamente son accesibles a usuarios de forma sencilla y directa.
    - **WAN** (Wide Area Networks o red de área amplia) abarcan un área más grande y están formadas por subredes distintas.
    - **VPN** (Virtual Private Network o red privada virtual) es una red de comunicación virtual que utiliza la infraestructura de una red física para asociar sistemas informáticos de manera lógica y que se puede encontrar en cualquiera de las redes anteriores.
- Según su propiedad:
    - **Redes públicas**: abiertas a todos los usuarios
    - **Redes privadas**: son propiedad de empresas, organizaciones o asociaciones, particulares.

Con esta subdivisión de redes podemos pasar a definir las grandes infraestructuras que podemos encontrar en él mundo, como son las **intranet**, que son redes privadas formadas por subredes conectadas entre sí y que pueden abarcar o no una gran área. Estas redes pueden pertenecer a empresas o instituciones como por ejemplo la red interna de la UGR. Y como no la red de redes, **Internet**, una red pública mundial formada por millones de subredes y con un alcance mundial.
Las redes están ahí para enviar y recibir información, compartir recursos, dar acceso a determinados sistemas, control de mecanismos, etc… Esta información en muchas ocasiones es sensible, delicada y privada y los usuarios quieren y buscan que su información siga siendo privada y que sus sistemas no se vean comprometidos a que alguien externo sea capaz de controlarlo o incluso “destruirlo”, o incluso que la red se vea comprometida por una configuración pobre que ponga en riesgo la integridad de la misma.

Ahora nos surge una pregunta ¿Cuales son los puntos sensibles de una red y que podemos hacer para cubrirlos?

Partes y aspectos de una red segura

Este problema se puede ver desde distintos puntos de vista podemos verlo a nivel de protocolos y capas de red o podemos verlo desde un punto más general, un punto más cercano al usuario y destinado a prevenir acciones maliciosas de terceros o malos hábitos de los usuarios. Partiendo desde este último punto de vista, podemos distinguir algunos puntos claves:

- **El concepto de capas:** Debemos no debemos tratar la red como un todo inamovible sino como un conjunto de capas que se superponen, se completan y se apoyan entre ellas, y extrapolar este concepto a todos sus ámbitos. En cada capa tendremos un problema distinto que abordar, así dividimos en problema grande (una red entera) en partes más pequeñas y manejables. Por lo tanto si tenemos una red debemos distinguir entre capa de seguridad, capa de arquitectura logica y fisica...etc. Al mismo tiempo dentro de la capa como la seguridad podemos subdividir los problemas, autenticación, accesos, privilegios de usuarios, dominios y demás.
- **autenticación:** A la hora de acceder a la red o a algún recurso que esta ofrece, normalmente queremos que NO cualquier persona pueda acceder y/o mantener un registro de quien accede a que cosas. Por ello debemos implementar una capa de seguridad al acceso de la red y sus recursos. Por ejemplo con nombre de usuario y contraseña, que representa e identifica a un usuario particular. Podemos encontrar

también autenticación de doble factor donde se le reclama al usuario no solo nombre y clave sino también algo que “tenga”, una llave, una tarjeta o un teléfono. Incluso podemos pedirle algo que le identifique personalmente: huella dactilar, reconocimiento facial, etc... Esto se conoce como autenticación de triple factor.

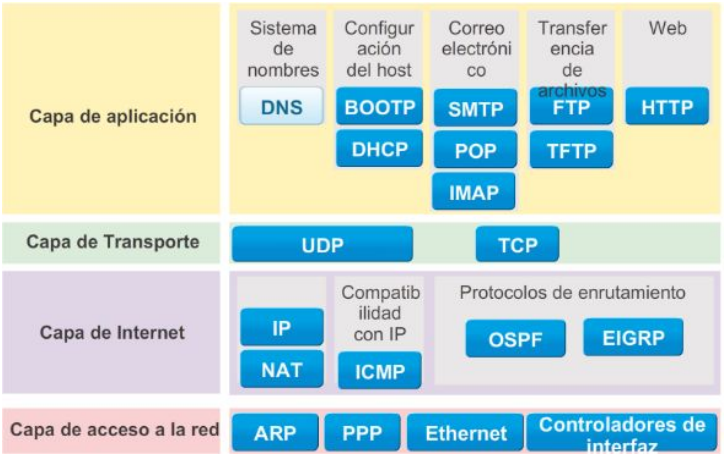
- **políticas de acceso**: Añadimos una nueva capa de seguridad con las políticas de acceso donde por ejemplo definimos qué usuarios pueden acceder a qué recursos esto lo podemos aplicar con cortafuegos, que puede ser software o hardware destinado a bloquear accesos no autorizados, controlar comunicaciones e incluso delimitar redes internas entre si. A todo esto también le podemos añadir sistemas de prevención de intrusos, que según él comportamiento y anomalías en la red pueden predecir si hay algún intruso intentando acceder a la red y a sus sistemas.
- **Control de acceso:** Una paso más allá de las políticas de acceso donde la preocupación ahora reside en el tráfico mismo de los datos, quien accede a ellos cuantas veces y en qué momento o quién hace qué cosa. Este control del tráfico que podemos hacer se basa en las llamadas listas blancas o whitelisting, son listas niegan o permiten el acceso a determinados usuarios según su comportamiento. Por ejemplo si un usuario hace mil peticiones en menos de un minuto, este será inscrito en una lista de elementos maliciosos en la red, indicando así que se le niega el acceso.
- **Arquitectura de red**: Si una red está construida tanto física como lógicamente de forma pobre sin tener en cuenta futuras configuraciones o sin un estudio previo, puede conllevar graves fallas de seguridad e integridad de la misma, por ello este es unos de los aspectos más importantes de la red. Se debe hacer un estudio preliminar de la arquitectura que sea mejor para la empresa o institución que la necesite. Y por supuesto las redes se deben construir bajo los conceptos de capas que ya hemos ido adelantando. Este modelo por capas nos facilitará la comprensión y configuración de la red, que se separa en distintas capas y cada capa a su vez se divide en distintas capas de configuración. Pongamos un ejemplo: si necesitamos una red para una empresa podríamos distinguir parte de gestión de la empresa y parte de servicios, creando dos capas distintas asociadas a dos subredes que luego configuraremos con las necesidades de seguridad de cada una, así si alguien externo atacar la red debería debería recorrer varias capas para de seguridad dentro de varias subredes, cosa que le llevaría más bastante tiempo. cuanto más capas más difícil atacar todo el conjunto de la red. Otra técnica que se puede utilizar es crear redes señuelo con configuraciones pobres de redes que no sirven para la organización solo sirven para que los intrusos se “pierdan” en ellas. Por lo tanto a la hora de construir una red debemos tener en cuenta dos dimensiones distintas, siendo la “profundidad” de la defensa, con configuraciones firewalls, técnicas de encriptación etc… y la “altura” o planos de la red dividiendo los distintos aspecto de la institución en distintas subredes (red de gestión, de servicio, DMZs, VPN, VLAN).
- **Seguridad física de dispositivos finales:** Los elementos físicos también están expuestos a ataques o averías, deben estar monitorizados, actualizados y protegidos para cualquier imprevisto. Podemos reforzar la seguridad no dejando lla vista los dispositivos de red y dedicando espacios seguros y cerrados como por ejemplo cajas con cerrojos y habitaciones cerradas. También invirtiendo en sistemas de prevención de incendios o de sobrecargas.
- **Encriptación:** Para mayor privacidad todas las comunicaciones deben ser encriptadas, es la parte de las redes más olvidadas de todas y él que siempre es el

