与示例处理和信息包交换的SSL介绍

目录

简介

本文描述安全套接字协议层(SSL)协议基本概念，并且提供示例处理和数据包捕获。

SSL记录概述

数据基本单元在SSL的是记录。每个记录包括一五字节记录首标，跟随由数据。

记录格式

- **类型**：uint8 -列出的值
- **版本**：uint16
- **长度**：uint16

类型 version 长度
T VH VL LH LL

记录类型

有在SSL的四种记录类型：

- **握手**(22， 0x16)
- **崔凡吉莱密码器Spec** (20， 0x14)
- **警报**(21， 0x15)
- **应用程序数据**(23， 0x17)

记录版本

记录版本是16字节值和按顺序网络秩序被格式化。

> **Note**:对于SSL版本3 (SSLv3)，版本是0x0300。对于传输层安全版本1 (TLSv1)，版本是0x0301。Cisco可适应安全工具(ASA)比TLSv1不支持SSL版本2 (SSLv2)，使用版本0x0002，或者TLS任何版本极大。

记录长度

记录长度是16字节值和按顺序网络秩序被格式化。

在理论上，这意味着单个记录可以是长度65,535个(2^16 -1)字节。TLSv1 RFC2246阐明，最大长度是16,383个(2^14 -1)字节。Microsoft产品(微软Internet Explorer和互联网信息服务)知道超过这些限额。

记录的类型

此部分描述SSL记录的四种类型。

握手记录

握手记录包含是被使用的为了握手的一套消息。这些是消息和他们的值：

- **Hello请求**(0， 0x00)
- **客户端Hello** (1， 0x01)
- **服务器问候**(2， 0x02)
- **证书**(11， 0x0B)
- **服务器密钥Exchange** (12， 0x0C)
- **证书请求**(13， 0x0D)
- **执行的服务器问候**(14， 0x0E)
- **Certificate verify** (15， 0x0F)
- **客户端密钥交换**(16， 0x10)
- **已完成**(20， 0x14)

在简单案件中，握手记录没有加密。然而，包含一个已完成消息的握手记录总是加密，因为总是发生，在崔凡吉莱密码器Spec (CCS)后记录。

CCS记录

CCS记录用于为了指示在密码密码器上的一个变化。在CCS记录，所有数据用新的密码器之后加密。CCS记录可能或也许不加密;在与单个握手的一简单连接， CCS记录没有加密。

提醒的记录

提醒的记录用于为了表明对对等体情况发生。而其他致命并且造成连接发生故障，一些警报是警告。在数据传输期间，警报可能或也许不加密，并且也许发生在握手期间或。有警报的两种类型：

