HTTPS、CA、私钥、公钥

**一、对称加密**

对称加密是指双方持有相同的密钥进行通信，加密速度快，但是有一个安全问题，双方怎样获得相同的密钥？常见的对称加密算法有DES、3DES、AES等

**二、非对称加密**

非对称加密，又称为公开密钥加密，是为了解决对称加密中的安全问题而诞生，一个称为公开密钥(public key)，即公钥；另一个称为私钥(private key)，即私钥。但是它的加密速度相对于对称加密来说很慢。

* 公钥(public key)是对外开放的，私钥(private key)是自己拥有的。
* 公钥(public key)加密的数据，只能用私钥(private key)解密。
* 私钥(private key)加密的数据，只能用公钥(public key)解密。

**三、信息安全问题**

在信息安全性问题中，要做到三点才能保证信息的安全：

* 信息的保密性
* 信息的完整性
* 身份识别

**3.1 信息的保密性(加密算法)**

信息的保密性我们可以使用【**对称加密**】和【**非对称加密**】来完成；通用的方法是使用非对称加密+对称加密来完成。客户端使用公钥对【**对称加密**】的密钥进行加密，然后传递给服务端，服务端使用私钥进行解密确认密钥，开始传输数据。

Diagram

Description automatically generated

#### 3.2 信息的完整性(数字签名)

信息传输的途中，我们的信息很有可能被第三方劫持篡改，所以我们需要保证信息的完整性，通用方法是使用散列算法如SHA1，MD5将传输内容hash，获得hash值，即摘要(Digest)。客户端使用服务端的公钥对Digest和信息内容进行加密，然后传输给服务端，服务端使用私钥进行解密获得原始内容和摘要值，这时服务端使用相同的hash算法对原始内容进行hash，然后与摘要值比对，如果一致，说明信息是完整的。

Diagram

Description automatically generated

1. 对发送的内容进行 一次Hash，生成信件的摘要（digest）

Diagram

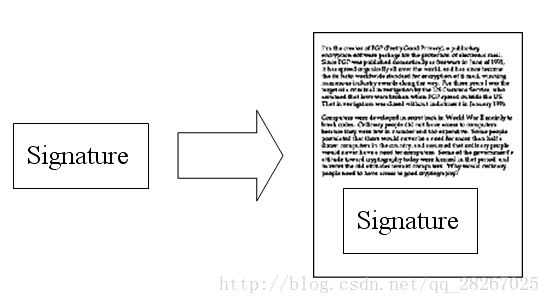
Description automatically generated

b) 使用 私钥，对这个Digest加密，生成”数字签名”（signature）

Logo, company name

Description automatically generated

c) 将Signature和发送内容一起发送到客户端



d) 客户端获取数字签名，用公钥解密，得到信件的摘要。

A picture containing text

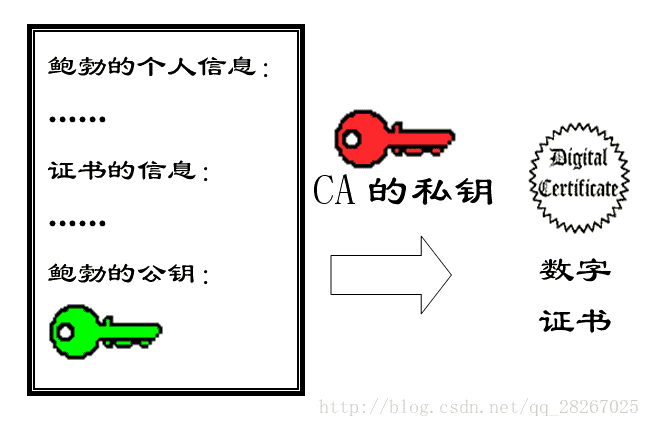
Description automatically generated

e) 客户端对信件本身也进行一次Hash，将得到的Diges与服务端的摘要进行对比。如果两者一致，就证明这封信未被修改过

Diagram

Description automatically generated

f) 为了防止公钥被第三方劫持，需要使用CA认证的数字证书。



Text, letter

Description automatically generated

Diagram

Description automatically generated

#### 3.3 身份识别(数字证书)

在信息传输的过程中，我们通常需要验证信息的发送方的身份，这时我们转化一下思路就可以完成，把发送端的公钥发送给接收端，发送端通过把自己的内容使用私钥加密然后发送给接收端，接收端只能用发送端的公钥解密，自然就验证了发送端的身份。