último en implementarse o incluso cuando el desastre ya ha ocurrido. Encriptar comunicaciones puede complicar la computación de la información pero el precio que se pasa es merecido si con ello nuestra información puede ser más segura. por ello más adelante nos centraremos en técnicas y protocolos que usan la encriptación para hacer más segura una red

- **Redes inalámbricas:** debemos añadir una mención especial a las redes inalámbricas que conlleva un riesgo adicional a la seguridad. por lo que deben implementarse únicamente bajo necesidad o decisión particular por la organización y siempre bajo condiciones justificadas. Y en el caso de su uso utilizar mecanismos lo más seguros posibles, que veremos más adelante.

1.1 Modelo de capas TCP/IP

El modelo TCP/IP se usa actualmente para comunicaciones entre computadoras en red, es un modelo que describe una serie de protocolos de red, apilados y organizados por capas.



Seguiremos este modelo para ver qué mecanismos de seguridad encontramos en cada capa.

Comenzamos por las capas más altas **Aplicación y transporte** aquí encontramos protocolos de comunicación como HTTP, FTP, TELNET proporcionan formas de transmisión de datos como texto plano, cualquier tipo de archivo o líneas de órdenes. También tenemos protocolos TCP y UDP que proporcionan un mecanismo para enviar esa información. la mayoría de protocolos en su concepción obviaba una parte muy importante y era la privacidad y de integridad, por ello se crean nuevos protocolos y se modifican otras para introducir privacidad, integridad de datos y disponibilidad. Encontramos así en estas capas protocolos como SSL/TLS usados para encriptar comunicaciones. Bajo este protocolo TLS se modifican y crean nuevos protocolos como HTTPS, FTP, SNMP v3 y SSH. En cuanto a protocolos de transporte TCP en sus últimas versiones y modificaciones incorpora una capa extra de integridad, dedicada a minimizar la pérdida de datos y maximizar la latencia, e implementa caminos de datos junto a protocolos de capas inferiores. también tenemos protocolos destinados a mejorar el rendimiento de las transmisiones HTTP como como SPDY. Es similar el protocolo SCPT, que se caracteriza por fragmentar datos, agrupación de mensajes, multiplexación y alta tolerancia a fallos haciendo que la comunicación sea más segura y fiable.

Bajando nos encontramos con una **capa de internet** donde encontramos protocolos de direccionamiento y enrutamiento, usados para encaminar los datos a través de la red. Lo que buscan los protocolos de esta capa es maximizar la disponibilidad y acelerar la búsqueda de un camino óptimo para nuestros mensajes. También gracias a estos protocolos somos capaces de ocultar el camino que toman los datos para evitar que alguien externo pueda seguirlos, por ejemplo usando VPN(OpenVPN, PPTP y L2TP/IPSec) o Proxy IP.

Por último **capa de acceso a red,** donde se encuentran todos los protocolos relacionados con el medio que se va a usar para transmitir la información. Sobretodo aquí encontramos protocolos que eviten la explotación de posibles vulnerabilidades como STP (Spanning Tree Protocol), o protocolos MAC y ARP que incorpora mecanismos de resolución de conflictos en red. Esta capa es sensible a manipulación, al ser muy cercana al usuario, por ejemplo quien se conecte a una WIFI él medio por donde la información se transmite es público, todo el mundo puede acceder a él. Necesitamos mecanismos que eviten un uso inapropiado del medio, tenemos WEP que encripta las comunicaciones inalámbricas.

Con este repaso damos pie a un análisis en profundidad de un par de mecanismos y protocolos de seguridad como son WEP y TLS.

2. Protocolos de seguridad en WLAN

El cifrado de redes LAN inalámbricas se utiliza para añadir seguridad a una red inalámbrica mediante un protocolo de autenticación, que solicita una contraseña o clave de red cuando un usuario o dispositivo intenta conectarse. Si tu red inalámbrica no está asegurada con algún tipo de cifrado, es posible que usuarios no autorizados accedan a ella y obtengan información personal, o que utilicen tu conexión a Internet con fines maliciosos o ilegales. Además, también puede reducirse la velocidad o el rendimiento de tu red si otras personas la utilizan sin tu conocimiento. Por ellos vamos a hablar sobre el funcionamiento de alguno de los protocolos más utilizados en la seguridad de redes inalámbricas.

2.1. WEP

El **protocolo WEP** (Wired Equivalent Privacy) es el protocolo de cifrado para redes inalámbricas incluido en el estándar IEEE 802.11 y presentado en el año 1999 que permite ofrecer seguridad de acceso a la red y proteger la información transmitida. Debido a la naturaleza del medio de transmisión es más fácil que el intercambio de información sea interceptado. Debido a la aparición de ciertas vulnerabilidades importantes en 2003 sería sustituido por su predecesor WPA. WEP proporciona confidencialidad a través del uso del criptosistema RC4 e integridad a través del sistema CRC32.

