# Introducción a SSL con ejemplo de transacción y intercambio de paquetes

## Contenido



## Introducción

Este documento describe los conceptos básicos del protocolo Secure Sockets Layer (SSL) y proporciona una transacción de ejemplo y captura de paquetes.

## Descripción general del registro SSL

La unidad básica de datos en SSL es un registro. Cada registro consta de un encabezado de registro de cinco bytes, seguido de datos.

### Formato de grabación

- **Tipo:** uint8 - valores enumerados
- **Versión:** uint16
- **Longitud:** uint16

| Tipo | Versión | | Longitud | |
|---|---|---|---|---|
| T | VH | VL | LH | LL |

### Tipo de registro

Hay cuatro tipos de registro en SSL:

- **Apretón de manos** (22, 0x16)
- **Cambiar la especificación del cifrado** (20 x 14)
- **Alerta** (21, 0x15)
- **Datos de la aplicación** (23 x 17)

### Grabar versión

La versión de registro es un valor de 16 bits y se formatea en orden de red.

> **Nota:** Para SSL versión 3 (SSLv3), la versión es 0x0300. Para Transport Layer Security Version 1 (TLSv1), la versión es 0x0301. Cisco Adaptive Security Appliance (ASA) no admite SSL versión 2 (SSLv2), que utiliza la versión 0x0002, ni ninguna versión de TLS mayor que TLSv1.

### Longitud del registro

La longitud del registro es un valor de 16 bytes y se formatea en orden de red.

En teoría, esto significa que un único registro puede tener hasta 65,535 (2^16 -1) bytes de longitud. El RFC2246 de TLSv1 establece que la longitud máxima es de 16.383 bytes (2^14 -1). Se sabe que los productos de Microsoft (Microsoft Internet Explorer e Internet Information Services) exceden estos límites.

## Tipos de registros

Esta sección describe los cuatro tipos de registros SSL.

### Registros de intercambio de señales

Los registros de intercambio de señales contienen un conjunto de mensajes que se utilizan para el intercambio de señales. Estos son los mensajes y sus valores:

- **Solicitud de saludo** (0, 0x00)
- **Saludo del cliente** (1, 0x01)
- **Server Hello** (2, 0x02)
- **Certificado** (11, 0x0B)
- **Intercambio de claves de servidor** (12, 0x0C)
- **Solicitud de certificado** (13, 0x0D)
- **Servidor Hello Finalizado** (14, 0x0E)
- **Verificación de certificado** (15, 0x0F)
- **Intercambio de claves de cliente** (16, 0x10)
- **Finalizado** (20 x 14)

En el caso simple, los registros de entrada en contacto no se cifran. Sin embargo, un registro de entrada en contacto que contiene un mensaje finalizado siempre se cifra, ya que siempre ocurre después de un registro Change Cipher espec (CCS).

### Registros de CCS

Los registros CCS se utilizan para indicar un cambio en los cifrados criptográficos. Inmediatamente después del registro de CCS, todos los datos se cifran con el nuevo cifrado. Los registros de CCS pueden o no cifrarse; en una conexión simple con un solo intercambio de señales, el registro CCS no está cifrado.

### Registros de alertas

Los registros de alerta se utilizan para indicar al par que se ha producido una condición. Algunas alertas son advertencias, mientras que otras son fatales y provocan que la conexión falle. Las alertas pueden cifrarse o no, y pueden ocurrir durante un intercambio de señales o durante la transferencia de datos. Hay dos tipos de alertas:

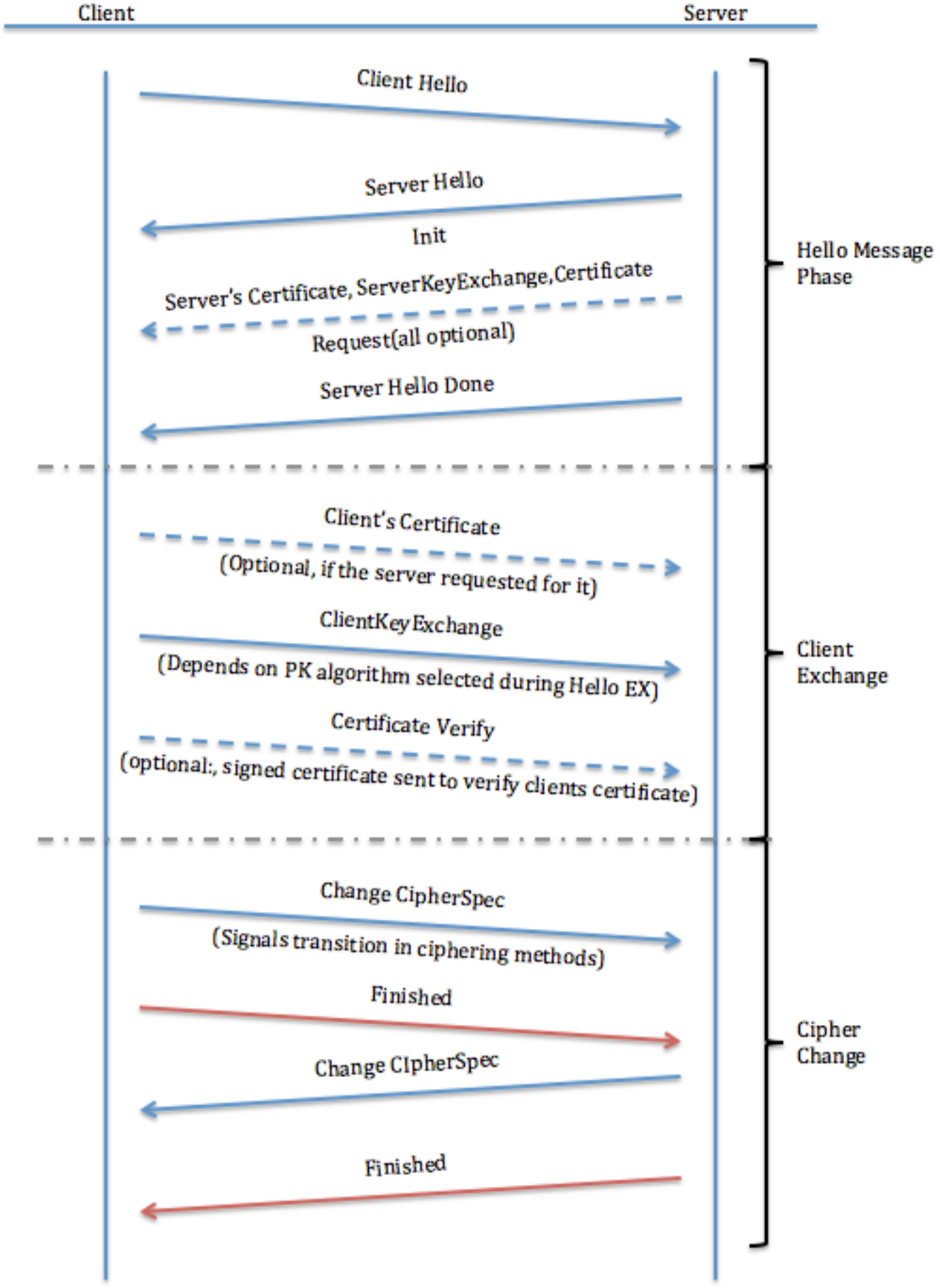
- **Alertas de cierre:** La conexión entre el cliente y el servidor se debe cerrar correctamente para evitar cualquier tipo de ataques de truncamiento. Se envía un mensaje **close_notify que indica al destinatario que el remitente no enviará más mensajes sobre esa conexión.**
- **Alertas de error:** Cuando se detecta un error, la parte que realiza la detección envía un mensaje a la otra parte. Tras la transmisión o recepción de un mensaje de alerta fatal, ambas partes cierran inmediatamente la conexión. Algunos ejemplos de alertas de error son:
  - **mensaje_inesperado** (fatal)
  - **decompression_failure**
  - **handshake_failure**

### Registro de datos de la aplicación

Estos registros contienen los datos reales de la aplicación. Estos mensajes son transportados por la capa de registro y están fragmentados, comprimidos y cifrados, según el estado de conexión actual.

## Ejemplo de transacción

Esta sección describe una transacción de ejemplo entre el cliente y el servidor.




