ICS 01.140.20
A 14

C A D A L 项 目 标 准

CADAL 20904—2012

数字图书馆安全传输标准

Digital Library Security Transport Standards

第一稿

2012-05-08 发布　　2012-05-09 实施

CADAL 项目管理中心　　发　布

目　　次

前　言

《数字图书馆安全标准规范集》包括以下 4 个部分：

——第 1 部分：数字图书馆数字对象存储安全规范；

——第 2 部分：数字图书馆访问安全规范；

——第 3 部分：数字图书馆数字资源长期保存规范；

——第 4 部分：数字图书馆安全传输标准。

本标准是其中的第 4 部分。

本部分的制定依据了《标准化工作导则》(GB/T 1.1—2009)　第 1 部分的要求。

本部分是由大学数字图书馆国际合作计划(CADAL)项目管理中心提出。

本标准由 CADAL 项目管理中心归口。

本部分起草单位：数字图书馆教育部工程研究中心。

本部分起草人：洪鑫、张寅、王宇奇。

引　　言

数字资源是将计算机技术、通信技术及多媒体技术相互融合而形成的以数字形式发布、存取、利用的信息资源总和，它的安全传输对于保护版权人权益以及支撑数字借阅服务模式十分重要。

本规范在 CADAL 项目数字资源安全传输的实践基础之上，编制了数字资源安全传输相关的规范，清晰说明了符合 CADAL 项目要求的数字资源安全传输方式，为数字资源安全传输提供了可参考的规范。

数字图书馆安全传输标准

1 范围

本标准确定了数字资源传输安全规范。

本标准规定了数字资源在安全传输过程中所要遵循的协议，以及整个过程中用到的加密算法。

本标准适用于 CADAL 项目中对数字资源的安全管理，适用于对数字资源的传输过程进行安全管理。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

HTTP Over TLS

The TLS Protocol

The Secure Sockets Layer (SSL) Protocol Version 3.0

3 术语和定义

3.1 超文本传输协议 HyperText Transfer Protocol 缩写：HTTP

超文本传输协议是一种详细规定了浏览器和万维网服务器之间互相通信的规则，通过因特网传送万维网文档的数据传送协议。

3.2 安全套接层的超文本传输协议 HyperText Transfer Protocol over Secure Socket Layer 缩写：HTTPS

安全套接层的超文本传输协议是以安全为目标的 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。它是一个抽象标识符体系(URI scheme)，句法类同 http 体系。用于安全的 HTTP 数据传输。

3.3 安全套接层 Secure Socket Layer 缩写：SSL

用以保障在 Internet 上数据传输之安全，利用数据加密(encryption)技术，可确保数据在网络的传输过程中不会被截取及窃听。

3.4 安全传输层协议 Transport Layer Security 缩写：TLS

用于在两个通信应用程序之间提供保密性和数据完整性。该协议由两层组成：TLS 记录协议(TLS Record)和 TLS 握手协议(TLS Handshake)。较低的层为 TLS 记录协议，位于某个可靠的传输协议(例如 TCP)上面。

4 安全传输标准

数据的安全传输是数字图书馆需要解决的一个重要问题。数字资源在传输过程中如果被恶意用户盗用，就会损害图书版权所有人的权益，同时也会影响数字图书馆提供给用户的服务质量。本标准采用的安全协议包括 HTTPS、SSL、TLS，以及自定义的加密算法。下面具体介绍三个协议以及详细的加密过程。

4.1 HTTPS 协议 HyperText Transfer Protocol over Secure Socket Layer

HTTPS 是以安全为目标的 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。它是一个抽象标识符体系(URI scheme)，句法类同 http 体系。用于安全的 HTTP 数据传输。HTTPS 存在不同于 HTTP 的默认端口及一个加密/身份验证层(在 HTTP 与 TCP 之间)。这个系统提供了身份验证与加密通讯方法，现在它被广泛用于万维网上安全敏感的通信。