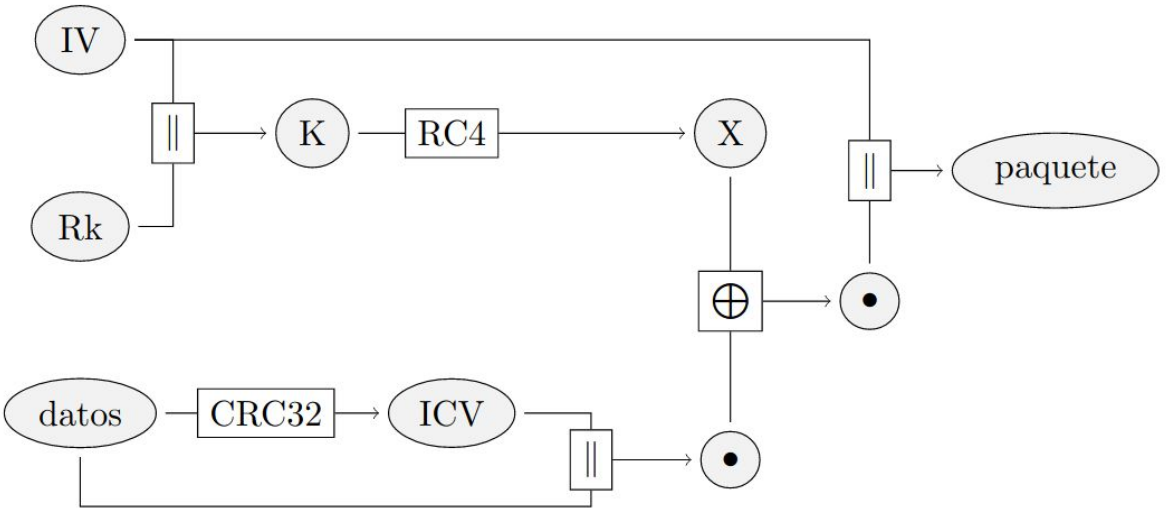
Dentro de la criptografía, **RC4** es el sistema de cifrado de flujo más utilizado. Los cifradores de flujo son algoritmos de cifrado que pueden realizar el cifrado incrementalmente,

convirtiendo el texto claro en texto cifrado bit a bit. La clave secreta utilizada por RC4 consiste en una ***root key*** (Rk), que comparten todos los usuarios autorizados, a la que se le añade un **vector de inicialización** (IV) que cambia en cada paquete que se transmite.

Para comprobar la integridad de los datos, cada paquete tiene un campo llamado ***Integer Check Value*** (ICV) que es el *checksum* (función hash) de los datos que se envían en el paquete. Tanto los datos del paquete como el ICV están cifrados utilizando RC4.

A continuación vamos a ver como es el proceso de cifrado llevado a cabo por WEP para enviar un paquete por la red:

1. CRC32 calcula el *checksum* ICV a partir de los datos y es concatenado al final de estos.
2. Se elige un IV o bien, partiendo de un IV inicial e incrementando en 1 para cada paquete, o bien, generando un número pseudoaleatorio para cada paquete.
3. El IV y el Rk se concatenan y forman la clave K = IV || Rk.
4. La clave K se introduce en el algoritmo RC4 para generar el *keystream* X de la misma longitud que los datos y el ICV.
5. Se realiza la operación XOR entre los datos en claro y el *ICV* con X y se obtienen los datos cifrados.
6. Los datos cifrados, el IV en claro y las cabeceras construyen un paquete para ser enviado.



Al poco tiempo se le descubrieron vulnerabilidades serias al protocolo WEP de manera que tuvo que ser reemplazado. Algunas de estas vulnerabilidades son:

- **Claves cortas.** Si se utilizan claves cortas se puede aplicar fuerza bruta para descifrar los mensajes.

- **Reutilización de *keystreams.*** Si un *keystream* es descubierto, todos los mensajes cifrados con él pueden ser descifrados. El keystream puede ser obtenido a partir de tener un texto claro y el cifrado.
- **RC4.** La clave de una red WEP puede ser revelada si se utilizan IV débiles en el criptosistema RC4.
- Los últimos ataques han conseguido recolectar 40.000 en 60 segundos para conseguir una **eficacia del 50%** rompiendo el cifrado.
- Es posible alterar los datos y actualizar el ICV sin conocer la clave, rompiendo así la integridad.

2.2. WPA y WPA2

Como hemos comentado anteriormente WPA (*Wi-fi Protected Access*) fue creado para corregir las deficiencias de WEP. WPA implementaría la mayoría del estándar IEEE 802.11i y fue creado como una medida intermedia para ocupar el lugar de WEP mientras 802.11i era finalizado. Una vez terminado se crea el WPA2 basado en este. WPA se puede considerar la versión de migración.

WPA adopta la autenticación de usuarios mediante el uso de un servidor, que distribuye claves diferentes a cada uno. En este se almacenan las credenciales y contraseñas de los usuarios de la red. Para no obligar al uso de tal servidor para el despliegue de redes, WPA permite la autenticación mediante una clave precompartida, de modo similar a WEP. Este modo es menos seguro pero se puede utilizar para redes pequeñas como usuarios domésticos o pequeñas oficinas. La información es cifrada utilizando RC4 con una clave de 128 bits y un vector de inicialización de 48 bits.

Una de las mejoras sobre WEP, es la integración del Protocolo de Integridad de Clave Temporal (TKIP), que cambia las claves dinámicamente a medida que el sistema es utilizado. Cuando esto se combina con un vector de inicialización mucho más grande, evita los ataques de recuperación de clave a los que es susceptible WEP.

Adicionalmente a la autenticación y cifrado, WPA también mejora la integridad de la información cifrada. El algoritmo CRC32 utilizado por WEP ya hemos visto que es inseguro. WPA implementa *Message Integrity Check* (MIC) para la integridad.