El intercambio Hello

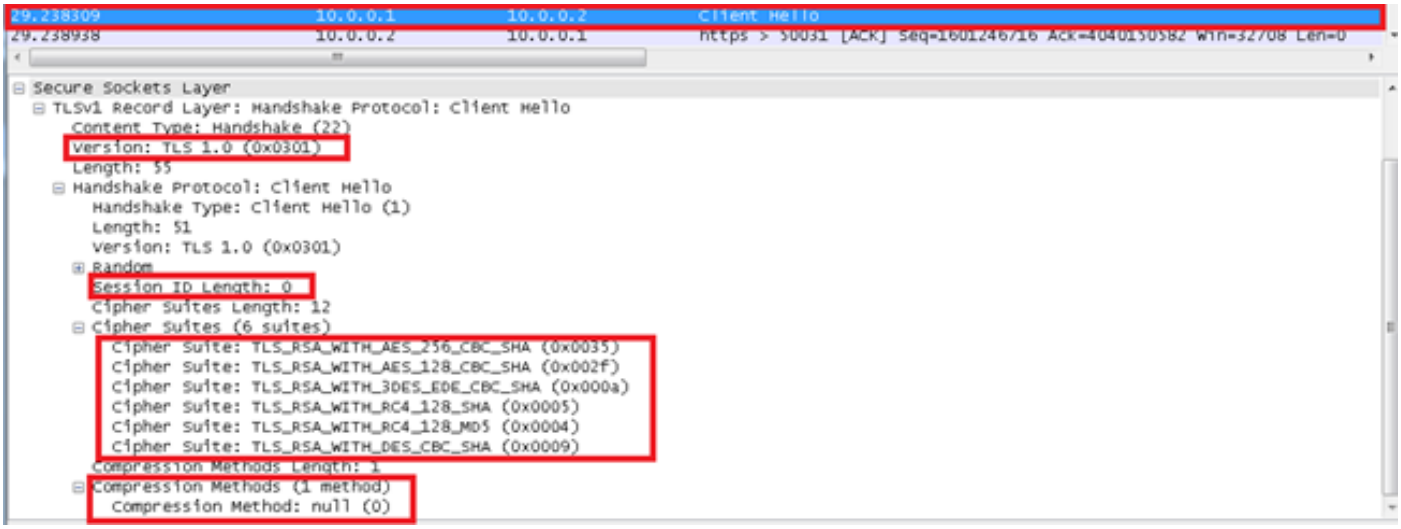
Cuando un cliente SSL y un servidor comienzan a comunicarse, acuerdan una versión de protocolo, seleccionan algoritmos criptográficos, opcionalmente se autentican mutuamente y utilizan técnicas de cifrado de clave pública para generar secretos compartidos. Estos procesos se realizan en el protocolo de intercambio de señales. En resumen, el cliente envía un mensaje de saludo del cliente al servidor, que debe responder con un mensaje de saludo del servidor o se produce un error fatal y la conexión falla. El cliente Hello y el servidor Hello se utilizan para establecer las capacidades de mejora de la seguridad entre el cliente y el servidor.

### Hola de cliente

Client Hello envía estos atributos al servidor:

- **Versión del protocolo:** Versión del protocolo SSL por el cual el cliente desea comunicarse durante esta sesión.
- **ID de Sesión:** ID de una sesión que el cliente desea utilizar para esta conexión. En el primer cliente Hello del intercambio, el ID de sesión está vacío (consulte la captura de pantalla de captura de paquetes después de la nota).
- **Suite Cipher:** Esto se pasa del cliente al servidor en el mensaje de saludo del cliente. Contiene las combinaciones de algoritmos criptográficos soportados por el cliente en orden de preferencia del cliente (primera opción). Cada conjunto de cifrado define un algoritmo de intercambio de claves y una especificación de cifrado. El servidor selecciona un conjunto de claves o, si no se presentan opciones aceptables, devuelve una alerta de error de entrada en contacto y cierra la conexión.
- **Método de compresión:** Incluye una lista de algoritmos de compresión soportados por el cliente. Si el servidor no admite ningún método enviado por el cliente, la conexión falla. El método de compresión también puede ser nulo.