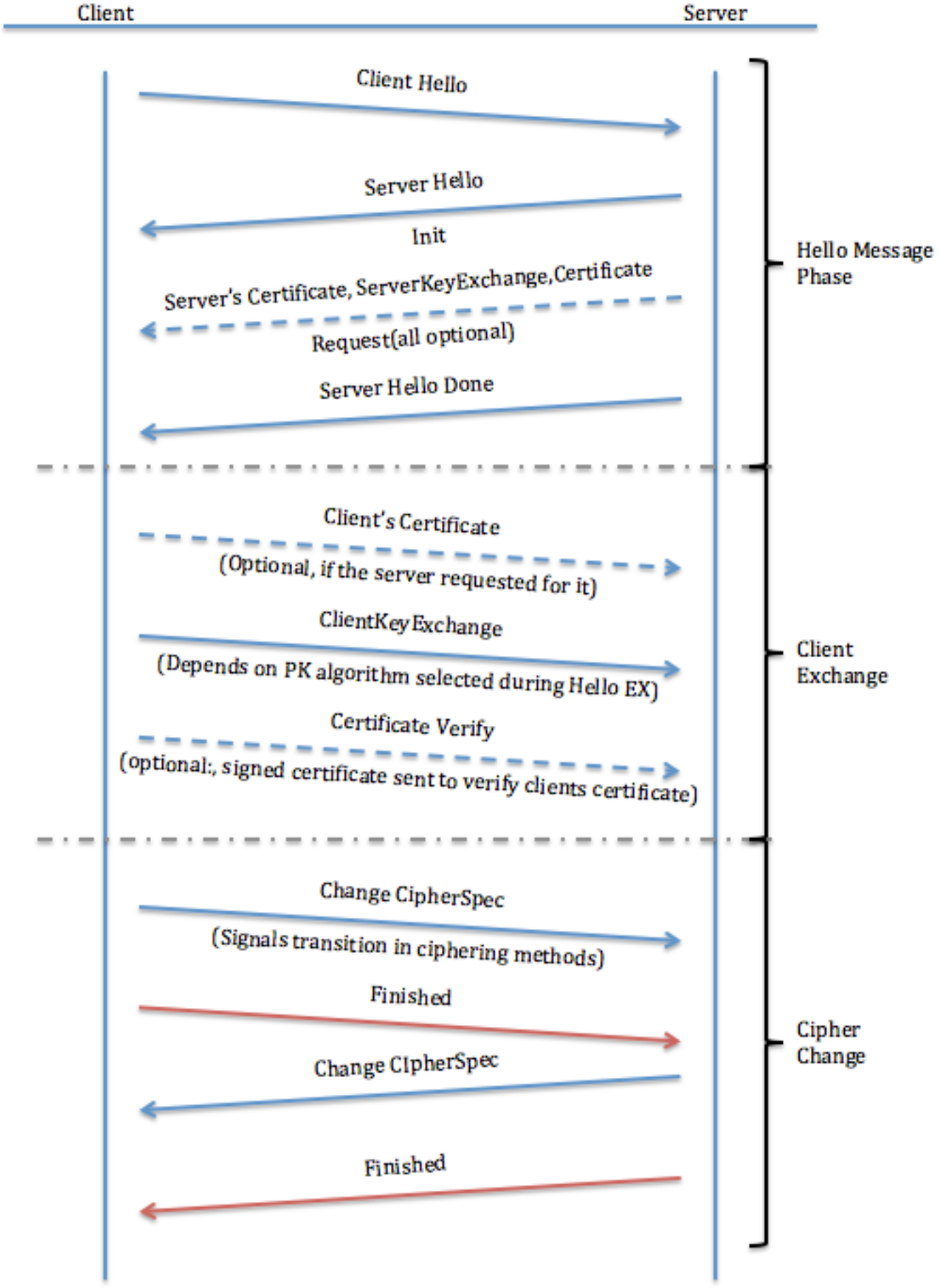
- **关闭警报**：必须适当地关闭客户端和服务器之间的连接为了避免任何截断攻击。表明到收件人的close_notify信息传送发送方不会再将传送在该连接的信息。
- **错误警报**：当错误检测时，检测的当事人传送信息对另一个当事人。 在一个致命警报消息的发射或收据，两个当事人立即断开连接。错误警报某些示例是：
  - unexpected_message (致命)
  - decompression_failure
  - handshake_failure

应用程序数据

这些记录包含实际应用程序数据。这些消息由记录层传播和根据当前连接状态被分段，被压缩，并且加密。

示例处理

此部分描述在客户端和服务器之间的示例处理。


Client
Server
Client Hello
Server Hello
Init
Server's Certificate, ServerKeyExchange,Certificate
Request(all optional)
Server Hello Done
Hello Message Phase
Client's Certificate
(Optional, if the server requested for it)
ClientKeyExchange
(Depends on PK algorithm selected during Hello EX)
Certificate Verify
(optional:, signed certificate sent to verify clients certificate)
Client Exchange
Change CipherSpec
(Signals transition in ciphering methods)
Finished
Change CIpherSpec
Finished
Cipher Change

