目录

[一、 基础 3](#_Toc379977791)

[1. 简介 3](#_Toc379977792)

[2. https 3](#_Toc379977793)

[3. 不安全因素 3](#_Toc379977794)

[二、 加密通道 3](#_Toc379977795)

[1. 原理 3](#_Toc379977796)

[三、 安全证书 4](#_Toc379977797)

[四、 通信过程 5](#_Toc379977798)

[1. 主要任务 5](#_Toc379977799)

[2. 具体流程 5](#_Toc379977800)

[五、 Tomcat下SSL双向认证 6](#_Toc379977801)

1. 基础
2. 简介

SSL(server socket layout)，是一种保证网络上的两个节点进行安全通信的协议。IETF(Internet engineering task force)组织对SSL进行了标准化，制订了RFC2246规范，并称为TLS(transport layout secure)。

1. https

SSL和TLS建立在TCP/IP协议的基础上，一些应用层协议，如HTTP和IMAP协议都可以采用SSL来保证通信的安全。建立在SSL协议上的HTTP被称为HTTPS协议。HTTP使用的默认端口为80，而HTTPS使用的默认端口为443.

|  |  |
| --- | --- |
| **协　议　层** | **协　　议** |
| **应用层** | HTTP、IMAP、NNTP、Telnet、FTP等 |
| **安全套接字层** | SSL、TLS |
| **传输层** | TCP |
| **网络层** | IP |

1. 不安全因素

当用户在网上商店购物时输入信用卡信息，进行网上支付交易时，存在以下不安全因素：

•用户的信用卡信息在网络上传输时有可能被他人截获。

•用户发送的信息在网络上传输时可能被非法篡改，数据完整性被破坏。

•用户正在访问的Web站点是个非法站点，专门从事网上欺诈活动，比如骗取客户的资金。

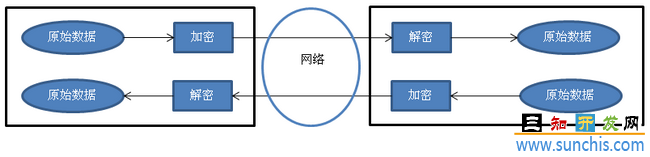
•非法用户访问一个合法的Web站点，该非法用户试图窃取Web站点的机密信息。

•其它的可能存在的不安全因素。

SSL采用加密技术来实现安全通信，保证通信数据的保密性和完整性，并且保证通信双方可以验证对方的身份。

1. 加密通道
2. 原理

加密技术的基本原理是：数据从一端发送到另一端时，发送者先对数据加密，然后再把它发送给接收者。这样，在网络上传输的是经过加密的数据。如果有人在网络上非法截获了这些数据，由于没有解密的密钥，就无法获得真正的原始数据。接收者收到加密的数据后，先对数据进行解密，然后再处理。客户机和服务器的加密通信需要在两端同时进行。下图显示了采用SSL加密的通信过程。



1. 安全证书

除了对数据加密通信，SSL还采用了身份认证机制，确保通信双方都可以验证对方的真实身份。它和现时日常生活中我们使用身份证来证明自己的身份很相似。比如你到银行去取钱，你自称自己叫张三，如何让银行相信你的真实身份呢？最有效的方法就是出示你的身份证。每个人都拥有唯一的身份证，这个身份证上记录了你的真实信息，身份证由国家权威机构颁发，不允许伪造。在身份证不能被别人假冒复制的前提下，只要你出示身份证，就可以证明你的确是你自称的那个人。

个人可以通过身份证来证明自己的身份，对于一个单位，比如商场，可以通过营业执照来表明身份，营业执照也是由国家权威机构颁发，不允许伪造，它保证了营业执照的可信性。

SSL通过安全证书来证明客户或服务器的身份。当客户通过安全的连接和服务器通信时，服务器会先向客户出示它的安全证书，这个证书声明该服务器是安全的，而且的确是这个服务器。每一个证书在全球范围内都是唯一的，其他非法服务器无法假冒原始服务器的身份。可以把安全证书比做电子身份证。