HTTPS 通信时序如图 1 所示。

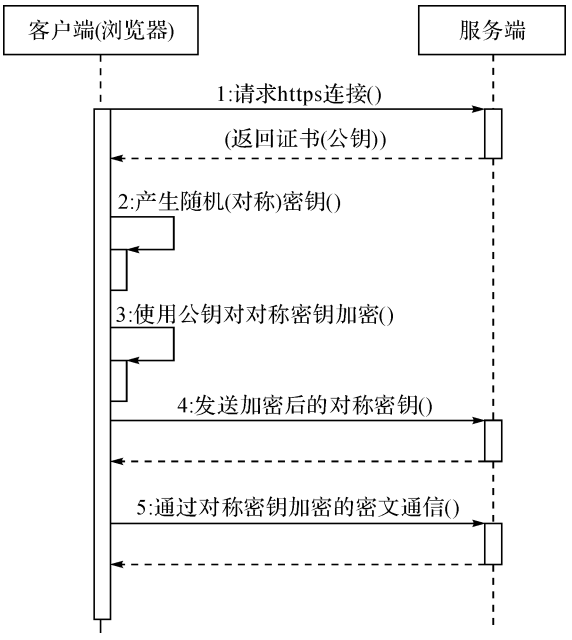

图 1 HTTPS 通信时序图

采用 HTTPS 的服务器必须从 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。服务端和客户端之间的所有通信，都是加密的。具体讲，是客户端产生一个对称的密钥，通过服务器的证书来交换密钥，即一般意义上的握手过程。接下来所有的信息往来都是加密的。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。

4.2 SSL 协议 Secure Socket Layer

SSL 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(encryption)技术，可确保数据在网络的传输过程中不会被截取及窃听。SSL 协议位于 TCP/IP 协议与各种应用层协议之间，为数据通信提供安全支持。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、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。HTTPS 协议栈如图 2 所示。

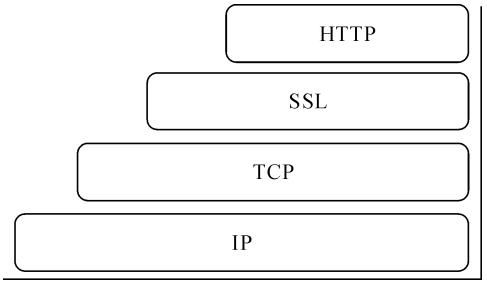

图 2 HTTPS 协议栈

SSL 协议负责实现 HTTPS 协议栈中的加密层。它的工作流程如下：

(1)客户端向服务器发出请求，询问对方支持的对称加密算法和非对称加密算法；服务器回应自己支持的算法。

(2)客户端选择双方都支持的加密算法，并且请求服务器出示自己的证书；服务器回应自己的证书。

(3)客户端随机产生一个用于本次会话的对称加密的钥匙，并使用服务器证书中附带的公钥对该钥匙进行加密后传递给服务器；服务器为本次会话保持该对称加密的钥匙。非对称加密让任何客户端都可以与服务器进行加密会话。

(4)客户端使用对称加密的钥匙对请求消息加密后传送给服务器，服务器使用该对称加密的钥匙进行解密；服务器使用对称加密的钥匙对响应消息加密后传送给客户端，客户端使用该对称加密的钥匙进行解密。对称加密提高了加密速度。

SSL 协议能够认证用户和服务器，确保数据发送到正确的客户机和服务器；加密数据以防止数据中途被窃取；维护数据的完整性，确保数据在传输过程中不被改变。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。

