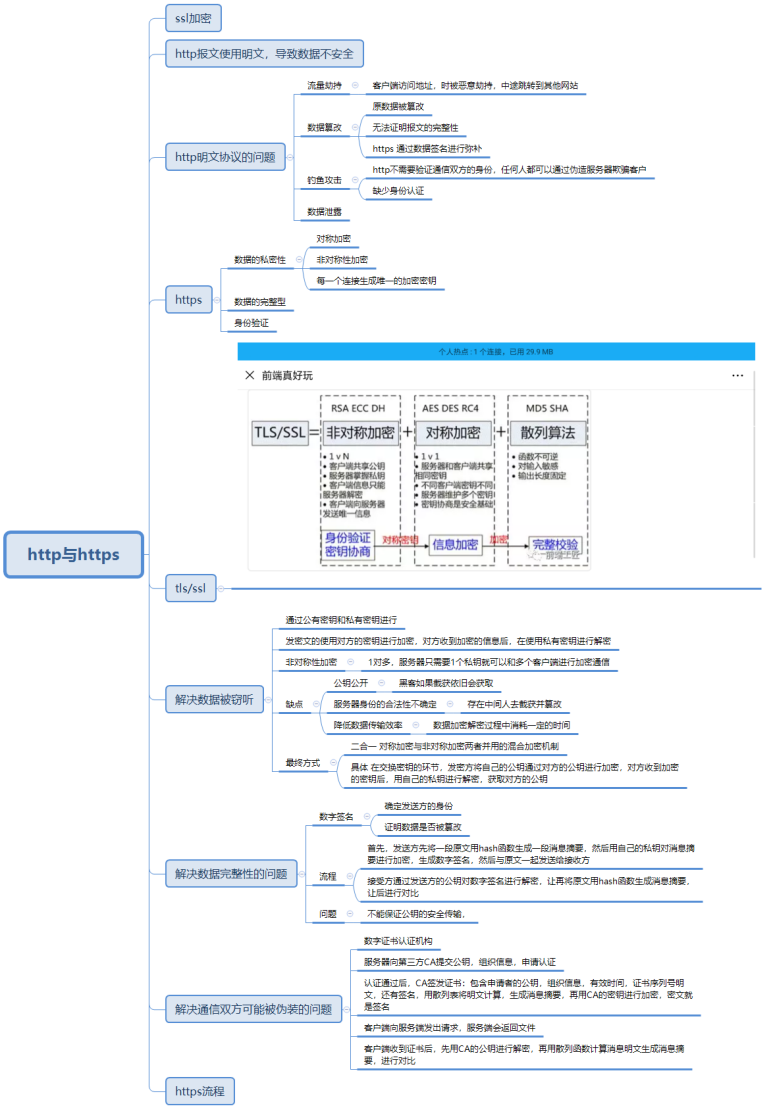
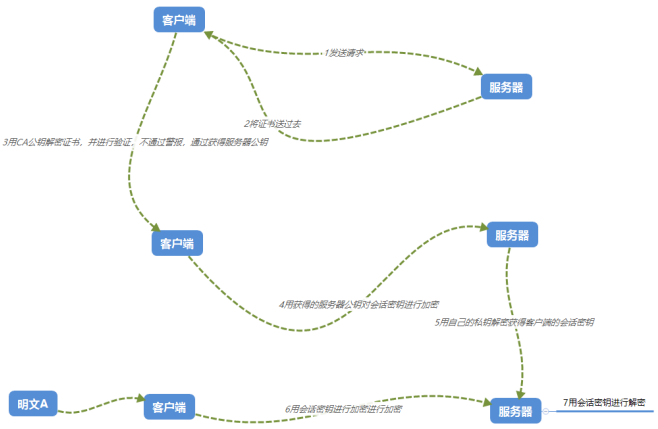
http与https