Hello交换

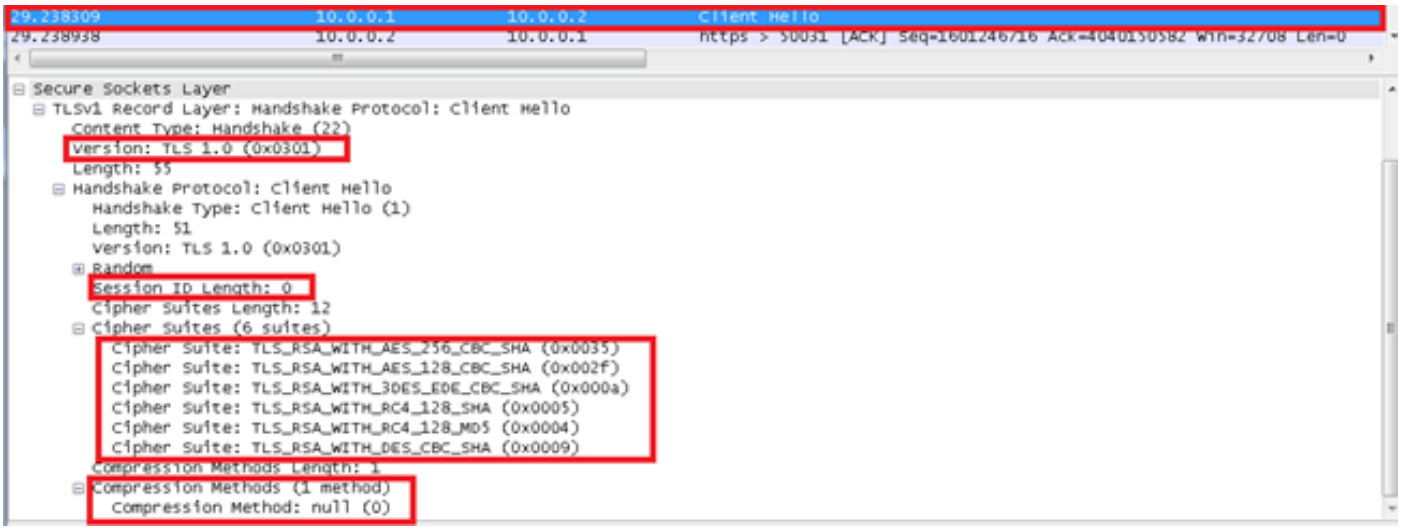
当SSL客户端和服务器开始通信时，他们对协议版本达成协议，挑选加密算法，或者互相验证，并且使用公用密钥加密技术为了生成共享秘密。这些进程在握手协议进行。总之，客户端传送客户端Hello消息到服务器，必须回应服务器问候消息或致命错误生成，并且连接发生故障。客户端Hello和服务器问候用于设立在客户端和服务器之间的安全性增强功能。

客户端Hello

客户端Hello发送这些属性到服务器：

- **协议版本**：客户端希望通信SSL协议的版本在此会话期间。
- **会话 ID**:客户端希望使用此连接会话的ID。在交换的第一个客户端Hello，会话ID是空的(参考数据包捕获屏幕画面在注意以后)。
- **密码器套件**：这从服务器的客户端通过在客户端Hello消息。它包含客户端支持的加密算法的组合按照客户端首选(首先第一选择的顺序)。每个密码器套件定义了密钥交换算法和密码器spec。服务器选择密码器套件或，如果没有提交可接受选择，返回握手失败警报并且断开连接。
- **压缩方法**：包括客户端支持的压缩算法列表。如果服务器不支持客户端发送的任何方法，连接发生故障。压缩方法可以也空。

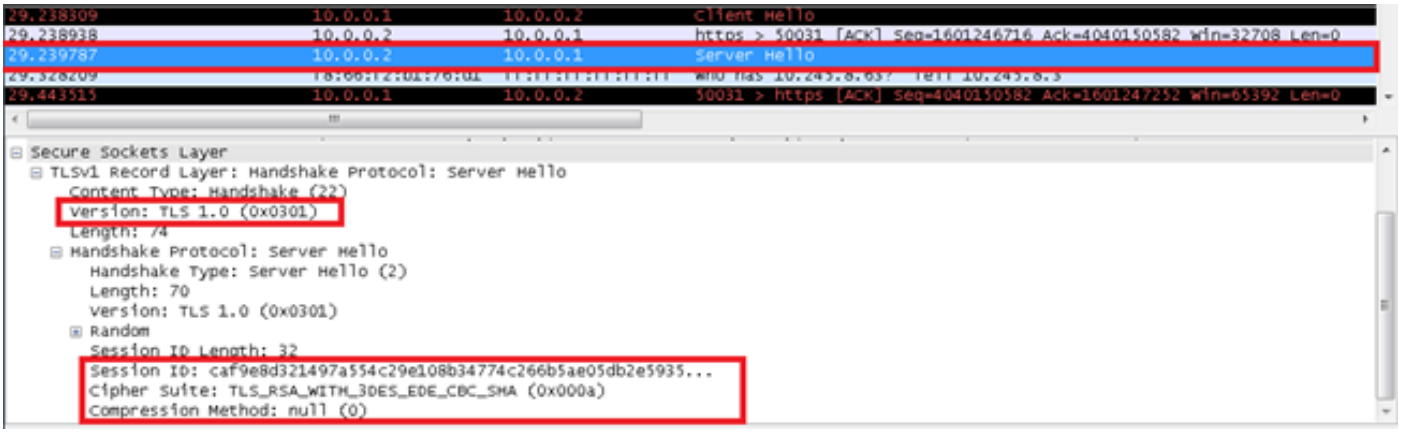
Note:在捕获的服务器IP地址是10.0.0.2，并且客户端IP地址是10.0.0.1。