2.3. WPS

WPS (**Wi-fi Protected Setup**) se trata de un conjunto de diversos mecanismos para facilitar la configuración de una red WLAN segura con WPA2, pensados para minimizar la intervención de los usuarios. Concretamente, WPS define los mecanismos a través de los cuales los diferentes dispositivos de la red obtienen las credenciales necesarias para iniciar el proceso de autenticación. WPS define una arquitectura con tres elementos con diferentes roles:

- **Registrar**: dispositivo con la autoridad de generar o revocar las credenciales en la red. Tanto un AP como cualquier otra estación o PC de la red pueden ejercer este rol.
- **Enrollee** (Matriculado): dispositivo que solicita el acceso a la red WLAN.
- **Autenticador**: AP funcionando de proxy entre el Registrar y el Enrollee.

WPS contempla cuatro tipos de configuraciones para realizar el intercambio de credenciales:

- **PIN**: tiene que existir un PIN asignado a cada elemento que vaya a asociarse en la red. Este PIN tiene que ser conocido tanto por el Registrar, como por el usuario (Enrollee).
- **PBC**: la generación y el intercambio de credenciales son desencadenados a partir de que el usuario presiona un botón (físico o virtual) en el AP (o en otro elemento Registrar) y otro en el dispositivo que se quiere conectar a la red. En un corto lapso de tiempo en el que se presionan ambos botones, cualquier estación puede ganar el acceso a la red.
- **NFC**: El intercambio de credenciales se puede realizar a través de NFC.
- **USB**: con este método, las credenciales se transfieren mediante un dispositivo de memoria flash desde registrar al Enrollee.

3 Introducción a SSL / TLS

La mayoría de las veces hemos utilizado el protocolo SSL o TLS sin nisiquiera darnos cuenta. Una prueba de ello sería el protocolo HTTPS. Este protocolo utilizado mundialmente usa estos métodos para garantizar la seguridad en la red.

SSL (Secure Socket Layer) es un protocolo antiguo en desuso en favor de TLS (Transport Layer Security). Esta denominación hace referencia a la capa de transporte del modelo TCP/IP. TLS es un proceso que encripta los flujos de datos de Internet para que solo sus legítimos destinatarios puedan leerlos.

TLS es un protocolo para la transmisión segura de datos basado en SSLv3 (una capa de seguridad de 1996). Ofrece confidencialidad, integridad y autenticación . En términos sencillos, esto significa:

- Confidencialidad: oculta el contenido de los mensajes, es decir, es la capacidad de garantizar que la información será protegida y solamente podrá acceder a ella la persona a quien va dirigida
- Integridad: detecta cuando los mensajes han sido manipulados, es decir, es la capacidad de garantizar que los datos no se modifican desde su envío hasta su recepción.
- Autenticación: es la capacidad de garantizar que el interlocutor es quien realmente dice ser.

Además, detecta mensajes perdidos y duplicados.

TLS es la forma principal de proteger el tráfico web y se utiliza principalmente para ese propósito. Muchas páginas confían en que TLS es seguro (desde la tienda en línea más pequeña hasta Facebook), por eso cosas como POODLE (este ataque requiere que exista previamente un ataque Man-in-the-Middle y además hacer creer al cliente que la versión más estable que soporta es SSL 3.0) y Heartbleed (este fallo de implementación en OpenSSL permitía que un atacante pudiese leer hasta 64 Kb de memoria por ataque en cualquiera de ambos extremos) reciben tanta atención.

Como ya hemos dicho, vulnerabilidades como Heartbleed, Poodle o incluso Winshock habían terminado de minar la poca confianza que quedaba en SSLv3. Además, alguna de estas vulnerabilidades tenía como objetivo también a TLS, lo que puede que haya ayudado a acelerar el desarrollo de nuevas versiones. Aun teniendo todo esto en cuenta, aún son muchas las webs, aplicaciones y todo tipo de servidores que siguen soportando SSLv3.

3.1 Confidencialidad e integridad en TLS

Para la **confidencialidad**, TLS utiliza una combinación de cifrado simétrico y asimétrico para asegurar la confidencialidad de los mensajes. En la fase de handshake, el cliente y servidor acuerdan un algoritmo de cifrado y una clave compartida con la que cifrarán todos los mensajes transmitidos entre ambos, garantizando confidencialidad del mensaje incluso si es interceptado. TLS utiliza AES con varios modos disponibles (CBC, CCM, GCM). Para el intercambio de claves, están disponibles Diffie-Hellman (compatible con el modo de curva elíptica) o RSA. Debido a los problemas de Heatbleed, ahora se recomienda usar un mecanismo de intercambio de claves secretas (PFS - que garantiza que el descubrimiento de las claves utilizadas actualmente no compromete la seguridad de las claves usadas con anterioridad), lo que implica Diffie-Hellman en este caso.

Uno de los principales problemas de RSA y Diffie-Hellman es que se establece la clave de sesión secreta haciendo uso de las claves públicas y privadas de los interlocutores, siendo siempre la misma clave secreta para la comunicación entre ambos. Esto da lugar a que las claves privadas puedan ser comprometidas en un futuro, dado que si un atacante adivina la clave secreta, este entonces podrá descifrar todos los mensajes de todas las comunicaciones que hubiese grabado entre ambos extremos.

Otro problema por el cual se sustituyen los algoritmos anteriores por algoritmos de curvas elípticas es porque estos sistemas utilizan curvas elípticas para crear claves criptográficamente más fuertes y de menor longitudes, lo que hace a estos algoritmos más eficientes y rápidos de implementar.

No obstante, ambos han quedado obsoletos gracias a algoritmos que proveen Secreto Perfecto Hacia Adelante (Perfect Forward Secrecy), el cual permite crear una clave secreta distinta para cada comunicación. Esta clave es posteriormente eliminada, volviendo imposible recuperar las claves en un futuro.

Para la **integridad**, TLS proporciona integridad de los mensajes enviados calculando un resumen o hash del mensaje. Este algoritmo de hash es acordado

entre ambos ambos interlocutores. Se usa HMAC comúnmente, hoy en día es prácticamente obligatorio usar SHA256.