http与https
ssl加密
http报文使用明文，导致数据不安全
http明文协议的问题
流量劫持
客户端访问地址，时被恶意劫持，中途跳转到其他网站
数据篡改
原数据被篡改
无法证明报文的完整性
https 通过数据签名进行弥补
钓鱼攻击
http不需要验证通信双方的身份，任何人都可以通过伪造服务器欺骗客户
缺少身份认证
数据泄露
https
数据的私密性
对称加密
非对称性加密
每一个连接生成唯一的加密密钥
数据的完整型
身份验证
tls/ssl
个人热点:1个连接，已用 29.9 MB
前端真好玩
RSA ECC DH
AES DES RC4
MD5 SHA
TLS/SSL
非对称加密
对称加密
散列算法
1 v N
客户端共享公钥
服务器掌握私钥
客户端信息只能服务器解密
客户端向服务器发送唯一信息
1 v 1
服务器和客户端共享相同密钥
不同客户端密钥不同
服务器维护多个密钥
密钥协商是安全基础
函数不可逆
对输入敏感
输出长度固定
身份验证
密钥协商
对称密钥
信息加密
加密
完整校验
一般使用这
解决数据被窃听
通过公有密钥和私有密钥进行
发密文的使用对方的密钥进行加密，对方收到加密的信息后，在使用私有密钥进行解密
非对称性加密
1对多，服务器只需要1个私钥就可以和多个客户端进行加密通信
缺点
公钥公开
黑客如果截获依旧会获取
服务器身份的合法性不确定
存在中间人去截获并篡改
降低数据传输效率
数据加密解密过程中消耗一定的时间
最终方式
二合一 对称加密与非对称加密两者并用的混合加密机制
具体 在交换密钥的环节，发密方将自己的公钥通过对方的公钥进行加密，对方收到加密的密钥后，用自己的私钥进行解密，获取对方的公钥
解决数据完整性的问题
数字签名
确定发送方的身份
证明数据是否被篡改
流程
首先，发送方先将一段原文用hash函数生成一段消息摘要，然后用自己的私钥对消息摘要进行加密，生成数字签名，然后与原文一起发送给接收方
接受方通过发送方的公钥对数字签名进行解密，让再将原文用hash函数生成消息摘要，让后进行对比
问题
不能保证公钥的安全传输，
解决通信双方可能被伪装的问题
数字证书认证机构
服务器向第三方CA提交公钥，组织信息，申请认证
认证通过后，CA签发证书：包含申请者的公钥，组织信息，有效时间，证书序列号明文，还有签名，用散列表明明文计算，生成消息摘要，再用CA的密钥进行加密，密文就是签名
客户端向服务端发出请求，服务端会返回文件
客户端收到证书后，先用CA的公钥进行解密，再用散列函数计算消息明文生成消息摘要，进行对比
https流程


客户端
服务器
1发送请求
2将证书送过去
3用CA公钥解密证书，并进行验证，不通过警报，通过获得服务器公钥
客户端
服务器
4用获得的服务器公钥对会话密钥进行加密
5用自己的私钥解密获得客户端的会话密钥
明文A
客户端
6用会话密钥进行加密进行加密
服务器
7用会话密钥进行解密