**Nota:** La dirección IP del servidor en las capturas es 10.0.0.2 y la dirección IP del cliente es 10.0.0.1.



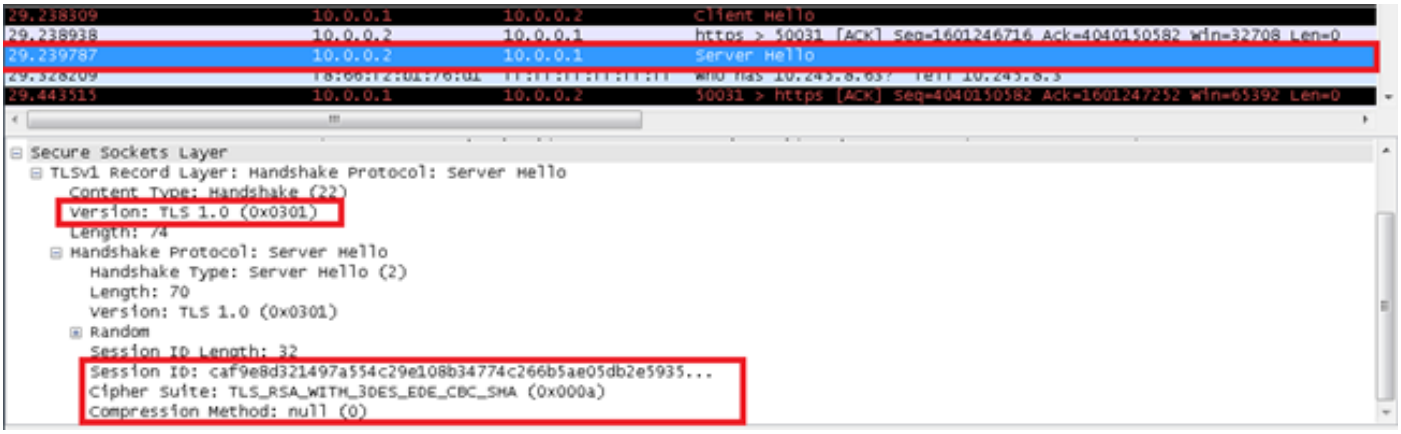

### Hola de servidor

El servidor devuelve estos atributos al cliente:

- **Versión del protocolo:** La versión elegida del protocolo SSL que soporta el cliente.
- **ID de Sesión:** Esta es la identidad de la sesión que corresponde a esta conexión. Si el ID de

sesión enviado por el cliente en el Hello de cliente no está vacío, el servidor busca una coincidencia en la memoria caché de sesión. Si se encuentra una coincidencia y el servidor está dispuesto a establecer la nueva conexión usando el estado de sesión especificado, el servidor responde con el mismo valor que fue proporcionado por el cliente. Esto indica una continuación del período de sesiones y exige que las partes procedan directamente a los mensajes terminados. De lo contrario, este campo contiene un valor diferente que identifica la nueva sesión. El servidor podría devolver un **session_id** vacío para indicar que la sesión no se almacenará en caché y, por lo tanto, no se puede reanudar.

- **Suite Cipher:** Según lo seleccionado por el servidor de la lista que se envió desde el cliente.
- **Método de compresión:** Según lo seleccionado por el servidor de la lista que se envió desde el cliente.
- **Solicitud de certificado:** El servidor envía al cliente una lista de todos los certificados configurados en ella y permite al cliente seleccionar el certificado que desea utilizar para la autenticación.