服务器问候

服务器退还这些属性给客户端：

- **协议版本**：该SSL的协议的选定的版本客户端支持。
- **会话 ID**:这是对应于此连接会话的标识。如果客户端的Hello客户端发送的会话ID不是空的，服务器在会话缓存查找为匹配。如果找到匹配使用指定的会话状态，并且服务器是愿意建立新连接，服务器回应客户端提供的同一个值。这指示一恢复的会话并且指明当事人必须继续直接地到已完成消息。否则，此字段包含识别个新会话的一个不同的值。服务器也许返回一空session_id为了表明不会缓存会话，并且不可能恢复。
- **密码器套件**：如选择由从从客户端发送的列表的服务器。
- **压缩方法**：如选择由从从客户端发送的列表的服务器。
- **证书请求**：服务器发送客户端配置对此所有证书的列表，并且允许证书它要使用验证的客户端选择。

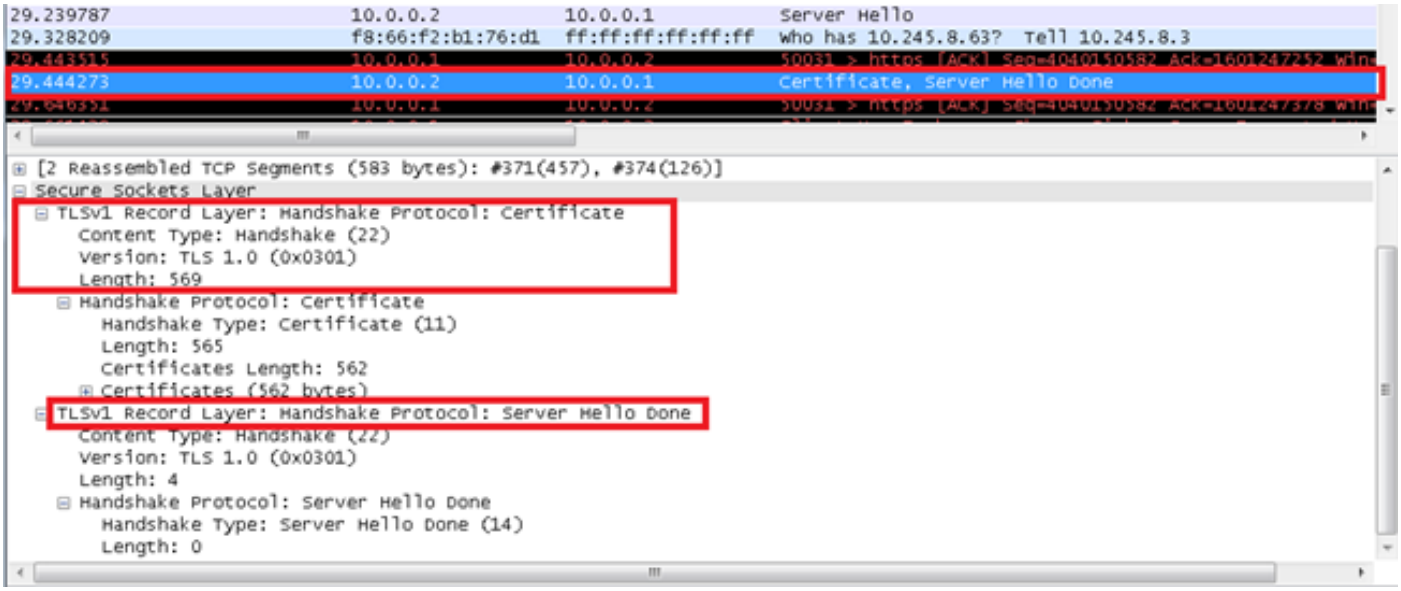


SSL会话恢复请求：

- 服务器能发送Hello请求对客户端。这是为了只提醒客户端应该开始与客户端Hello请求的重新协商，当方便。如果握手进程已经进行中，客户端忽略从服务器的Hello请求。
- 握手消息有在应用程序数据发射的更多优先。重新协商必须在不大于一两倍内开始最大长度应用程序数据数据信息的传输时间。