1. ssl加密

2. http报文使用明文，导致数据不安全

3. http明文协议的问题

   3.1. 流量劫持

      3.1.1. 客户端访问地址，时被恶意劫持，中途跳转到其他网站

   3.2. 数据篡改

      3.2.1. 原数据被篡改

      3.2.2. 无法证明报文的完整性

      3.2.3. https 通过数据签名进行弥补

   3.3. 钓鱼攻击

      3.3.1. http不需要验证通信双方的身份，任何人都可以通过伪造服务器欺骗客户

      3.3.2. 缺少身份认证

   3.4. 数据泄露

4. https

   4.1. 数据的私密性

      4.1.1. 对称加密

      4.1.2. 非对称性加密

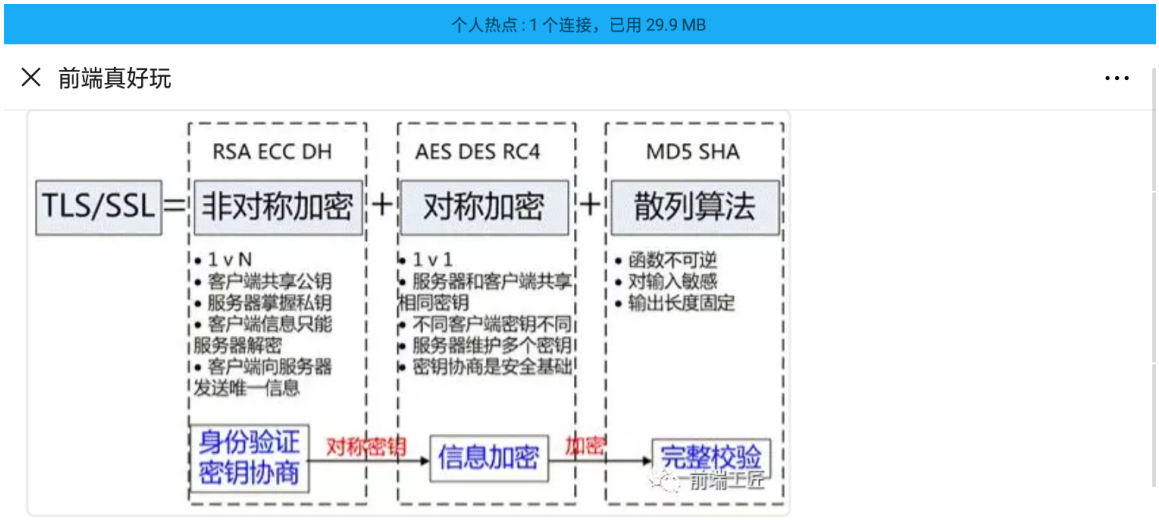
      4.1.3. 每一个连接生成唯一的加密密钥

4.2. 数据的完整型

4.3. 身份验证

5. tls/ssl

5.1.



6. 解决数据被窃听

6.1. 通过公有密钥和私有密钥进行

6.2. 发密文的使用对方的密钥进行加密，对方收到加密的信息后，在使用私有密钥进行解密

6.3. 非对称性加密

6.3.1. 1对多，服务器只需要1个私钥就可以和多个客户端进行加密通信

6.4. 缺点

6.4.1. 公钥公开

黑客如果截获依旧会获取

6.4.2. 服务器身份的合法性不确定

存在中间人去截获并篡改

6.4.3. 降低数据传输效率

数据加密解密过程中消耗一定的时间

6.5. 最终方式

6.5.1. 二合一 对称加密与非对称加密两者并用的混合加密机制

6.5.2. 具体

在交换密钥的环节，发密方将自己的公钥通过对方的公钥进行加密，对方收到加密的密钥后，用自己的私钥进行解密，获取对方的公钥

7. 解决数据完整性的问题

7.1. 数字签名

7.1.1. 确定发送方的身份

7.1.2. 证明数据是否被篡改

7.2. 流程

7.2.1. 首先，发送方先将一段原文用hash函数生成一段消息摘要，然后用自己的私钥对消息摘要进行加密，生成数字签名，然后与原文一起发送给接收方

7.2.2. 接受方通过发送方的公钥对数字签名进行解密，让再将原文用hash函数生成消息摘要，让后进行对比

7.3. 问题

7.3.1. 不能保证公钥的安全传输，

8. 解决通信双方可能被伪装的问题

8.1. 数字证书认证机构

8.2. 服务器向第三方CA提交公钥，组织信息，申请认证

8.3. 认证通过后，CA签发证书：包含申请者的公钥，组织信息，有效时间，证书序列号明文，还有签名，用散列表将明文计算，生成消息摘要，再用CA的密钥进行加密，密文就是签名

8.4. 客户端向服务端发出请求，服务端会返回文件

8.5. 客户端收到证书后，先用CA的公钥进行解密，再用散列函数计算消息明文生成消息摘要，进行对比

9. https流程

客户端

参见: 服务器 (1发送请求), 服务器 (2将证书送过去), 客户端 (3用CA公钥解密证书, 并进行验证, 不通过警报, 通过获得服务器公钥)

服务器

参见: 客户端 (1发送请求), 客户端 (2将证书送过去)

客户端

参见: 客户端 (3用CA公钥解密证书, 并进行验证, 不通过警报, 通过获得服务器公钥), 服务器 (4用获得的服务器公钥对会话密钥进行加密)

服务器

参见: 客户端 (4用获得的服务器公钥对会话密钥进行加密), 服务器 (5用自己的私钥解密获得客户端的会话密钥)

服务器

参见: 服务器 (5用自己的私钥解密获得客户端的会话密钥), 客户端 (6用会话密钥进行加密进行加密)

7用会话密钥进行解密

客户端

参见: 明文A, 服务器 (6用会话密钥进行加密进行加密)

明文A

参见: 客户端