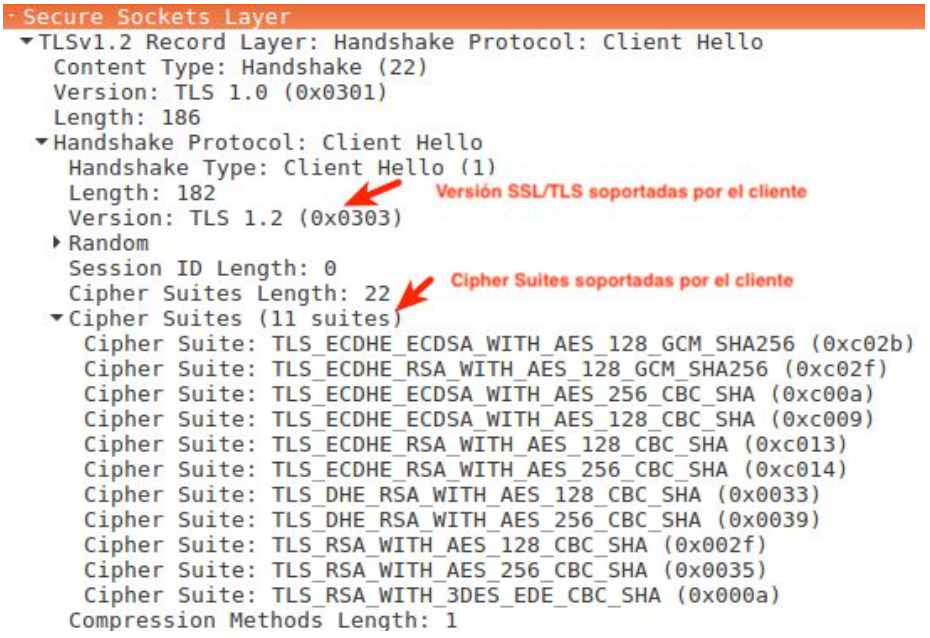
La **autenticación** es la parte más costosa, aunque también es esencial para la seguridad de las comunicaciones. Esto es debido a que la confidencialidad y la integridad no valen nada sin la autenticación. Si no hay forma de asegurarse de que alguien es en realidad quien dice ser, entonces podríamos cifrar y validar una conexión falsa.

Por así decirlo, sin autenticación, estaríamos abiertos a ataques MitM (Man in the Middle), por ejemplo. Para autenticar un servidor e intercambiar credenciales con él utilizamos certificados TLS. Los certificados son solo una clave pública con una gran cantidad de información adjunta, como el FQDN siendo autenticado, fechas de "emisión" y "caducidad", entre otras cosas. El servidor almacena y mantiene en secreto la clave privada correspondiente. TLS usa estas claves para autenticar el servidor ante el cliente (un cliente también puede usar TLS para autenticarse en un servidor).

Como bien sabemos, los mensajes cifrados con la clave pública solo se pueden descifrar utilizando la clave privada, pero los mensajes cifrados con la clave privada se pueden descifrar con cualquiera de estas. El propietario de las claves mantiene la clave privada en secreto y distribuye la clave pública libremente. Este primer paso de negociación y autenticación se realiza mediante criptografía asimétrica o de clave pública (por la que no hace falta que exista un canal seguro). Una vez acordada la clave secreta, ambos utilizan dicha clave para cifrar y descifrar los mensajes, estableciendo una conexión donde los datos viajan cifrados mediante criptografía simétrica.

3.2 Funcionamiento y fases

En la fase de Handshake, el cliente envía un mensaje Client Hello, en el que le indica al servidor la versión del protocolo TLS, los algoritmos de cifrado, los métodos de compresión que soporta, etc. El servidor lo compara con su listado y entre ambos negocian los algoritmos que utilizarán durante la comunicación.



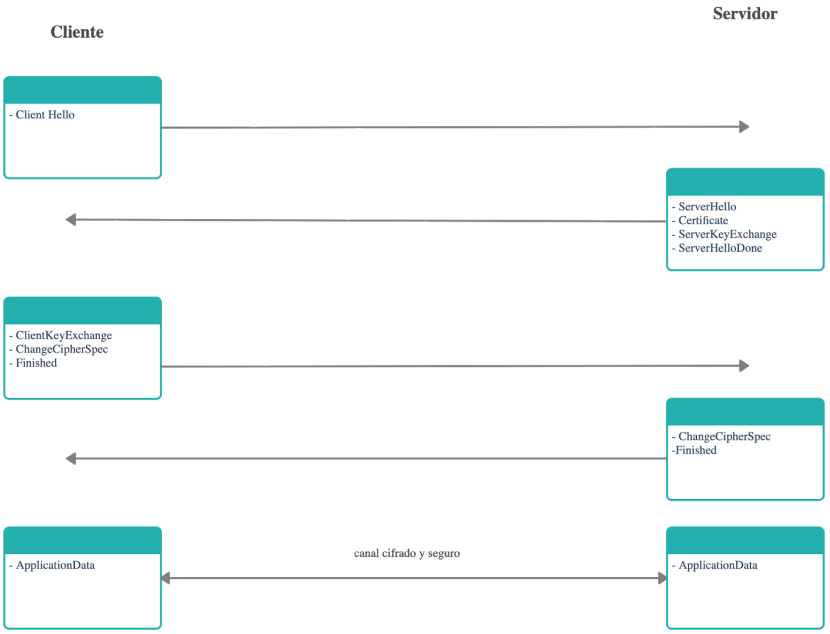
Si llegamos a analizar algún mensaje Client Hello, con herramientas como Wireshark, podremos ver que Cipher Suites se refiere a la lista de protocolos soportados por el cliente.

La capa donde opera el protocolo TLS se encuentra entre la **capa de aplicación** (donde operan los protocolos como HTTP, SMTP, etc) y entre la capa de transporte. TLS se suele implementar sobre el protocolo de transporte TCP, aunque actualmente también se puede implementar sobre UDP.

Para operar, el protocolo TLS encripta la información para protegerla al emitirla por la red. Para encriptar esta información, el protocolo TLS se divide en dos fases diferentes, por un lado se utiliza el protocolo de de mutuo acuerdo (**TLS Handshake Protocol**) que negocia los parámetros de· seguridad mediante los cuales se van a comunicar ambos extremos, y por otro lado el protocolo de registro (**TLS Record Protocol**) que especifica la forma de encapsular los datos a emitir (incluyendo los parámetros de configuración acordados).

