

SSL/TLS算法流程解析

SSL/TLS 早已不是陌生的词汇，然而其原理及细则却不是太容易记住。本文将试图通过一些简单图示呈现其流程原理，希望读者有所收获。

一、相关版本

V e r s i o n	Source	Description	Browser Support
S S L v 2 . 0	Vendor Standard (from Netscape Corp.) [SSL2]	First SSL protocol for which implementations exist	- NS Navigat or 1.x/2.x - MS IE 3.x -

公告



美码师，老码农一枚，
喜欢聊聊代码，唠唠职场
故事，爱技术也爱生活
欢迎关注我的公众号



昵称：美码师
园龄：8年5个月
粉丝：90
关注：7
+加关注

< 2019年6月 >						
日	一	二	三	四	五	六
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

搜索

 找找看

常用链接

[我的随笔](#)[我的评论](#)[我的参与](#)[最新评论](#)[我的标签](#)

随笔分类

[0.JAVA技术\(39\)](#)[1.架构设计\(3\)](#)[2.安全技术\(9\)](#)[3.前端技术\(2\)](#)[4.测试技术\(2\)](#)[5.数据库中间件\(21\)](#)[7.工具技巧\(9\)](#)[8.构建技术\(5\)](#)[9.基础原理\(2\)](#)[O.开放平台\(3\)](#)[P.行业相关\(1\)](#)[S.敏捷管理](#)[Z.心得杂谈\(9\)](#)

随笔档案

[2019年5月 \(2\)](#)[2019年4月 \(4\)](#)[2019年3月 \(7\)](#)[2019年2月 \(2\)](#)[2018年12月 \(2\)](#)[2018年11月 \(6\)](#)[2018年9月 \(3\)](#)

			Lynx/2.8+OpenSSL
SSLv3.0	Expired Internet Draft (from Netscape Corp.) [SSL3]	Revisions to prevent specific security attacks, add non-RSA ciphers and support for certificate chains	- NS Navigator 2.x/3.x/4.x - MS IE 3.x/4.x - Lynx/2.8+OpenSSL
TLSv1.0	Proposed Internet Standard (from IETF) [TLS1]	Revision of SSL 3.0 to update the MAC layer to HMAC , add block padding for block ciphers, message order standardization and more alert messages.	- Lynx/2.8+OpenSSL

SSL全称为 Socket Security Layer，TLS全称为Transport Layer Security，这两者没有本质的区别，都是做的传输层之上的加密(介于传输层及应用层之间)。TLS是后续SSL版本分支的名称，花费长时间去争论两者的优劣没有意义。目前TLS最新版本为 TLS1.2(也称为SSL3.3)

二、SSL/TLS 解决的问题

信息被窃听(wiretap)，第三方随时随地获得通讯内容；

SSL/TLS 实现了传输信息的加密。

数据被篡改(tampering)，第三方可修改传输中的数据；

SSL/TLS 实现了数据签名及校验。

身份被冒充(pretending)，第三方可冒充通讯者身份传输数据；

SSL/TLS 采用了CA数字证书认证机制。

三、握手阶段

简单点说，SSL/TLS对于传输层的加密是通过动态密钥对数据进行加密实现的，而动态密钥则通过握手流程协商制定；为了保证动态密钥的安全性，其中免不了使用公钥加密算法(非对称)、数字证书签名等技术手段。

一个SSL/TLS 握手过程需要协商的信息包括：

- 1 协议的版本号；
- 2 加密算法，包括非对称加密算法、动态密钥算法；
- 3 数字证书，传输双方通过交换证书及签名校验对彼此进行鉴权；
- 4 动态密钥，传输数据过程使用该密钥进行对称加解密，该密钥通过非对称密钥进行加密传输。