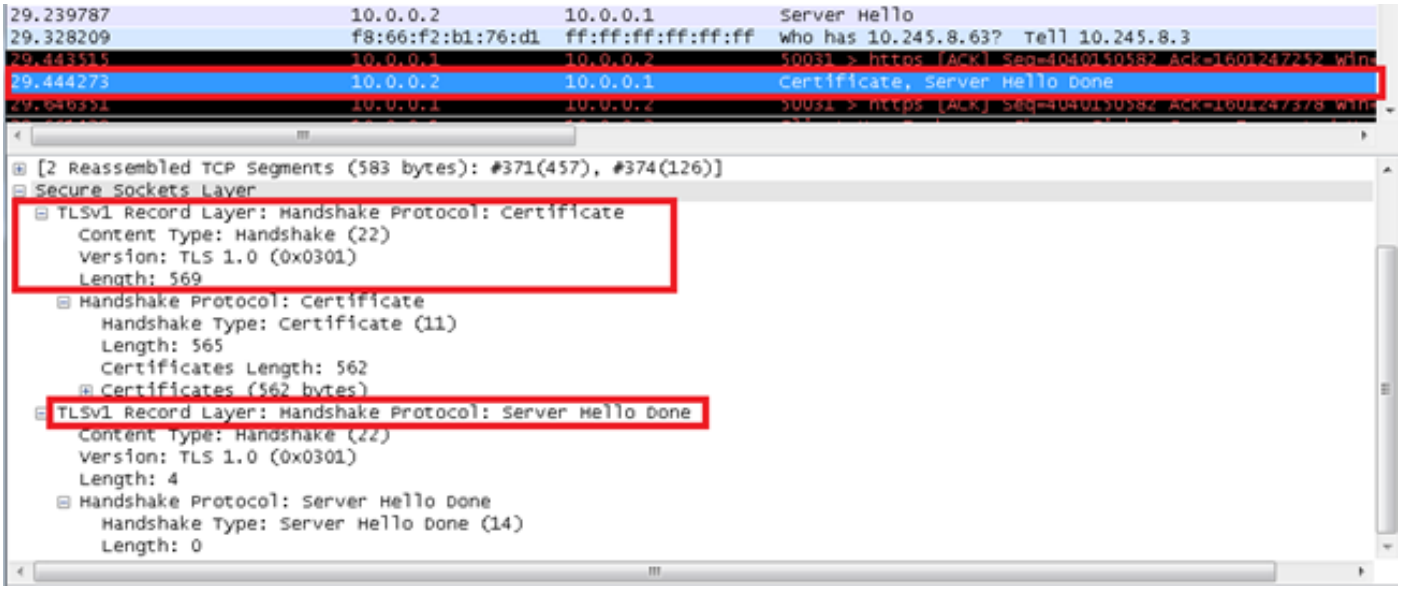



Para solicitudes de reanudación de sesión SSL:

- El servidor también puede enviar una solicitud Hello al cliente. Esto es sólo para recordar al cliente que debe iniciar la renegociación con una solicitud de saludo de cliente cuando sea conveniente. El cliente ignora la solicitud Hello del servidor si el proceso de intercambio de señales ya está en marcha.
- Los mensajes de intercambio de señales tienen más prioridad sobre la transmisión de los datos de la aplicación. La renegociación debe comenzar en no más de una o dos veces el tiempo de transmisión de un mensaje de datos de aplicación de longitud máxima.

### Servidor Hello Finalizado

El servidor envía el mensaje Hello Done del servidor para indicar el final del servidor hello y los mensajes asociados. Después de enviar este mensaje, el servidor espera una respuesta del cliente. Cuando recibe el mensaje Server Hello Done, el cliente verifica que el servidor haya proporcionado un certificado válido, si es necesario, y verifica que los parámetros Server Hello sean aceptables.




### Certificado de servidor, intercambio de claves de servidor y solicitud de certificado (opcional)

- **Certificado de servidor:** Si el servidor debe autenticarse (que suele ser el caso), el servidor envía su certificado inmediatamente después del mensaje Server Hello. El tipo de certificado debe ser apropiado para el algoritmo de intercambio de claves del conjunto de claves de cifrado seleccionado y generalmente es un certificado X.509.v3.
- **Server Key Exchange:** El servidor envía el mensaje Server Key Exchange si no tiene certificado. Si los parámetros Diffie-Hellman(DH) se incluyen con el certificado del servidor, no se utiliza este mensaje.
- **Solicitud de certificado:** Un servidor puede solicitar opcionalmente un certificado del cliente, si procede, para el conjunto de cifrado seleccionado.