A picture containing diagram

Description automatically generated

#### 3.4 数字证书

在传输的过程中，客户端如何获得服务器端的公钥呢？当时是服务器分发给客户端，如果一开始服务端发送的公钥到客户端的过程中有可能被第三方劫持，然后第三方自己伪造一对密钥，将公钥发送给客户端，当服务器发送数据给客户端的时候，中间人将信息进行劫持，用一开始劫持的公钥进行解密后，然后使用自己的私钥将数据加密发送给客户端，而客户端收到后使用公钥解密，反过来亦是如此，整个过程中间人是透明的，但信息泄露却不得而知。

Diagram

Description automatically generated

为了防止这种情况，数字证书就出现了，它其实就是基于上上面所说的私钥加密数据，公钥解密来验证其身份。  
数字证书是由权威的CA（Certificate Authority）机构给服务端进行颁发，CA机构通过服务端提供的相关信息生成证书，证书内容包含了持有人的相关信息，服务器的公钥，签署者签名信息（数字签名）等，最重要的是公钥在数字证书中。  
数字证书是如何保证公钥来自请求的服务器呢？数字证书上由持有人的相关信息，通过这点可以确定其不是一个中间人；但是证书也是可以伪造的，如何保证证书为真呢？  
一个证书中含有三个部分:"证书内容，散列算法，加密密文"，证书内容会被散列算法hash计算出hash值，然后使用CA机构提供的私钥进行RSA加密。

Diagram
t
Description automatically generated

当客户端发起请求时，服务器将该数字证书发送给客户端，客户端通过CA机构提供的公钥对加密密文进行解密获得散列值（数字签名），同时将证书内容使用相同的散列算法进行Hash得到另一个散列值，比对两个散列值，如果两者相等则说明证书没问题。

Diagram

Description automatically generated

#### 3.5 常见证书文件类型

X.509#DER 二进制格式证书，常用后缀.cer .crt  
X.509#PEM 文本格式证书，常用后缀.pem  
有的证书内容是只包含公钥（服务器的公钥），如.crt、.cer、.pem  
有的证书既包含公钥又包含私钥（服务器的私钥），如.pfx、.p12

### 四、HTTPS、TLS/SSL

Hyper Text Transfer Protocol over Secure Socket Layer，安全的超文本传输协议，网景公式设计了SSL(Secure Sockets Layer)协议用于对Http协议传输的数据进行加密，保证会话过程中的安全性。

使用TCP端口默认为443

TLS: (Transport Layer Security，传输层安全协议)，用于两个应用程序之间提供保密性和数据完整性。

SSL: (Secure Socket Layer，安全套接字层)，位于可靠的面向连接的网络层协议和应用层协议之间的一种协议层。SSL通过互相认证、使用数字签名确保完整性、使用加密确保私密性，以实现客户端和服务器之间的安全通讯。

SSL协议即用到了对称加密也用到了非对称加密(公钥加密)，在建立传输链路时，SSL首先对对称加密的密钥使用公钥进行非对称加密，链路建立好之后，SSL对传输内容使用对称加密。

对称加密  
速度高，可加密内容较大，用来加密会话过程中的消息

公钥加密  
加密速度较慢，但能提供更好的身份认证技术，用来加密对称加密的密钥

Diagram

Description automatically generated

Diagram

Description automatically generated

#### 4.1 HTTPS单向认证

Graphical user interface, application

Description automatically generated

1. 客户端向服务端发送客户端支持的**SSL协议版本号、加密套件候选列表**、**压缩算法候选列表、扩展字段、**客户端**随机数A**等;

* 支持的最高 TLS 协议版本 version，从低到高依次 SSLv2, SSLv3, TLSv1, TLSv1.1, TLSv1.2, 当前基本不再使用低于 TLSv1 的版本
* 客户端支持的加密套件 cipher suites 列表， 每个加密套件对应前面 TLS 原理中的四个功能的组合：

1. 认证算法 Au (身份验证)
2. 密钥交换算法 KeyExchange(密钥协商)
3. 对称加密算法 Enc (信息加密)
4. 信息摘要 Mac(完整性校验)

* 支持的压缩算法compression methods 列表，用于后续的信息压缩传输
* **随机数A**用于后续的密钥的生成

1. 服务端返回协商结果：双方都支持的**SSL协议版本号、对称加密算法**，服务端**随机数B**等，同时返回服务器端的证书链，即公钥证书链;
2. 客户端使用服务端返回的信息验证服务器的合法性，包括：
   * 证书是否过期;
   * 发行服务器证书的CA是否可信;(通过查询浏览器或本机内的CA证书)
   * 返回的公钥是否能正确解开返回证书中的数字签名;（通过使用本机或浏览器内置的CA公钥进行解密）
   * 服务器证书上的域名是否和服务器的实际域名相匹配;
   * 验证通过后，将继续进行通信，否则，终止通信;

