https

本词条由[“科普中国”百科科学词条编写与应用工作项目](http://www.cast.org.cn/) 审核 。

HTTPS（全称：Hyper Text Transfer Protocol over Secure Socket Layer），是以安全为目标的[HTTP](https://baike.baidu.com/item/HTTP)通道，简单讲是HTTP的安全版。即HTTP下加入SSL层，HTTPS的安全基础是SSL，因此加密的详细内容就需要SSL。 它是一个URI scheme（抽象标识符体系），句法类同[http](https://baike.baidu.com/item/http):体系。用于安全的HTTP数据传输。https:URL表明它使用了HTTP，但HTTPS存在不同于HTTP的默认[端口](https://baike.baidu.com/item/%E7%AB%AF%E5%8F%A3)及一个加密/身份验证层（在HTTP与TCP之间）。这个系统的最初研发由[网景](https://baike.baidu.com/item/%E7%BD%91%E6%99%AF)公司(Netscape)进行，并内置于其浏览器Netscape Navigator中，提供了身份验证与加密[通讯](https://baike.baidu.com/item/%E9%80%9A%E8%AE%AF)方法。现在它被广泛用于[万维网](https://baike.baidu.com/item/%E4%B8%87%E7%BB%B4%E7%BD%91)上安全敏感的通讯，例如交易支付方面。[1]

**中文名**

网络协议

**外文名**

HTTPS

**目    标**

安全

**应    用**

安全数据传输

目录

1 [历史](https://baike.baidu.com/item/https/285356?fr=aladdin#1)

2 [区别](https://baike.baidu.com/item/https/285356?fr=aladdin#2)

3 [解决问题](https://baike.baidu.com/item/https/285356?fr=aladdin#3)

4 [限制](https://baike.baidu.com/item/https/285356?fr=aladdin#4)

5 [ssl](https://baike.baidu.com/item/https/285356?fr=aladdin#5)

6 [握手过程](https://baike.baidu.com/item/https/285356?fr=aladdin#6)

7 [名词释义](https://baike.baidu.com/item/https/285356?fr=aladdin#7)

历史

[网景](https://baike.baidu.com/item/%E7%BD%91%E6%99%AF)在1994年创建了HTTPS，并应用在[网景导航者](https://baike.baidu.com/item/%E7%BD%91%E6%99%AF%E5%AF%BC%E8%88%AA%E8%80%85)浏览器中。 最初，HTTPS是与[SSL](https://baike.baidu.com/item/SSL)一起使用的；在SSL逐渐演变到[TLS](https://baike.baidu.com/item/TLS)时，最新的HTTPS也由在2000年五月公布的RFC 2818正式确定下来。

它是由[Netscape](https://baike.baidu.com/item/Netscape)开发并内置于其[浏览器](https://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8)中，用于对数据进行加密和解密操作，并返回网络上传送回的结果。HTTPS实际上应用了[Netscape](https://baike.baidu.com/item/Netscape)的安全套接层（[SSL](https://baike.baidu.com/item/SSL)）作为[HTTP](https://baike.baidu.com/item/HTTP)应用层的子层。（HTTPS使用[端口](https://baike.baidu.com/item/%E7%AB%AF%E5%8F%A3)443，而不是像HTTP那样使用端口80来和TCP/IP进行通信。）SSL使用40 位关键字作为RC4流加密算法，这对于商业信息的加密是合适的。HTTPS和SSL支持使用X.509数字认证，如果需要的话用户可以确认发送者是谁。[1]

也就是说它的主要作用可以分为两种：一种是建立一个[信息安全](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF%E5%AE%89%E5%85%A8)通道，来保证数据传输的安全；另一种就是确认网站的真实性，凡是使用了 https 的网站，都可以通过点击浏览器地址栏的锁头标志来查看网站认证之后的真实信息，也可以通过 CA 机构颁发的安全签章来查询[2]  。

区别

**HTTPS和HTTP的区别**

超文本传输协议HTTP协议被用于在Web浏览器和网站服务器之间传递信息。HTTP协议以明文方式发送内容，不提供任何方式的数据加密，如果攻击者截取了Web浏览器和网站服务器之间的传输报文，就可以直接读懂其中的信息，因此HTTP协议不适合传输一些敏感信息，比如信用卡号、密码等。

为了解决HTTP协议的这一缺陷，需要使用另一种协议：安全套接字层超文本传输协议HTTPS。为了数据传输的安全，HTTPS在HTTP的基础上加入了SSL协议，SSL依靠证书来验证服务器的身份，并为浏览器和服务器之间的通信加密。