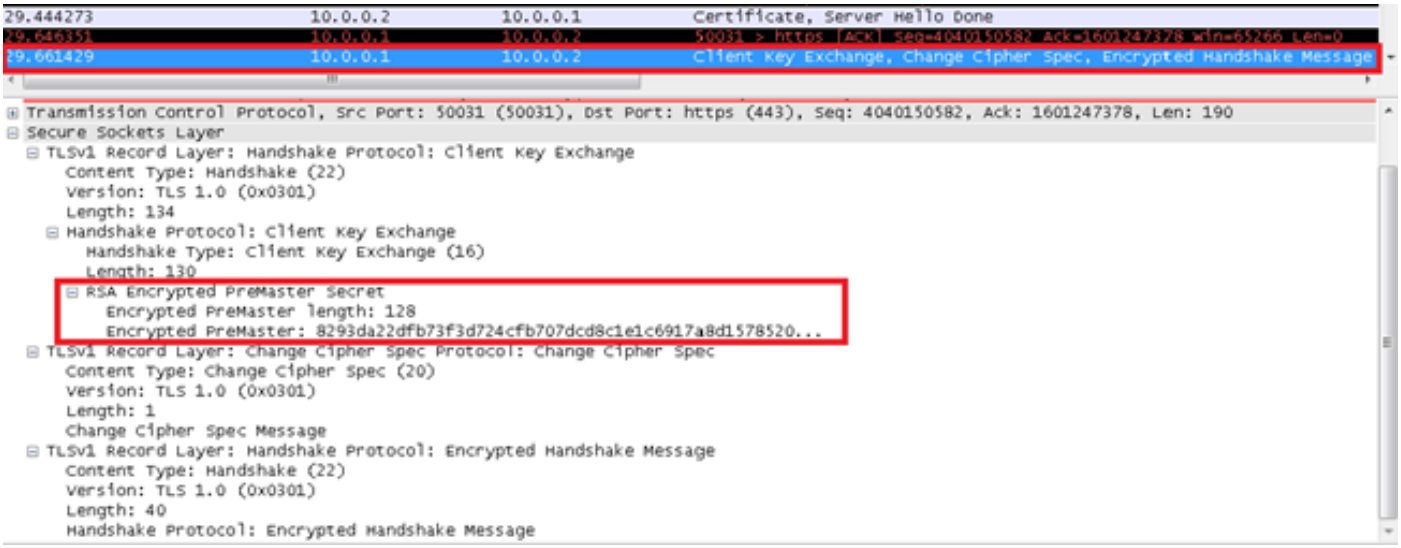
## Intercambio de clientes

### Certificado de cliente (opcional)

Este es el primer mensaje que el cliente envía después de recibir un mensaje de Server Hello Done. Este mensaje sólo se envía si el servidor solicita un certificado. Si no hay ningún certificado adecuado disponible, el cliente envía una alerta **no_certificate** en su lugar. Esta alerta es sólo una advertencia; sin embargo, el servidor podría responder con una alerta de falla de entrada en contacto fatal si se requiere autenticación del cliente. Los certificados DH del cliente deben coincidir con los parámetros DH especificados por el servidor.

### Intercambio de claves de cliente

El contenido de este mensaje depende del algoritmo de clave pública seleccionado entre los mensajes Client Hello y Server Hello. El cliente utiliza una clave premaster cifrada por el algoritmo Rivest-Shamir-Addleman (RSA) o DH para el acuerdo de clave y la autenticación. Cuando se utiliza RSA para la autenticación del servidor y el intercambio de claves, el cliente genera un **pre_master_secret de** 48 bytes, se cifra bajo la clave pública del servidor y se envía al servidor. El servidor utiliza la clave privada para descifrar el **pre_master_secret**. Ambas partes convierten el **pre_master_secret** en el **master_secret**.




### Verificación de certificado (opcional)

Si el cliente envía un certificado con capacidad de firma, se envía un mensaje de verificación de certificado firmado digitalmente para verificar explícitamente el certificado.