四、流程解析

一个典型的SSL/TLS 握手流程包括双向认证，如下所示：

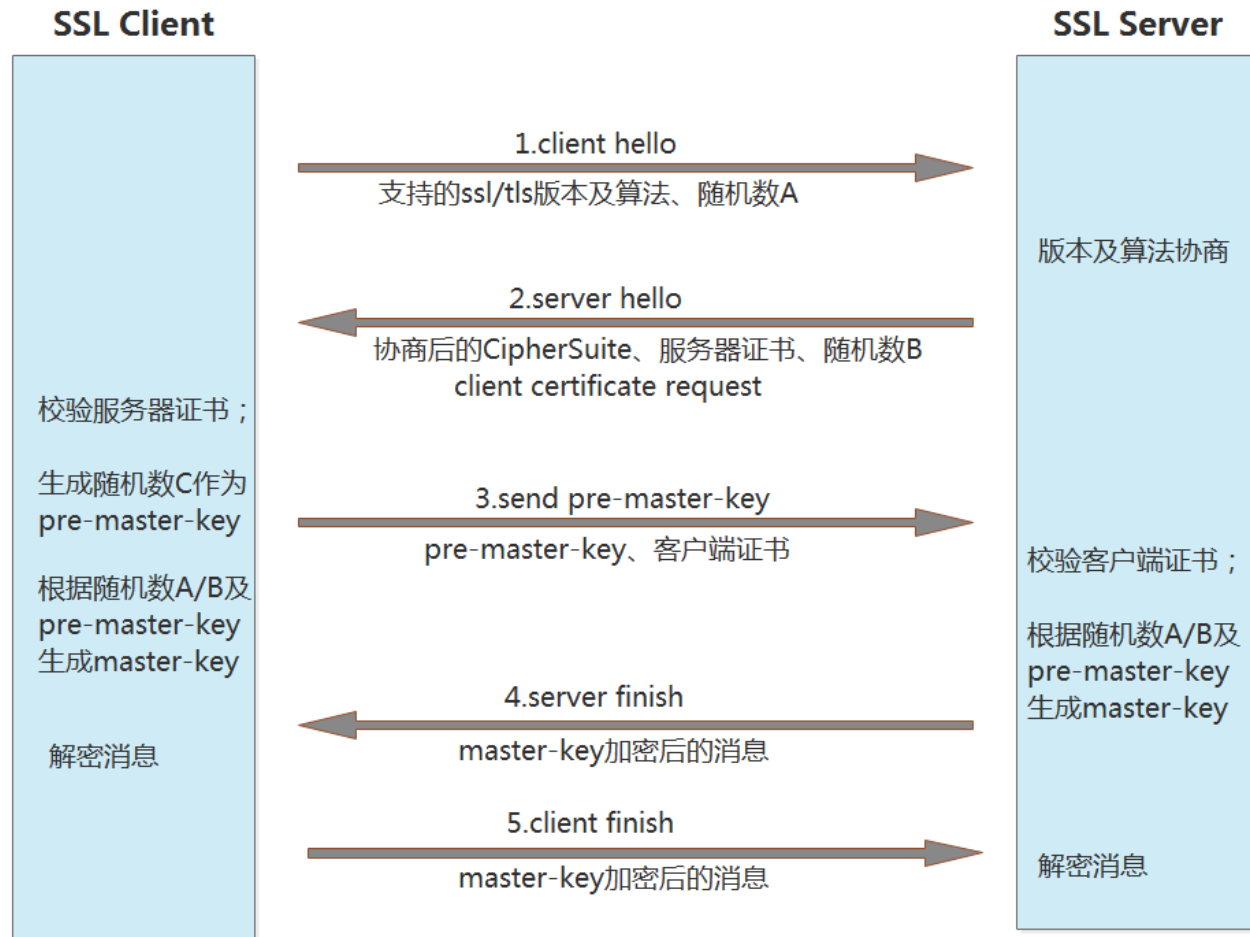
2018年8月 (4)
2018年7月 (8)
2018年6月 (2)
2018年5月 (4)
2018年3月 (3)
2018年2月 (5)
2017年10月 (1)
2017年8月 (1)
2017年6月 (1)
2017年3月 (2)
2017年2月 (1)
2017年1月 (4)
2016年12月 (5)
2016年11月 (1)
2016年10月 (4)
2016年9月 (4)
2016年8月 (1)
2016年4月 (1)
2015年12月 (1)
2015年9月 (3)
2015年8月 (1)
2015年7月 (4)
2015年6月 (1)
2015年4月 (3)
2015年3月 (3)
2011年11月 (2)
2011年10月 (2)
2011年9月 (5)

links

ascii 图表
ascii艺术字

最新评论

1. Re:Java条形码生成技术-Barcode4j



1. 客户端发出一个 client hello 消息，携带的信息包括：
所支持的SSL/TLS 版本列表；支持的与加密算法；所支持的数据压缩方法；随机数A；
2. 服务端响应一个 server hello 消息，携带的信息包括：
协商采用的SSL/TLS 版本号；会话ID；随机数B；服务端数字证书 serverCA；

怎么隐藏条形码下方的文本?或者设置为自定义文本?