HTTPS和HTTP的区别主要为以下四点：

一、https协议需要到ca申请证书，一般免费证书很少，需要交费。

二、http是[超文本传输协议](https://baike.baidu.com/item/%E8%B6%85%E6%96%87%E6%9C%AC%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE)，信息是明文传输，https 则是具有[安全性](https://baike.baidu.com/item/%E5%AE%89%E5%85%A8%E6%80%A7)的[ssl](https://baike.baidu.com/item/ssl)加密传输协议。

三、http和https使用的是完全不同的连接方式，用的端口也不一样，前者是80，后者是443。

四、http的连接很简单，是无状态的；HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的[网络协议](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%8D%8F%E8%AE%AE)，比http协议安全。

解决问题

**信任主机的问题**

采用https的[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)必须从CA （Certificate Authority）申请一个用于证明服务器用途类型的证书。该证书只有用于对应的服务器的时候，客户端才信任此主机。所以所有的银行系统网站，关键部分应用都是https 的。客户通过信任该证书，从而信任了该主机。其实这样做效率很低，但是银行更侧重安全。这一点对局域网对内提供服务处的服务器没有任何意义。局域网中的服务器，采用的证书不管是自己发布的还是从公众的地方发布的，其客户端都是自己人，所以该局域网中的客户端也就肯定信任该服务器。

**通讯过程中的数据的泄密和被篡改**

1． 一般意义上的https，就是服务器有一个证书。

a) 主要目的是保证服务器就是他声称的服务器，这个跟第一点一样。

b)[服务端](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E7%AB%AF)和[客户端](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E7%AB%AF)之间的所有[通讯](https://baike.baidu.com/item/%E9%80%9A%E8%AE%AF)，都是加密的。

i. 具体讲，是客户端产生一个对称的[密钥](https://baike.baidu.com/item/%E5%AF%86%E9%92%A5)，通过服务器的证书来交换密钥，即一般意义上的握手过程。

ii. 接下来所有的信息往来就都是加密的。第三方即使截获，也没有任何意义，因为他没有密钥，当然篡改也就没有什么意义了。

2． 少许对客户端有要求的情况下，会要求客户端也必须有一个证书。

a) 这里客户端证书，其实就类似表示个人信息的时候，除了用户名/密码，还有一个CA 认证过的身份。因为个人证书一般来说是别人无法模拟的，所以这样能够更深的确认自己的身份。

b) 目前大多数个人银行的专业版是这种做法，具体证书可能是拿U盘（即U盾）作为一个备份的载体。

限制

它的安全保护依赖浏览器的正确实现以及[服务器软件](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8%E8%BD%AF%E4%BB%B6)、实际加密算法的支持。

一种常见的误解是“银行用户在线使用https:就能充分彻底保障他们的银行卡号不被偷窃。”实际上，与服务器的加密连接中能保护银行卡号的部分，只有用户到服务器之间的连接及服务器自身。并不能绝对确保服务器自己是安全的，这点甚至已被攻击者利用，常见例子是模仿银行[域名](https://baike.baidu.com/item/%E5%9F%9F%E5%90%8D)的钓鱼攻击。少数罕见攻击在网站传输客户数据时发生，攻击者会尝试窃听传输中的数据。

商业网站被人们期望迅速尽早引入新的特殊处理程序到金融[网关](https://baike.baidu.com/item/%E7%BD%91%E5%85%B3)，仅保留传输码(transaction number)。不过他们常常存储银行卡号在同一个[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93)里。那些[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93)和服务器少数情况有可能被未授权用户攻击和损害。[1]

TLS 1.1之前，这段仅针对TLS 1.1之前的[状况](https://baike.baidu.com/item/%E7%8A%B6%E5%86%B5)。因为SSL位于http的下一层，并不能理解更高层协议，通常SSL服务器仅能颁证给特定的IP/[端口](https://baike.baidu.com/item/%E7%AB%AF%E5%8F%A3)组合。这指它经常不能在[虚拟主机](https://baike.baidu.com/item/%E8%99%9A%E6%8B%9F%E4%B8%BB%E6%9C%BA)(基于域名)上与HTTP正常组合成HTTPS。

这一点已被即将来临的TLS 1.1更新为—种完全支持基于域名的[虚拟主机](https://baike.baidu.com/item/%E8%99%9A%E6%8B%9F%E4%B8%BB%E6%9C%BA)。

ssl

SSL(Secure Sockets Layer 安全套接层),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在传输层对网络连接进行加密。