La capa donde actúa el protocolo handshake se encuentra ubicada en medio de la capa donde opera el protocolo de registro y la capa aplicación. La capa donde opera el protocolo de registro está ubicada sobre la capa de transporte y acondiciona los mensajes que recibe del handshake y los envía.

El protocolo handshake o de negociación inicia una fase de negociación siguiendo un orden determinado:



- Primero, el cliente es quien envía el mensaje **ClientHello**, de esta manera especifica la información de cifrado, la versión del protocolo TLS/SSL, la lista de algoritmos de cifrado y los métodos de compresión que soporta. En el caso en el que se intente restablecer a una sesión anterior, se puede enviar el identificador de sesión, con esto se consigue que el proceso de negociación sea más corto puesto que el proceso de negociación ya se ha realizado con anterioridad.
- Después el servidor realiza una respuesta al cliente con un mensaje **ServerHello** indicando todos los parámetros mencionados anteriormente y una cadena de 32 bytes aleatorios. El protocolo TLS/SSL escogido debe ser el más actual que soporten tanto cliente como servidor. Con esto el cliente puede verificar la integridad del servidor. El servidor envía su mensaje **ServerKeyExchange**, el cual incluye la clave pública y la clave del certificado. El servidor solicita mediante el mensaje Certificate Request el certificado del cliente, si lo tiene, para que la conexión pueda ser mutuamente autenticada. El servidor envía un mensaje **ServerHelloDone**, dando por concluida la fase de negociación asimétrica.
- Luego, el cliente responde con un mensaje con su certificado y una vez ha comprobado y validado el certificado responde con un mensaje **ClientKeyExchange** que contiene la **PreMasterSecret** cifrada con la clave pública del servidor. Con esto hemos concluido en que el cliente y servidor usan las cadenas de números aleatorios y la **PreMasterSecret** para generar la clave de sesión secreta con la que ambos cifrarán y descifrarán la información enviada.
- Finalmente, una vez que se termine la negociación el cliente envía al servidor un registro **ChangeCipherSpec** y mensaje cifrado Finished. Con esto indicando que todo los mensajes enviados a partir de este momento estarán cifrados con el cifrado acordado previamente. con Finished da por finalizada la fase de negociación asimétrica (mensaje que incluye un hash y un MAC de todos los handshake anteriores, para la integridad). El servidor hace una operación con la misma funcionalidad.

Con el protocolo de negociación o handshake terminado, se consigue crear un canal entre ambos bajo el protocolo TLS/SSL escogido, garantizando que todos los datos que viajen por este canal irán cifrados. La **ventaja** que tiene TLS es que utiliza la criptografía asimétrica de clave pública y esto nos aporta fiabilidad, ya que permite a ambos extremos de la comunicación poder negociar una clave secreta compartida sin tener que hacerlo a través de un canal encriptado ni de conocerse previamente entre ambos. Una vez acordada la clave secreta compartida se utiliza criptografía simétrica (más rápida y menos costosa). La **desventaja** es que son necesarios varios mensajes en cada sentido de la comunicación y esto agrega mayor latencia a las conexiones TLS.

Ahora explicaremos por medio de un ejemplo la forma habitual de autenticar a alguien mediante la criptografía de la clave pública (suponemos que Mallory quiere autenticar a Alice):

1. Mallory envía a Alice un mensaje aleatorio encriptado con la clave pública de Alice.
2. Alice descifra el mensaje con su clave privada y lo envía de vuelta.
3. Mallory compara el mensaje de Alice con el aleatorio que cifró usando la clave pública de Alice, si coinciden, entonces Alice es quien dice ser, porque solo ella pudo haber descifrado el mensaje aleatorio, porque solo ella tiene la clave privada correspondiente.

TLS usa una variación de esta técnica, pero es esencialmente la misma. Ahora bien, incluso con este truco, la validación de un certificado no es sencilla. Ahora vamos a ver 3 soluciones algo dudosas, pero que nos ayudarán a ver cómo funciona realmente el protocolo.

Primera posible solución:

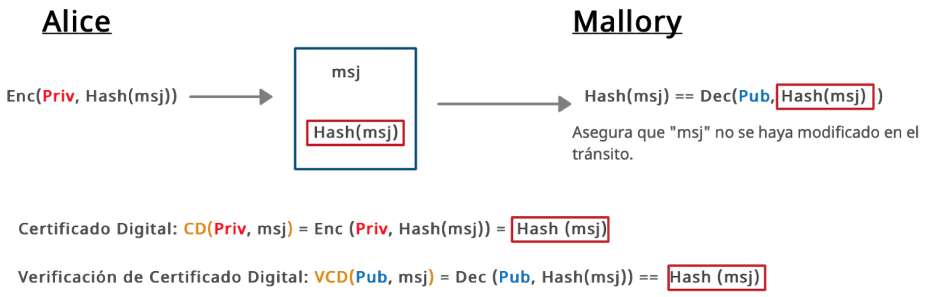
1. Se crea un hash del certificado, se cifra con la clave privada y luego se agrega al certificado (un certificado es solo la clave pública correspondiente con un montón de metadatos adjuntos).
2. El servidor envía el certificado a los clientes que se conectan a él.
3. Para verificar el certificado el cliente descifra el hash usando la clave pública del certificado, luego calcula su propio hash y los compara, si son iguales el certificado es válido.
4. Luego envía un mensaje aleatorio al servidor encriptado con la clave pública proporcionada, si el servidor envía el mensaje original no encriptado, entonces se considera autenticado.