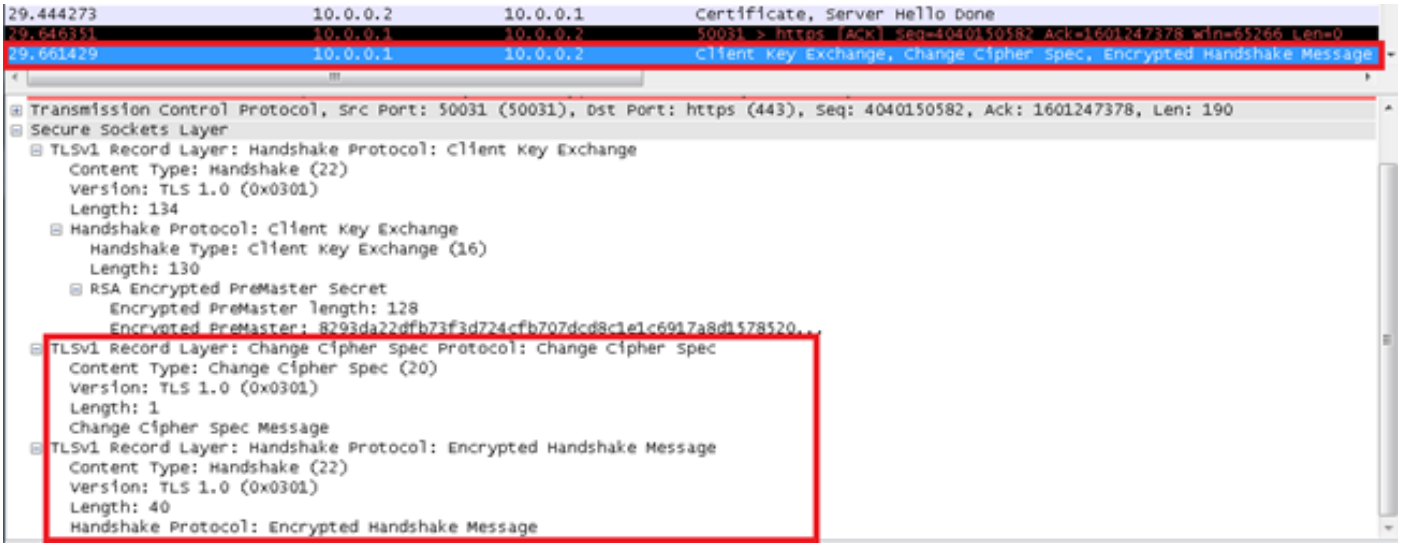
## Cambio de cifrado

### Cambio de los mensajes de especificación del cifrado

El cliente envía el mensaje Cambiar especificación del cifrado y el cliente copia la especificación del cifrado pendiente (la nueva) en la especificación del cifrado actual (la que se utilizó anteriormente). El protocolo de especificación del cifrado de cambio existe para indicar transiciones en las estrategias de cifrado. El protocolo consta de un único mensaje, que se cifra y comprime bajo la especificación actual (no la pendiente) del Cipher. Tanto el cliente como el servidor envían el mensaje para notificar a la parte receptora que los registros subsiguientes están protegidos bajo la especificación y las claves del Cipher negociadas más recientemente. La recepción de este mensaje hace que el receptor copie el estado pendiente de lectura en el estado actual de lectura. El cliente envía un mensaje Change Cipher espec después de los mensajes de intercambio de claves de entrada en contacto y verificación de certificados (si existe), y el servidor envía uno después de procesar correctamente el mensaje de intercambio de claves que recibió del cliente. Cuando se reanuda una sesión anterior, se envía el mensaje Cambiar especificación del cifrado después de los mensajes Hello. En las capturas, los mensajes Client Exchange, Change Cipher y Finished se envían como un único mensaje del cliente.

### Mensajes finalizados

Siempre se envía un mensaje Finalizado inmediatamente después de un mensaje Change Cipher Specification para verificar que los procesos de intercambio de claves y autenticación fueron correctos. El mensaje Finalizado es el primer paquete protegido con los algoritmos, claves y secretos negociados más recientemente. No se requiere reconocimiento del mensaje Finalizado; las partes pueden comenzar a enviar datos cifrados inmediatamente después de enviar el mensaje Finalizado. Los destinatarios de los mensajes Finalizados deben verificar que el contenido es correcto.




# Información Relacionada

- RFC 6101: protocolo de capa de sockets seguros versión 3.0
- Wireshark SSL wiki: descifrar paquetes SSL con Wireshark
- Soporte Técnico y Documentación - Cisco Systems