SSL (Secure Socket Layer)为Netscape所研发，用以保障在Internet上数据传输之安全，利用[数据加密](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%8A%A0%E5%AF%86)(Encryption)技术，可确保数据在网络上之传输过程中不会被截取及窃听。目前一般通用之规格为40 bit之安全标准，[美国](https://baike.baidu.com/item/%E7%BE%8E%E5%9B%BD)则已推出128 bit之更高安全标准，但限制出境。只要3.0版本以上之I.E.或[Netscape浏览器](https://baike.baidu.com/item/Netscape%E6%B5%8F%E8%A7%88%E5%99%A8)即可支持SSL。

当前版本为3.0。它已被广泛地用于[Web浏览器](https://baike.baidu.com/item/Web%E6%B5%8F%E8%A7%88%E5%99%A8)与服务器之间的身份认证和加密数据传输。

SSL协议位于[TCP/IP协议](https://baike.baidu.com/item/TCP%2FIP%E5%8D%8F%E8%AE%AE)与各种应用层协议之间，为[数据通讯](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%80%9A%E8%AE%AF)提供安全支持。SSL协议可分为两层：SSL记录协议（SSL Record Protocol）：它建立在可靠的传输协议（如TCP）之上，为高层协议提供[数据封装](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%B0%81%E8%A3%85)、压缩、加密等基本功能的支持。SSL握手协议（SSL Handshake Protocol）：它建立在SSL记录协议之上，用于在实际的数据传输开始前，通讯双方进行身份认证、协商加密算法、交换加密密钥等。

**SSL协议提供的服务主要有哪些**

1）认证用户和服务器，确保数据发送到正确的[客户机](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E6%9C%BA)和服务器

2）加密数据以防止数据中途被窃取

3）维护数据的完整性，确保数据在传输过程中不被改变。

**SSL协议的工作流程**

服务器认证阶段：

1）客户端向服务器发送一个开始信息“Hello”以便开始一个新的会话连接；

2）服务器根据客户的信息确定是否需要生成新的主密钥，如需要则服务器在响应客户的“Hello”信息时将包含生成主密钥所需的信息；

3）客户根据收到的服务器响应信息，产生一个主密钥，并用服务器的[公开密钥加密](https://baike.baidu.com/item/%E5%85%AC%E5%BC%80%E5%AF%86%E9%92%A5%E5%8A%A0%E5%AF%86)后传给服务器；

4）服务器恢复该主密钥，并返回给客户一个用主密钥认证的信息，以此让[客户认证](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E8%AE%A4%E8%AF%81)服务器。

**用户认证阶段**

在此之前，服务器已经通过了[客户认证](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E8%AE%A4%E8%AF%81)，这一阶段主要完成对客户的认证。经认证的服务器发送一个提问给客户，客户则返回（[数字](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97)）签名后的提问和其公开密钥，从而向服务器提供认证。

从SSL 协议所提供的服务及其工作流程可以看出，SSL协议运行的基础是商家对[消费者](https://baike.baidu.com/item/%E6%B6%88%E8%B4%B9%E8%80%85)信息保密的承诺，这就有利于商家而不利于消费者。在[电子商务](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1)初级阶段，由于运作电子商务的企业大多是信誉较高的大公司，因此这问题还没有充分暴露出来。但随着电子商务的发展，各中小型公司也参与进来，这样在[电子支付](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E6%94%AF%E4%BB%98)过程中的单一认证问题就越来越突出。虽然在SSL3.0中通过[数字签名](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E7%AD%BE%E5%90%8D)和[数字证书](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E8%AF%81%E4%B9%A6)可实现浏览器和Web服务器双方的身份验证，但是SSL协议仍存在一些问题，比如，只能提供交易中客户与服务器间的双方认证，在涉及多方的电子交易中，SSL协议并不能协调各方间的安全传输和信任关系。在这种情况下，Visa和MasterCard两大[信用卡](https://baike.baidu.com/item/%E4%BF%A1%E7%94%A8%E5%8D%A1)公司组织制定了SET协议，为网上信用卡支付提供了全球性的标准。[3]

