**HTTPS认证流程简介**

# 概述

## 编写目的

主要介绍HTTP SSL 认证流程。

## 阅读对象

HTTPS 应用功能的开发人员与相关的测试人员。

# HTTPS简介

## 技术背景

实际的网络中TCP/IP共分为四层：应用层、传输层、网络层、数据链路层；这样做的目的是：分层能够解耦，动态替换层内协议。每一层包含的内容大致如下：

* 应用层：向用户提供应用服务时的通讯活动（ftp，dns，http）
* 传输层：网络连接中两台计算机的数据传输（tcp、udp）
* 网络层：处理网络上流动的数据包，通过怎样的传输路径把数据包传送给对方（ip）
* 数据链路层：与硬件相关的网卡、设备驱动等等

然而HTTP也有以下明显缺点：

* 通信使用明文，内容可能被窃听
* 不验证通信方的身份，因此有可能遭遇伪装
* 无法证明报文的完整性，所以有可能遭到篡改

为了弥补HTTP协议的不足，HTTPS就登场了。HTTPS中的S表示SSL或者TLS，就是在原HTTP的基础上加上一层用于数据加密、解密、身份认证的**安全层**，可以简单的描述为：

**HTTP + 加密 + 认证 + 完整性保护 = HTTPS**

## SSL简介

SSL(Secure Sockets Layer 安全套接字协议),及其继任者传输层安全（Transport Layer Security，TLS）是为网络通信提供安全及数据完整性的一种安全协议。TLS与SSL在传输层与应用层之间对网络连接进行加密。加入SSL后的TCP/IP模型如下图2.1所示：



图2.1 加入SSL后的TCP/IP模型

SSL的体系结构中包含两个协议子层，其中底层是SSL记录协议层（SSL Record Protocol Layer）；高层是SSL握手协议层（SSL HandShake Protocol Layer）。

* SSL记录协议针对HTTP协议进行了特别的设计，使得HTTP能够在SSL运行，具体实施压缩解压缩、加密解密、计算和校验MAC等与安全有关的操作。
* SSL握手协议用于SSL管理信息的交换，允许应用协议传送数据之间相互验证，协商加密算法和生成密钥等。

由上简介不难理解SSL协议可以具备以下三个特性：

1. 保密：在握手协议中定义了会话密钥后，所有的消息都被加密。
2. 鉴别：可选的客户端认证，和强制的服务器端认证。
3. 完整性：传送的消息包括消息完整性检查（使用MAC）。

# SSL认证流程

## SSL单向认证流程

SSL单向认证是指在使用HTTPS建立连接的时候，由客户端验证服务器身份的交互过程。单向认证多用于个与企业应该之间。具体流程如图3.1所示。过程描述如下：

* 1. 客户端向服务端发送SSL协议版本号、加密算法种类、随机数等信息。
  2. 服务端给客户端返回SSL协议版本号、加密算法种类、随机数等信息，同时也返回服务器端的证书，即公钥证书
  3. 客户端使用服务端返回的信息验证服务器的合法性，包括：
* 证书是否过期
* 服务器证书的CA是否可靠
* 服务器证书公钥是否能正确解开服务器证书中的数字签名
* 服务器证书上的域名是否和服务器的实际域名相匹配
* 验证通过后，将继续进行通信，否则，终止通信
  1. 客户端向服务端发送自己所能支持的对称加密方案，供服务器端进行选择
  2. 服务器端在客户端提供的加密方案中选择加密程度最高的加密方式。
  3. 服务器将选择好的加密方案通过明文方式返回给客户端
  4. 客户端接收到服务端返回的加密方式后，使用该加密方式生成产生随机码，用作通信过程中对称加密的密钥，使用服务端返回的公钥进行加密，将加密后的随机码发送至服务器
  5. 服务器收到客户端返回的加密信息后，使用自己的私钥进行解密，获取对称加密密钥。 在接下来的会话中，服务器和客户端将会使用该密码进行对称加密，保证通信过程中信息的安全。



图3.1 SSL单向认证流程

## SSL双向认证流程

SSL双向认证是指在使用HTTPS建立连接的时候，由客户端验证服务器身份而服务器也要求客户端提供客户端自己的证书并验证的交互过程。双向认证多用于企业之间。具体流程如图3.2所示。过程描述如下：

* 1. 客户端向服务端发送SSL协议版本号、加密算法种类、随机数等信息。
  2. 服务端给客户端返回SSL协议版本号、加密算法种类、随机数等信息，同时也返回服务器端的证书，即公钥证书
  3. 客户端使用服务端返回的信息验证服务器的合法性，包括：
* 证书是否过期
* 发行服务器证书的CA是否可靠
* 返回的公钥是否能正确解开返回证书中的数字签名
* 服务器证书上的域名是否和服务器的实际域名相匹配
* 验证通过后，将继续进行通信，否则，终止通信
  1. 服务端要求客户端发送客户端的证书，客户端会将自己的证书发送至服务端
  2. 验证客户端的证书，通过验证后，会获得客户端的公钥
  3. 客户端向服务端发送自己所能支持的对称加密方案，供服务器端进行选择
  4. 服务器端在客户端提供的加密方案中选择加密程度最高的加密方式
  5. 将加密方案通过使用之前获取到的公钥进行加密，返回给客户端
  6. 客户端收到服务端返回的加密方案密文后，使用自己的私钥进行解密，获取具体加密方式，而后，产生该加密方式的随机码，用作加密过程中的密钥，使用之前从服务端证书中获取到的公钥进行加密后，发送给服务端
  7. 服务端收到客户端发送的消息后，使用自己的私钥进行解密，获取对称加密的密钥，在接下来的会话中，服务器和客户端将会使用该密码进行对称加密，保证通信过程中信息的安全。



图3.2 SSL双向认证交互流程

## SSL两种认证方式的对比

双向认证 SSL 协议要求服务器和用户双方都有证书。不同点主要有两个方面：

* 单向认证 SSL 协议不需要客户拥有CA证书，相对于双向认证只需将服务器端验证客户证书的过程去掉；
* 在协商对称密码方案和对称会话密钥时，单向认证服务器发送给客户的是没有加过密的密码方案，而双向认证是经过加密的密码方案；

对于单向认证，双方具体的通讯内容，是加过密的数据，如果有第三方攻击，获得的只是加密的数据，第三方要获得有用的信息，需要对加密的数据进行解密，这时候的安全依赖于密码方案的安全。而幸运的是，目前所用的密码方案，只要通讯密钥长度足够的长，信息就足够的安全。这也是我们强调要求使用128位加密通讯的原因。

一般Web应用都是采用SSL单向认证的，原因很简单，用户数目广泛，且无需在通讯层对用户身份进行验证，一般都在应用逻辑层来保证用户的合法登入。但如果是企业应用对接，情况不一样，可能会要求对客户端做身份验证。这时需要做SSL双向认证。