SSL 协议通信过程如图 3 所示。

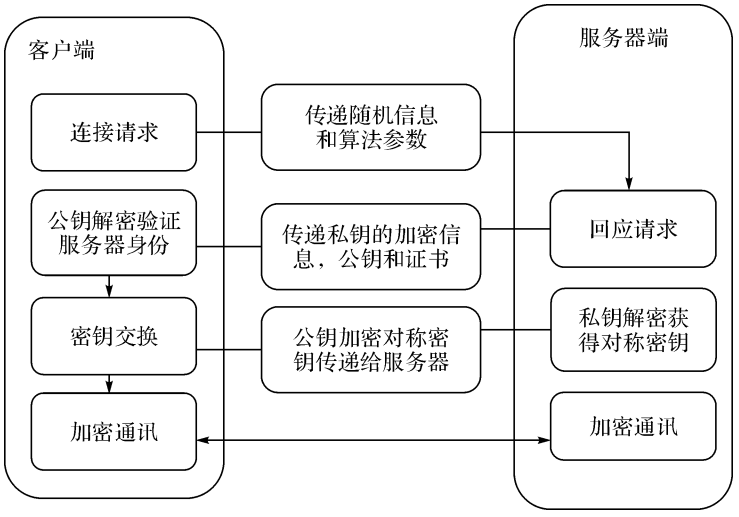

图 3　SSL 协议通信过程示意图

4.3　TLS 协议　Transport Layer Security

TLS 协议即安全传输层协议，用于在两个通信应用程序之间提供保密性和数据完整性。该协议由两层组成：TLS 记录协议(TLS Record)和 TLS 握手协议(TLS Handshake)。较低的层为 TLS 记录协议，位于某个可靠的传输协议(例如 TCP)上面。

TLS 记录协议提供的连接安全性具有两个基本特性：私有——对称加密用以数据加密(DES、RC4 等)。对称加密所产生的密钥对每个连接都是唯一的，且此密钥基于另一个协议(如握手协议)协商。记录协议也可以不加密使用。可靠——信息传输包括使用密钥的 MAC 进行信息完整性检查。安全哈希功能(SHA、MD5 等)用于 MAC 计算。记录协议在没有 MAC 的情况下也能操作，但一般只能用于这种模式，即有另一个协议正在使用记录协议传输协商安全参数。

TLS 握手协议提供的连接安全具有三个基本属性：可以使用非对称的，或公共密钥的密码术来认证对等方的身份，该认证是可选的，但至少需要一个结点方；共享加密密钥的协商是安全的，对偷窃者来说协商加密是难以获得的，此外经过认证过的连接不能获得加密，即使是进入连接中间的攻击者也不能；协商是可靠的，没有经过通信方成员的检测，任何攻击者都不能修改通信协商。

TLS 包含三个基本阶段：对等协商支援的密钥算法；基于私钥加密交换公钥，基于 PKI 证书的身份认证；基于公钥加密的数据传输保密。TLS 的最大优势就在于：TLS 是独立于应用协议。高层协议可以透明地分布在 TLS 协议上面。然而，TLS 标准并没有规定应用程序如何在 TLS 上增加安全性；它把如何启动 TLS 握手协议以及如何解释交换的认证证书的决定权留给协议的设计者和实施者来判断。

TLS 相对于 SSL 增强的内容：

(1)更安全的 MAC 算法。

(2)更严密的警报。

(3)“灰色区域”规范的更明确的定义。

TLS 对于安全性的改进：

(1)对于消息认证使用密钥散列法：TLS 使用“消息认证代码的密钥散列法(HMAC)”，当记录在开放的网络上传送时，该代码确保记录不会被变更。

(2)增强的伪随机功能(PRF)：PRF 生成密钥数据。在 TLS 中，HMAC 定义 PRF。PRF 使用两种散列算法保证其安全性。如果任一算法暴露了，只要第二种算法未暴露，则数据仍然是安全的。