握手过程

为了便于更好的认识和理解SSL 协议，这里着重介绍SSL 协议的握手协议。SSL 协议既用到了公钥加密技术又用到了对称加密技术，对称加密技术虽然比公钥加密技术的速度快，可是公钥加密技术提供了更好的[身份认证技术](https://baike.baidu.com/item/%E8%BA%AB%E4%BB%BD%E8%AE%A4%E8%AF%81%E6%8A%80%E6%9C%AF)。SSL 的握手协议非常有效的让客户和服务器之间完成相互之间的身份认证，其主要过程如下：

①客户端的[浏览器](https://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8)向服务器传送客户端SSL 协议的[版本号](https://baike.baidu.com/item/%E7%89%88%E6%9C%AC%E5%8F%B7)，加密算法的种类，产生的随机数，以及其他服务器和客户端之间通讯所需要的各种信息。

②服务器向客户端传送SSL 协议的版本号，加密算法的种类，随机数以及其他相关信息，同时服务器还将向客户端传送自己的证书。

③客户利用服务器传过来的信息验证服务器的合法性，服务器的合法性包括：证书是否过期，发行[服务器证书](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8%E8%AF%81%E4%B9%A6)的CA 是否可靠，发行者证书的公钥能否正确解开服务器证书的“发行者的数字签名”，服务器证书上的[域名](https://baike.baidu.com/item/%E5%9F%9F%E5%90%8D)是否和服务器的实际域名相匹配。如果合法性验证没有通过，通讯将断开；如果合法性验证通过，将继续进行第四步。

④用户端随机产生一个用于后面通讯的“对称密码”，然后用服务器的公钥（服务器的公钥从步骤②中的服务器的证书中获得）对其加密，然后将加密后的“预主密码”传给服务器。

⑤如果服务器要求客户的身份认证（在握手过程中为可选），用户可以建立一个随机数然后对其进行数据签名，将这个含有签名的随机数和客户自己的证书以及加密过的“预主密码”一起传给服务器。

⑥如果服务器要求客户的身份认证，服务器必须检验客户证书和签名随机数的合法性，具体的合法性验证过程包括：客户的证书使用日期是否有效，为客户提供证书的CA 是否可靠，发行CA 的公钥能否正确解开客户证书的发行CA 的数字签名，检查客户的证书是否在证书废止列表（CRL）中。检验如果没有通过，通讯立刻中断；如果验证通过，服务器将用自己的私钥解开加密的“预主密码”，然后执行一系列步骤来产生主[通讯密码](https://baike.baidu.com/item/%E9%80%9A%E8%AE%AF%E5%AF%86%E7%A0%81)（客户端也将通过同样的方法产生相同的主通讯密码）。

⑦服务器和客户端用相同的主密码即“通话密码”，一个[对称密钥](https://baike.baidu.com/item/%E5%AF%B9%E7%A7%B0%E5%AF%86%E9%92%A5)用于SSL 协议的安全[数据通讯](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%80%9A%E8%AE%AF)的加解密[通讯](https://baike.baidu.com/item/%E9%80%9A%E8%AE%AF)。同时在SSL 通讯过程中还要完成数据通讯的完整性，防止数据通讯中的任何变化。

⑧[客户端](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E7%AB%AF)向[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)端发出信息，指明后面的数据通讯将使用的步骤⑦中的主密码为[对称密钥](https://baike.baidu.com/item/%E5%AF%B9%E7%A7%B0%E5%AF%86%E9%92%A5)，同时通知服务器客户端的握手过程结束。

⑨服务器向客户端发出信息，指明后面的数据通讯将使用的步骤⑦中的主密码为对称密钥，同时通知客户端服务器端的握手过程结束。

⑩SSL 的握手部分结束，SSL 安全通道的数据通讯开始，客户和服务器开始使用相同的对称密钥进行数据通讯，同时进行通讯完整性的检验。

名词释义

如果要启用SSL通道，那么需要使用SSL证书来启用[http](https://baike.baidu.com/item/http)s协议，SSl证书包含信息：

证书版本号，不同版本的证书[格式](https://baike.baidu.com/item/%E6%A0%BC%E5%BC%8F)不同

Serial Number　序列号，同一身份验证机构签发的证书序列号唯一

Algorithm Identifier 　签名[算法](https://baike.baidu.com/item/%E7%AE%97%E6%B3%95)，包括必要的参数Issuer 身份验证机构的标识信息