获取安全证书是一件麻烦的事情。一些服务器会向客户出示自己的安全证书，但另一方面，为了扩大客户群并且便于客户访问，许多服务器不要求客户出示安全证书。在某些情况下，服务器也会要求客户出示安全证书，以便核实客户的身份，这主要用于B2B（Business to Business）事务中。

获取安全证书有两种方式，一种方式是从权威机构购买证书，还有一种方式是创建自我签名的证书。

安全证书既包含了用于加密数据的密钥，又包含了用于证实身份的数字签名。安全证书采用公钥加密技术。公钥加密是指使用一对非对称的密钥进行加密或解密。每一对密钥由公钥和私钥组成。公钥被广泛发布。私钥是隐秘的，不公开。用公钥加密的数据只能够被私钥解密。反过来，使用私钥加密的数据只能被公钥解密。这个非对称的特性使得公钥加密很有用。

在安全证书中包含这一对非对称的密钥。只有安全证书的所有者才知道私钥。当通信方A将自己的安全证书发送给通信方B时，实际上发给通信方B的是公开密钥，接着通信方B可以向通信方A发送用公钥加密的数据，只有通信方A才能使用私钥对数据进行解密，从而获得通信方B发送的原始数据。

安全证书中的数字签名部分是通信方A的电子身份证。数字签名告诉通信方B该信息确实由通信方A发出，不是伪造的，也没有被篡改。

1. 通信过程
2. 主要任务

客户与服务器通信时，首先要进行SSL握手，SSL握手主要完成以下任务：

•协商使用的加密套件。加密套件中包括一组加密参数，这些参数指定了加密算法和密钥的长度等信息。

•验证对方的身份，此操作是可选的。

•确定使用的加密算法。

1. 具体流程

客户将自己的SSL版本号、加密参数、与SSL会话有关的数据及其他一些必要信息发送到服务器。

服务器将自己的SSL版本号、加密参数、与SSL会话有关的数据及其他一些必要信息发送给客户，同时发给客户的还有服务器的证书。如果服务器需要验证客户身份，服务器还会发出要求客户提供安全证书的请求。

客户端验证服务器证书，如果验证失败，就提示不能建立SSL连接。如果成功，那么继续下一步骤。

客户端为本次SSL会话生成预备主密码（pre-master secret），并将其用服务器公钥加密后发送给服务器。

如果服务器要求验证客户身份，客户端还要对另外一些数据签名后，将其与客户端证书一起发送给服务器。

如果服务器要求验证客户身份，则检查签署客户证书的CA（Certificate Authority，证书机构）是否可信。如果不在信任列表中，结束本次会话。如果检查通过，服务器用自己的私钥解密收到的预备主密码（pre-master secret），并用它通过某些算法生成本次会话的主密码（master secret）。

客户端与服务器端均使用此主密码（master secret）生成此次会话的会话密钥（对称密钥）。在双方SSL握手结束后传递任何消息均使用此会话密钥。这样做的主要原因是对称加密比非对称加密的运算量要低一个数量级以上，能够显著提高双方会话时的运算速度。

客户端通知服务器此后发送的消息都使用这个会话密钥进行加密，并通知服务器客户端已经完成本次SSL握手。

服务器通知客户端此后发送的消息都使用这个会话密钥进行加密，并通知客户端服务器已经完成本次SSL握手。

本次握手过程结束，SSL会话已经建立。在接下来的会话过程中，双方使用同一个会话密钥分别对发送和接收的信息进行加密和解密。

1. Tomcat下SSL双向认证

keytool -genkey -v -alias tomcat -keyalg RSA -keystore d:\tomcat.keystore -validity 365

keytool -genkey -v -alias mykey -keyalg RSA -storetype PKCS12 -keystore d:\mykey.p12

keytool -export -alias mykey -keystore d:\mykey.p12 -storetype PKCS12 -storepass 654321 -rfc -file d:\mykey.cer

keytool -import -v -file d:\mykey.cer -keystore d:\tomcat.keystore

keytool -list -keystore d:\tomcat.keystore