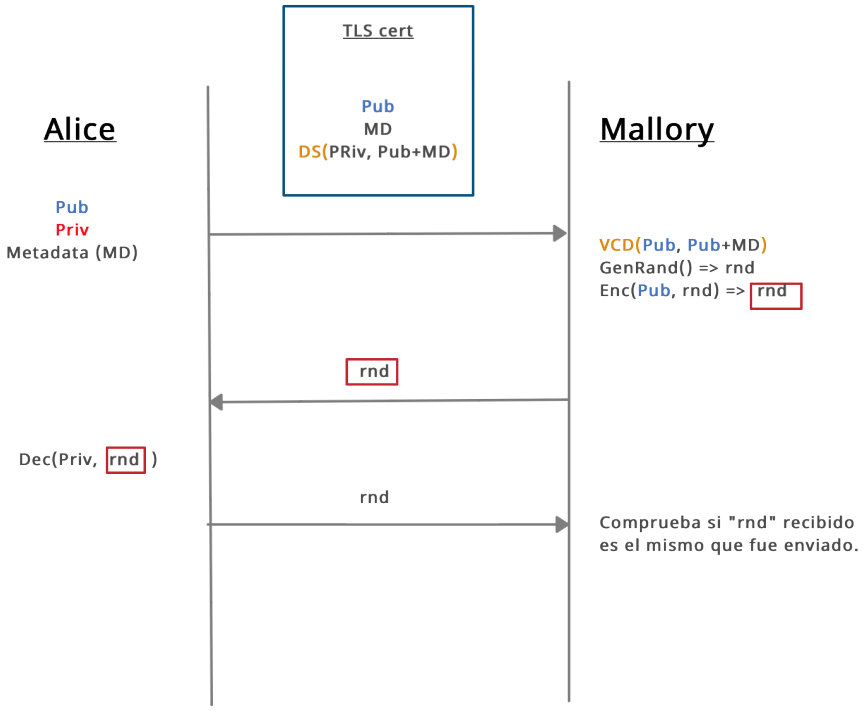
Este proceso asegura que:

- La clave pública proporcionada corresponde a la clave privada utilizada para cifrar el hash del certificado.
- El servidor tiene acceso a la clave privada.

Como bien sabemos el hash cifrado creado adjunto al certificado se llama firma digital. En este ejemplo, el servidor ha firmado digitalmente su propio certificado, esto se denomina certificado autofirmado. Ahora bien, este esquema no se autentica en absoluto. Si un atacante logra interceptar las comunicaciones o desviar el tráfico, puede reemplazar la clave pública del certificado con la suya, rehacer la firma digital con su propia clave privada y enviarla al cliente.

El problema radica en que toda la información necesaria para la verificación la proporciona el servidor, por lo que lo único de lo que se puede estar seguro es que la parte con la que estamos hablando tiene la clave privada correspondiente a la clave pública que él mismo había previsto.
Esta es la razón por la que cuando se conecta a una entidad con un certificado autofirmado, los navegadores le darán una advertencia, no pueden garantizar que quienquiera que se esté comunicando sea quien dice ser.
Sin embargo, este tipo de certificados son muy útiles en algunos escenarios. Se pueden crear de forma gratuita, rápida y sin complicaciones. Por lo tanto, son buenos para algunas comunicaciones internas y creación de prototipos.





Segunda posible solución:

Bueno, dado que el problema es que el servidor proporciona toda la información para autenticar el certificado, ¿por qué no trasladamos parte de esa información al cliente? Si conseguimos copiar al cliente una copia del certificado, por medio de una USB por ejemplo. Más tarde, cuando el servidor envía su certificado:

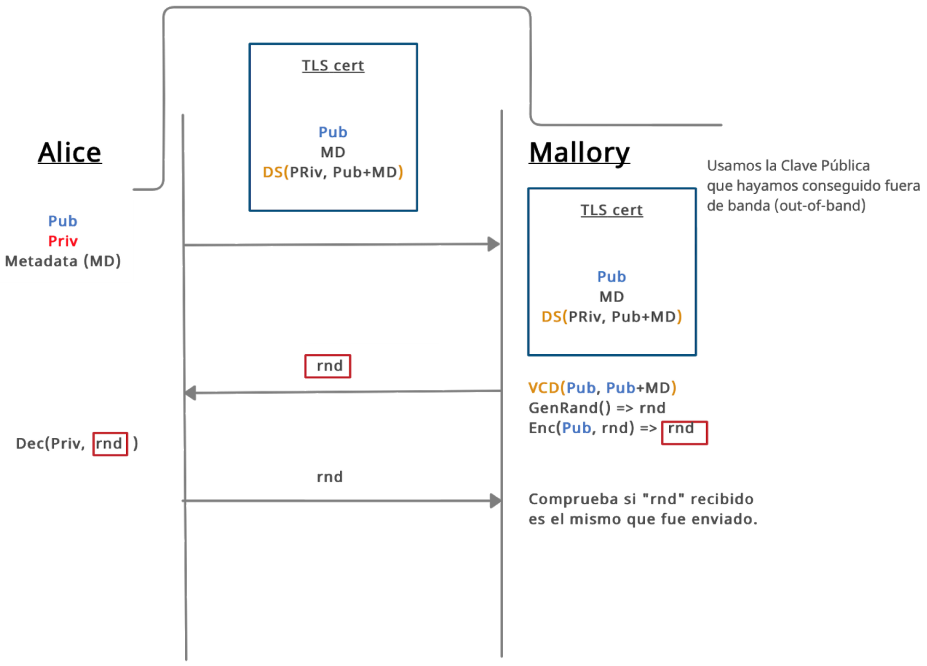
1. El cliente descifra el hash cifrado del certificado (la firma digital) con la clave pública de la copia del certificado que se proporcionó anteriormente en la unidad USB.
2. Luego genera su propio hash del certificado y lo compara con la versión descifrada. De esta forma se asegura que el certificado no haya sido alterado.
3. Luego envía un mensaje aleatorio al servidor encriptado con la clave pública. Si el servidor devuelve el mensaje original sin cifrar, lo consideramos autenticado.

Si alguien intercepta o desvía la conexión, no tiene posibilidades acertar y proporcionar un certificado válido, sin la clave privada es computacionalmente inviable. Este proceso con la unidad USB se denomina proceso fuera de banda.

Ahora bien, esta solución realmente cumple el proceso de autenticación y, a veces, se utiliza para comunicaciones internas. Sin embargo, no es muy práctico para más de varios casos de uso. Sería engorroso tener que descargar un certificado cada vez que necesitemos acceder a un nuevo sitio web seguro como un banco o una tienda electrónica, o peor aún, esperar a que alguien venga con un USB hasta nosotros.