Period of Validity 　有效期

Subject　证书持有人的标识信息

Subject’s Public Key　证书持有人的公钥

Signature　身份验证机构对证书的签名

证书的格式　 认证中心所发放的证书均遵循X.509 V3 标准，其基本格式如下：

证书版本号（Certificate Format Version）

含义：用来指定证书格式采用的X.509 版本号。

证书序列号（Certificate Serial Number）

含义：用来指定证书的唯一序列号，以标识CA 发出的所有公钥证书。

签名（Signature）算法标识（Algorithm Identifier）

含义：用来指定 CA 签发证书所用的签名算法。

签发此证书的 CA 名称（Issuer ）

含义：用来指定签发证书的 CA 的X.500 唯一名称（DN，Distinguished Name）。

证书有效期（Validity Period）起始日期（notBefore） 终止日期（notAfter）

含义：用来指定证书起始日期和终止日期。

用户名称（Subject）

含义：用来指定证书用户的X.500 唯一名称（DN，Distinguished Name）。

用户公钥信息（Subject Public Key Information）算法（algorithm） 算法标识（Algorithm　Identifier）用户公钥（subject　Public　Key）

含义：用来标识公钥使用的算法，并包含公钥本身。

证书扩充部分（扩展域）（Extensions）

含义：用来指定额外信息。

X.509 V3 证书的扩充部分（扩展域）及实现方法如下：

CA 的公钥标识（Authority　Key　Identifier）

公钥标识（SET 未使用）（Key　Identifier）

签发证书者证书的签发者的甄别名（Certificate　Issuer）

签发证书者证书的序列号（Certificate Serial Number）

X.509 V3 证书的扩充部分（扩展域）及实现CA 的公钥标识（Authority　Key　Identifier）

公钥标识（SET 未使用）（Key　Identifier）

签发证书者证书的签发者的甄别名（Certificat签发证书者证书的序列号（Certificate Serial Number）

含义：CA 签名证书所用的[密钥](https://baike.baidu.com/item/%E5%AF%86%E9%92%A5)对的唯一标识

用户的公钥标识（Subject　Key　Identifier）

含义：用来标识与证书中公钥相关的特定密钥进行解密。

证书中的公钥用途（Key　Usage）

含义：用来指定公钥用途。

用户的私钥有效期（Private　Key　Usage　Period）起始日期（Note　Before） 终止日期（Note　After）

含义：用来指定用户签名私钥的起始日期和终止日期。

CA 承认的证书政策列表（Certificate Policies）

含义：用来指定用户证书所适用的政策，证书政策可由对象[标识符](https://baike.baidu.com/item/%E6%A0%87%E8%AF%86%E7%AC%A6)表示。

用户的代用名（Substitutional　Name）

含义：用来指定用户的代用名。

CA 的代用名（Issuer　Alt　Name）

含义：用来指定 CA 的代用名。

基本制约（Basic　Constraints）

含义：用来表明证书用户是最终用户还是CA。 在SET 系统中有一些私有扩充部分（扩展域）Hashed　Root　Key 含义：只在[根证书](https://baike.baidu.com/item/%E6%A0%B9%E8%AF%81%E4%B9%A6)中使用，用于[证书更新](https://baike.baidu.com/item/%E8%AF%81%E4%B9%A6%E6%9B%B4%E6%96%B0)时进行回溯。

证书类型（Certificate　Type）

含义：用来区别不同的实体。该项是必选的。

商户数据（Merchant　Data）

含义：包含支付[网关](https://baike.baidu.com/item/%E7%BD%91%E5%85%B3)需要的所有商户信息。

持卡人证书需求（Card　Cert　Required）

含义：显示支付网关是否支持与没有证书的持卡人进行交易。

SET 扩展（SETExtensions）

含义：列出支付网关支持的支付命令的 SET 信息扩展。

CRL 数据定义版本（Version）

含义：显示 CRL 的版本号。

CRL 的签发者（Issuer）

含义：指明签发 CRL 的CA 的甄别名。

CRL 发布时间（this　Update）预计下一个 CRL 更新时间（Next　Update）撤销证书信息目录（Revoked　Certificates） CRL 扩展（CRL　Extension）CA 的公钥标识（Authority　Key　Identifier）CRL 号（CRL　Number）

SSL证书种类

CFCA，GlobalSign，VeriSign ，Geotrust ，Thawte 。

域名型 https 证书（DVSSL）：信任等级一般，只需验证网站的真实性便可颁发证书保护网站；

企业型 https 证书（OVSSL）：信任等级强，须要验证企业的身份，审核严格，安全性更高；

增强型 https 证书（EVSSL）：信任等级最高，一般用于银行证券等金融机构，审核严格，安全性最高，同时可以激活绿色网址栏。