--习惯沉淀

2. Re:成为高手前必懂的TCP干货

@
海向

--美码师

3. Re:成为高手前必懂的TCP干货

6

--海向

4. Re:redis通过pipeline提升吞吐量

@ericlfredis-stat , 可参考这里的: ...

--美码师

5. Re:redis通过pipeline提升吞吐量

楼主, 请教下性能测评是使用的什么工具? 3Q!

--ericlf

阅读排行榜

1. 使用 openssl 生成证书(62414)
2. mysql 索引过长1071-max key length is 767 byte(56642)
3. Java条形码生成技术-Barcode4j(36607)
4. MQTT服务器搭建-mosquitto1.4.4安装指南(26265)
5. 使用keytool 生成证书(18704)

评论排行榜

1. 情人节, 送女友一桶代码可否? (36)
2. 软能力那点事, 你知多少(18)
3. 老兵的十年职场之路(一)(9)
4. redis通过pipeline提升吞吐量(6)
5. MQTT服务器搭建-mosquitto1.4.4安装指南(6)

由于双向认证需求，服务端需要对客户端进行认证，会同时发送一个 **client certificate request**，表示请求客户端的证书；

3. 客户端校验服务端的数字证书；校验通过之后发送随机数C，该随机数称为pre-master-key，使用数字证书中的公钥加密后发出；

由于服务端发起了 **client certificate request**，客户端使用私钥加密一个随机数 **clientRandom**随客户端的证书 **clientCA**一并发出；

4. 服务端校验客户端的证书，并成功将客户端加密的随机数clientRandom 解密；

根据 随机数A/随机数B/随机数C(pre-master-key) 产生动态密钥 master-key，加密一个finish 消息发至客户端；

5. 客户端根据 同样的随机数和算法 生成master-key，加密一个finish 消息发送至服务端；

6. 服务端和客户端分别解密成功，至此握手完成，之后的数据包均采用master-key进行加密传输。

五、要点解析

双向认证和单向认证

双向认证更好的解决了身份冒充问题，服务端提供证书的同时要求对客户端身份进行认证；然而在一些常见的应用场景下往往只有单向认证，如采用https网站只要求客户端(浏览器)对服务端的证书进行认证。

在单向认证场景下，握手阶段2服务端不会发出 client certificate request，之后服务端也不需要校验客户端证书；

在双向认证场景下，客户端如果无法提供证书，会发出 no digital certificate alert 的警告信息，此时可能导致握手失败(根据服务端策略而定)；

随机数的使用

推荐排行榜

1. 软能力那点事，你知多少(17)
2. 情人节，送女友一桶代码可否？(15)
3. 老兵的十年职场之路(二)(9)
4. 回顾下自己都写了什么(9)
5. 老兵的十年职场之路(一)(8)

由于数字证书是静态的，因此要求使用随机因素来保证协商密钥的随机性；对于RSA 算法来说，pre-master-key本身就是一个随机数，再加上hello消息中的随机，三个随机数通过一个密钥导出器最终导出一个对称密钥。

之所以采用 pre-master-key 机制是因为SSL协议不信任每个主机都能产生完全随机的随机数，如果 pre-master-key 不随机，那么被猜出来的风险就很大，于是仅仅使用 pre-master-secret作为密钥不合适，需要引入新的随机因素，也就是同时结合hello消息中的双向随机数。

会话密钥重用

SSL/TLS握手过程比较繁琐，同时非对称加解密性能比对称密钥要差得多；如果每次重建连接时都需要进行一次握手会产生较大开销，因此有必要实现会话的重用以提高性能。

常用的方式包括：

SessionID(RFC 5246)，客户端和服务端同时维护一个会话ID和会话数据状态；重建连接时双方根据sessionID找到之前的会话密钥实现重用；

SessionTicket(RFC 5077)，由服务端根据会话状态生成一个加密的ticket，并将key也发给客户端保证两端都可以对其进行解密。该机制相较sessionID的方式更加轻量级，服务端不需要存储会话状态数据，可减轻一定压力。