Además, si tuviéramos que descargar el certificado a través de la red, tendría que asegurarse de que el certificado no sea manipulado en el camino, que es uno de los problemas que TLS debería resolver para nosotros en primer lugar. También existe el problema de qué hacer cuando el certificado caduca o cómo revocarlo si la clave privada se ve comprometida.

De esta manera entra en juego la última solución posible. Lo vamos a explicar de una simple manera.



Tercera solución:

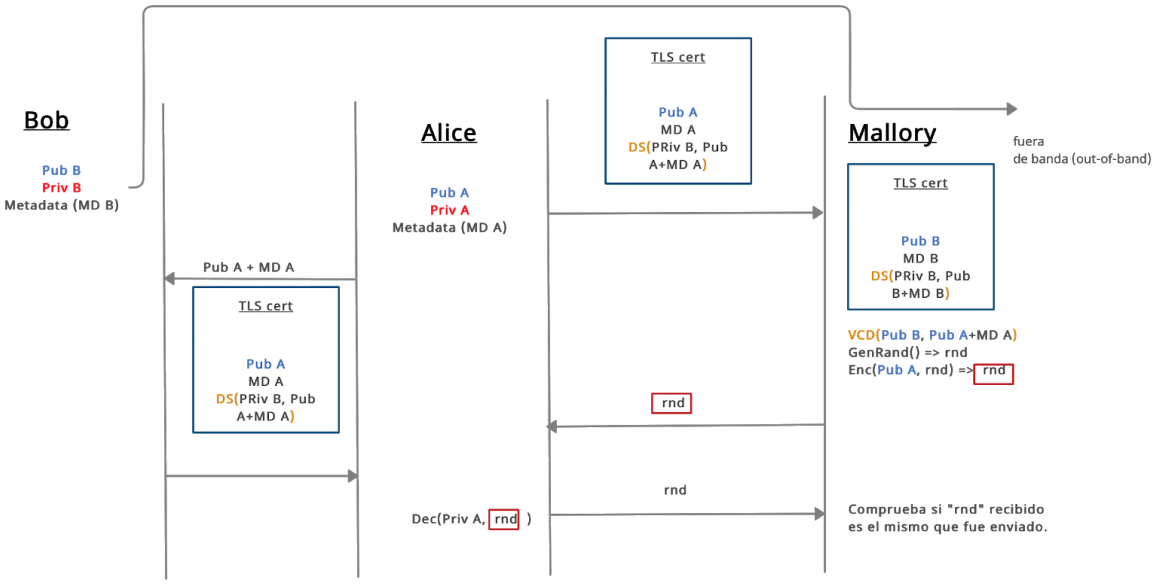
Pongamos un ejemplo con Bob, es un miembro muy conocido de la comunidad, un compañero verdaderamente responsable y leal hasta el extremo, y se le ocurre un negocio. Creará un certificado autofirmado y se lo dará gratuitamente a todo el mundo mediante un proceso fuera de banda (digamos, durante una barbacoa gratuita en todo el vecindario), luego le cobrará una tarifa por firmar digitalmente su certificado usando su clave privada.

Hasta este momento todos los certificados estaban autofirmados, esta vez es Bob quien los firmará, ahora estamos en una situación en la que:

- Todos tienen el certificado que Bob tan generosamente les dio.
- Todo el mundo confía en Bob (es muy buena gente).

Por lo tanto, cualquier persona que tenga su certificado firmado con la clave privada de Bob puede hacer que lo autentique cualquiera que tenga el certificado de Bob (porque el certificado de Bob tiene su clave pública). De esta manera, Bob se asegurará de que nadie se haga pasar por nadie. Si recibe un correo electrónico diciendo que desea que se firme cierto certificado para su empresa, Bob irá personalmente a su casa y se asegurará de que sea su certificado, que sea su empresa y que usted envió ese correo electrónico.

Cuando el certificado de Bob esté a punto de caducar, se asegurará de darte uno nuevo. Si alguien quiere revocar su certificado, puede decirle a Bob que lo ponga en su lista de certificados revocados. Cuando alguien intenta autenticar un certificado firmado por Bob, lo llamarán para verificar si ese certificado no ha sido revocado.



Bueno, así es como funciona el mundo real, con algunos cambios importantes:

- Bob es un CA (Autoridad de Certificación) como DigiCert, Sectigo, Symantec, IdenTrust, GoDaddy, GlobalSign, etc (a veces no son tan buena gente como Bob).
- El proceso fuera de banda no es una barbacoa. Sus certificados ya vienen incluidos con su sistema operativo y navegador (lo podemos ver en /etc/ssl/certs si está en Linux).
- Los mecanismos de revocación se denominan CRL y OCSP (que se suponía que reemplazaban a CRL). Todo el sistema de revocación de certificados es un desastre en este momento (ya que funciona por medio de una lista CRL o comprobación de estados OCSP). Otro sistema para verificar la validez de los certificados que funciona en conjunto con CRL / OCSP es la transparencia de certificados.
- Hay que “confiar” en ellos.

Sin embargo, la mayoría de ellos nos cobran. Dependiendo del tipo de certificado los precios varían . Aunque podemos buscar en LetsEncrypt una alternativa gratuita muy sólida y popular.

En resumen esto es un proceso en el que llegamos a confiar en un tercero, tanto el servidor como el cliente firma el certificado y crea una cadena de confianza. Con TLS, creamos cadenas de confianza utilizando CA como nuestros terceros.

Hay que tener en cuenta que nuestros datos no tendrán un cifrado más fuerte o más débil dependiendo de cuánto se pague, todas las conexiones TLS usan alguna forma de AES, qué tan fuerte depende de lo que el cliente y el servidor puedan y estén dispuestos a manejar o usar. Por ejemplo, muchos servidores se niegan a usar SSLv3 desde todo el escándalo de POODLE, algunos clientes antiguos no admiten los algoritmos de cifrado más nuevos (como por ejemplo IE), ya que algunos servidores no se han actualizado para usar los algoritmos de cifrado más nuevos.

Las CA generalmente nos cobran más por los certificados que se utilizarán en varias máquinas, por lo que, por ejemplo, un certificado para *.ugr.com costará más que uno para www.ugr.com. Sin embargo, lo que realmente venden las CA no son certificados, es confianza.