(3)改进的已完成消息验证：TLS 对两个端点提供已完成的消息，并将此已完成的消息基于 PRF 和 HMAC 值之上。

(4)一致证书处理：TLS 试图指定必须在 TLS 之间实现交换的证书类型。

(5)特定警报消息：TLS 提供更多的特定和附加警报，以指示任一会话端点检测到的问题。TLS 还对何时应该发送某些警报进行记录。

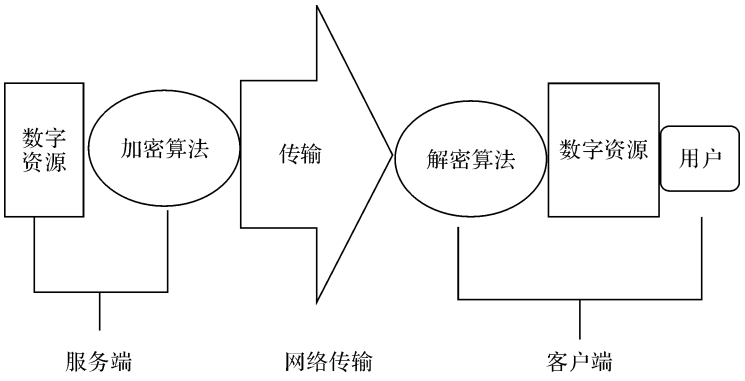

图 4　图书加密传输过程

4.4　加密传输

图 4 展示了图书加密传输的具体过程：客户端向服务端请求资源时，服务端先到资源存储器上得到数字资源，然后利用自定义加密算法将得到的数据加密，将加密后的数据通过网络传输到客户端，客户端再通过相应的解密算法将接收到的数据解密，整个过程完成了数字资源的安全传输。

本标准推荐的加密算法有 DES、RSA 和 MD5。下面是对这几个算法的介绍。

DES:DES 的原始思想可以参照二战德国的恩格玛机，其基本思想大致相同。传统的密码加密都是由古代的循环移位思想而来，恩格玛机在这个基础之上进行了扩散模糊。但是本质原理都是一样的。现代 DES 在二进制级别做着同样的事：替代模糊，增加分析的难度。

DES 的加密原理：DES 使用一个 56 位的密钥以及附加的 8 位奇偶校验位，产生最大 64 位的分组大小。这是一个迭代的分组密码，使用称为 Feistel 的技术，将其中加密的文本块分成两半。使用子密钥对其中一半应用循环功能，然后将输出与另一半进行“异或”运算；接着交换这两半，这一过程会继续下去，但最后一个循环不交换。DES 使用 16 个循环，使用异或、置换、代换、移位操作四种基本运算。

RSA:RSA 公钥加密算法是 1977 年由 Ron Rivest、Adi Shamir 和 Leonard Adleman 在美国麻省理工学院开发的。RSA 的名称取自开发者的名字。RSA 是目前最有影响力的公钥加密算法,它能够抵抗到目前为止已知的所有密码攻击,已被 ISO 推荐为公钥数据加密标准。RSA 是公开密钥密码体制。所谓的公开密钥密码体制就是使用不同的加密密钥与解密密钥,是一种"由已知加密密钥推导出解密密钥在计算上是不可行的"密码体制。

RSA 算法是第一个能同时用于加密和数字签名的算法,也易于理解和操作。RSA 是被研究得最为广泛的公钥算法,从提出到现在的三十多年里,经历了各种攻击的考验,逐渐为人们接受,普遍认为是目前最优秀的公钥方案之一。

MD5:Message Digest Algorithm MD5 为计算机安全领域广泛使用的一种散列函数,用以提供消息的完整性保护,用于确保信息传输完整一致,是计算机广泛使用的杂凑算法之一,它的作用是让大容量信息在用数字签名软件签署私人密钥前被"压缩"成一种保密的格式。

MD5 以 512 位分组来处理输入的信息,且每一分组又被划分为 16 个 32 位子分组,经过了一系列的处理后,算法的输出由 4 个 32 位分组组成,将这 4 个 32 位分组级联后将生成一个 128 位散列值。

参 考 文 献

[1] RESCORLA E. (2000). RFC 2818 HTTP Over TLS[online]fags. org[S/OL]. http: //www. faqs. org/rfcs/rfc2818. html [Accessed 检索时间].

[2] DIERKS T, ALLEN C. (1999). RFC 2246 The TLS Protocol[S/OL]. http: //www. ietf. org/rfc/rfc2246. txt.

[3] BARNES R. THOMSON M, PIRONTI A, et al. RFC 7568 The Secure Sockets Layer(SSL) Protocol Version 3. 0[S/OL]. http: //tools. ietf. org/html/rfc7568.