证书的校验

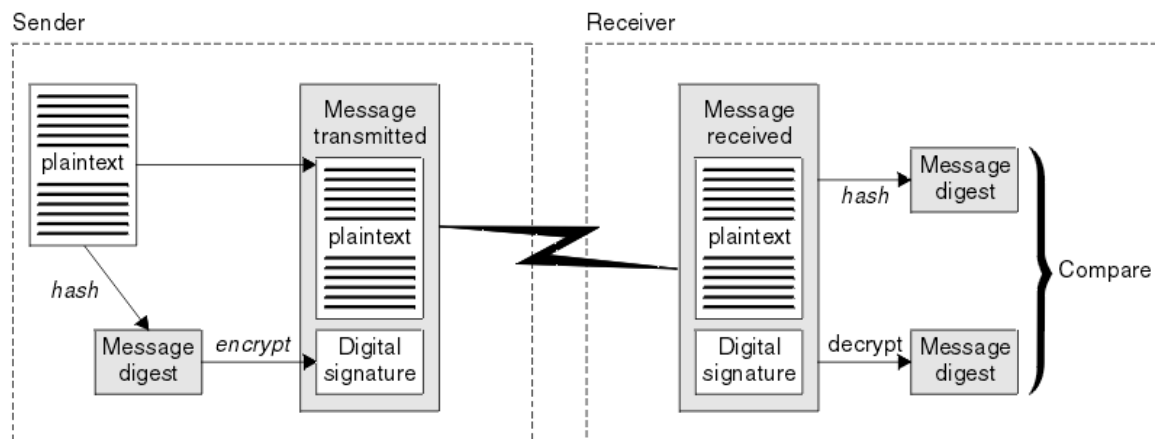
1. 检查数字签名；

数字签名通过数字摘要算法生成并通过私钥加密传输，对端公钥解密；

2. CA链授权检查；

3. 证书过期及激活时间检查；

数字摘要的计算图示



关于Server Name Indication

在普通 SSL/TLS握手的过程中，客户端发送的信息之中不包括服务器的域名；因此理论上服务器只能包含一个域名，否则会分不清应该向客户端提供哪一个域名的数字证书。在后续TLS的版本中实现了SNI(Server Name Indication) 扩展，用于支持一台服务器主机需服务多个域名的场景。

由客户端请求时发送指定的域名，服务器据此选择相应证书完成握手。

六、参考文档

[阮一峰 SSL/TLS协议运行机制的概述](#)

[An overview of the SSL or TLS handshake](#)

作者: [zale](#)



出处: <http://www.cnblogs.com/littleatp/>, 如果喜欢我的文章, 请关注我的公众号

本文版权归作者和博客园共有, 欢迎转载, 但未经作者同意必须保留此段声明, 且在文章页面明显位置给出 [原文链接](#) 如有问题, 可留言咨询.

分类: [2.安全技术](#)

标签: [ssl](#), [tls](#)

好文要顶

关注我

收藏该文



美码师

[关注 - 7](#)

[粉丝 - 90](#)

[+加关注](#)

0

推荐

0

反对

« 上一篇: [制作简易的启动脚本](#)

» 下一篇: [实现iul 日志重定向到 slf4j](#)

posted @ 2016-12-25 20:56 美码师 阅读(6966) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#).

【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码

【前端】SpreadJS表格控件, 可嵌入系统开发的在线Excel

【培训】从Java菜鸟到大牛的成长秘籍 6.18冰点价

【推荐】程序员问答平台, 解决您开发中遇到的技术难题

相关博文：

- [SSL握手流程](#)
- [SSL协议\(HTTPS\) 握手、工作流程详解\(双向HTTPS流程\)](#)
- [HTTPS,SSL,TLS理解和验证流程](#)
- [SSL/TLS概述](#)
- [SSL协议\(HTTPS\) 握手、工作流程详解\(双向HTTPS流程\)](#)

最新新闻：

- [天文学家称月球最大的陨石坑下方隐藏着神秘物质](#)
 - [可循环利用食品包装透明薄膜问世](#)
 - [亚马逊成全球最具价值品牌 阿里腾讯进前10强](#)
 - [学问经得起时间考验的傅立叶](#)
 - [美司法部警告科技巨头：没商业意义的收购视为垄断](#)
- » [更多新闻...](#)



Copyright ©2019 美码师