执行的服务器问候

服务器问候完成的信息由服务器传送为了指示服务器问候和相关的消息的末端。在它传送此信息后，服务器等待客户端答复。收到执行的服务器问候后消息，客户端验证服务器提供了一个有效证书，如果必须，并且检查服务器问候参数是可接受。



服务器证书、服务器密钥Exchange和证书请求(可选)

- **服务器证书：** 如果必须验证服务器(通常是实际情形)，服务器发送其在服务器问候消息之后的证书。证书类型一定是适当的为选定密码器套件密钥交换算法，并且通常是X.509.v3证书。
- **服务器密钥Exchange ：** 如果没有证书，服务器密钥交换消息由服务器传送。如果Diffie-Hellman (DH)参数包括与服务器证书，没有使用此消息。
- **证书请求：** 服务器能或者请求从客户端的一证书，如果适当为选定密码器套件。

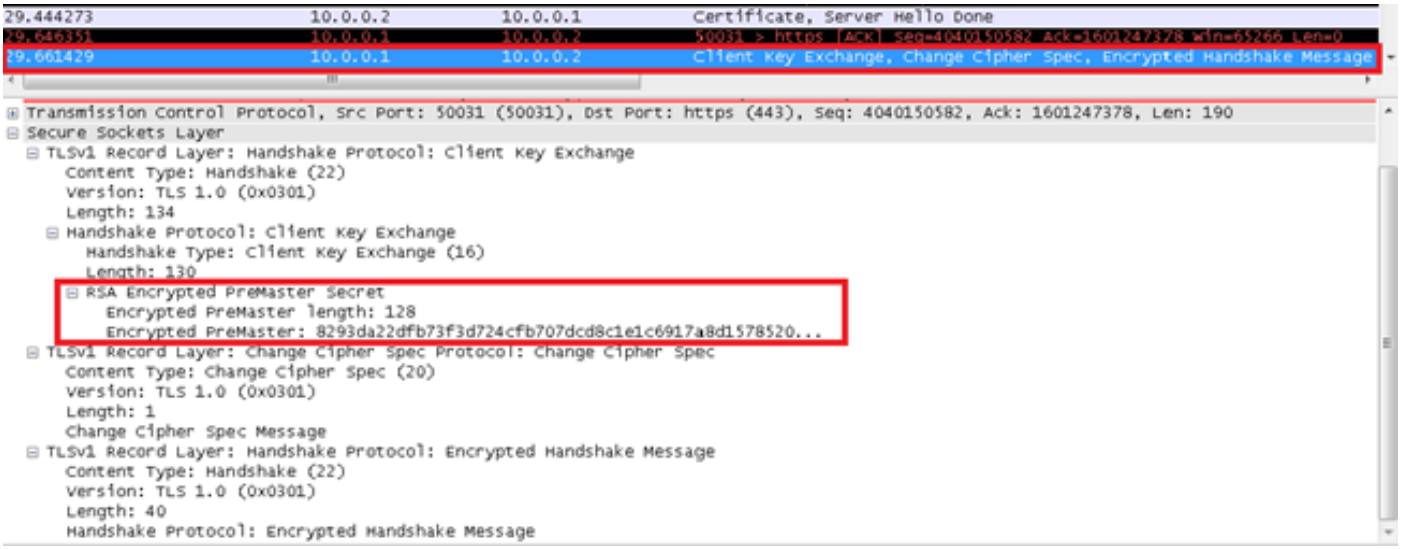
客户端Exchange

客户端证书(可选)

这是客户端发送的第一条消息，在他或她收到服务器问候完成的消息后。如果服务器请求证书，此信息只传送。如果适当的证书不是可用的，客户端发送**no_certificate**警报。此警报是仅警告;然而，如果客户端验证要求，服务器也许回应一致命握手失败警报。客户端DH证书必须匹配服务器指定的DH参数。

客户端密钥交换

此消息内容取决于公共密钥算法选择在客户端Hello和服务器问候消息之间。客户端使用Rivest沙米尔Addleman (RSA)算法加密的一premaster密钥或DH关键协议和验证。当RSA使用服务器验证和密钥交换时， 48字节**pre_master_secret**由客户端生成，加密在服务器公共密钥下，并且发送到服务器。服务器使用专用密钥为了解密**pre_master_secret**。两个当事人然后转换**pre_master_secret**到**master_secret**。



Certificate verify (可选)

如果客户端发送与签署的能力的一证书，数字式地签字的Certificate verify信息传送为了明确地验证证书。

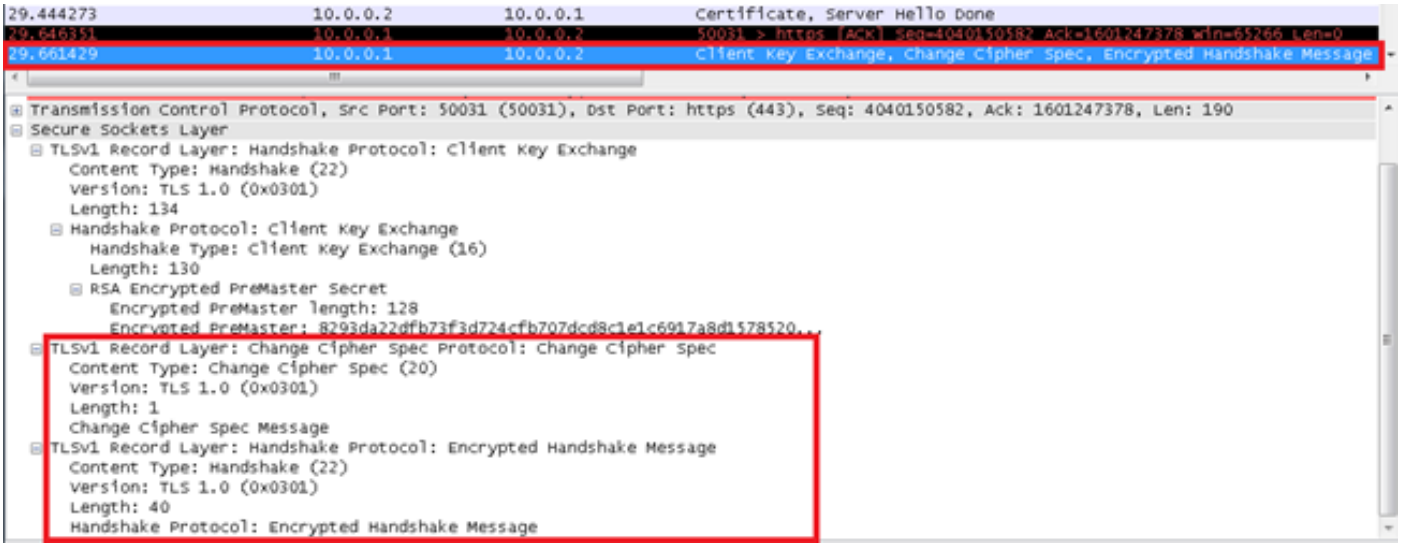
密码器崔凡吉莱

崔凡吉莱密码器Spec消息

崔凡吉莱密码器Spec信息由客户端传送，并且客户端复制待定密码器Spec (新的)到当前密码器Spec (以前使用)的那个。崔凡吉莱密码器Spec协议存在为了在加密的策略的信号转换。 协议包括单个消息，加密并且被压缩在当前(不是待定)密码器Spec.下。 信息由两个传送客户端和服务器为了通知接收方随后的记录保护在最近经过协商的密码器Spec和密钥下。此消息的接收造成接收方复制被读取等待状态到读的当前状态。客户端在握手密钥交换和Certificate verify消息以后发送崔凡吉莱密码器Spec消息(若有)和服务器发送一，在从客户端接收的顺利地处理密钥交换交换消息后。当上次会话恢复时，崔凡吉莱密码器Spec信息在Hello消息以后传送。在捕获，客户端Exchange，崔凡吉莱密码器和已完成信息传送作为从客户端的单个消息。

已完成消息

已完成信息总是传送，在崔凡吉莱密码器Spec消息为了验证之后密钥交换和认证过程是成功的。已完成消息是有最近经过协商的算法、密钥和秘密的第一保护的数据包。 已完成消息的确认没有要求;在他们传送已完成信息之后，当事人能开始发送已加密数据。已完成消息的收件人必须验证内容正确。



相关信息

- [RFC 6101 -安全套接字协议层协议版本3.0]
- [Wireshark SSL wiki]-请解码有Wireshark的SSL数据包
- [技术支持和文档 - Cisco Systems]