验证通过后，客户端回应服务器端，客户端生成一个“**随机数C**”(也被称为“**Pre-Master**”)，然后使用服务器端证书中的公钥加密“随机数C”，将加密后的“随机数C”发送给服务器端；

1. 服务器端使用密钥库中的私钥解密加密后的“**Pre-Master**”得到“随机数C”，此时客户端和服务器端都拿到了随机数A、随机数B、随机数C，双发通过这3个随机数使用相同的**密钥交换算法**计算得到相同的对称加密密钥，这个对称加密密钥就作为客户端和服务器端数据传输时对称加密的密钥;  
   在接下来的会话中，服务器和客户端将会使用该密码进行对称加密，保证通信过程中信息的安全;

#### 4.2 HTTPS双向认证

Graphical user interface, text, application, email

Description automatically generated

1. 客户端向服务端发送客户端支持的**SSL协议版本号、加密套件候选列表**、**压缩算法候选列表、扩展字段、**客户端**随机数A**等;

* 支持的最高 TLS 协议版本 version，从低到高依次 SSLv2, SSLv3, TLSv1, TLSv1.1, TLSv1.2, 当前基本不再使用低于 TLSv1 的版本
* 客户端支持的加密套件 cipher suites 列表， 每个加密套件对应前面 TLS 原理中的四个功能的组合：

1. 认证算法 Au (身份验证)
2. 密钥交换算法 KeyExchange(密钥协商)
3. 对称加密算法 Enc (信息加密)
4. 信息摘要 Mac(完整性校验)

* 支持的压缩算法compression methods 列表，用于后续的信息压缩传输
* **随机数A**用于后续的密钥的生成

1. 服务端返回协商结果：双方都支持的**SSL协议版本号、对称加密算法**，服务端**随机数B**、**服务器端的证书链**(即公钥证书链)、要求客户端发送**客户端的证书**(即客户端证书公钥);
2. 客户端使用服务端返回的信息验证服务器的合法性，包括：
   * 证书是否过期;
   * 发行服务器证书的CA是否可信;(通过查询浏览器或本机内的CA证书)
   * 返回的公钥是否能正确解开返回证书中的数字签名;（通过使用本机或浏览器内置的CA公钥进行解密）
   * 服务器证书上的域名是否和服务器的实际域名相匹配;
   * 验证通过后，将继续进行通信，否则，终止通信;

验证通过后，客户端回应服务器端:

a) 客户端证书

b) 客户端生成一个“**随机数C**”(也被称为“**Pre-Master**”)，然后使用服务器端证书中的公钥加密“随机数C”，将加密后的“随机数C”发送给服务器端。

c) 验证消息（Certificate Verify message）：客户端将之前所有收到的和发送的消息组合起来，并用hash算法得到一个hash值，然后用**客户端密钥库的私钥**对这个hash进行签名，这个签名就是Certificate Verify message。

服务器端收到客户端证书后，会做如下处理：

a) 确认客户端发送的证书是有效合法的。

b) 用客户端证书中的公钥验证收到信息中的签名，以确定这个证书是客户端发出的。

1. 服务器端使用密钥库中的私钥解密加密后的“Pre-Master”得到“随机数C”，此时客户端和服务器端都拿到了随机数A、随机数B、随机数C，双发通过这3个随机数使用相同的密钥交换算法计算得到相同的对称加密密钥，这个对称加密密钥就作为客户端和服务器端数据传输时对称加密的密钥;  
   在接下来的会话中，服务器和客户端将会使用该密码进行对称加密，保证通信过程中信息的安全;

### 五、密钥交换算法

密钥交换算法即DH算法：Diffie-Hellman算法。DH算法解决了密钥在双方不直接传递密钥的情况下完成密钥交换，这个神奇的交换原理完全由数学理论支持。

我们来看DH算法交换密钥的步骤。假设甲乙双方需要传递密钥，他们之间可以这么做：

1. 甲首选选择一个素数p，例如509，底数g，任选，例如5，随机数a，例如123，然后计算A=g^a mod p，结果是215，然后，甲发送p＝509，g=5，A=215给乙；
2. 乙方收到后，也选择一个随机数b，例如，456，然后计算B=g^b mod p，结果是181，乙再同时计算s=A^b mod p，结果是121；
3. 乙把计算的B=181发给甲，甲计算s＝B^a mod p的余数，计算结果与乙算出的结果一样，都是121。

所以最终双方协商出的密钥s是121。注意到这个密钥s并没有在网络上传输。而通过网络传输的p，g，A和B是无法推算出s的，因为实际算法选择的素数是非常大的。

所以，更确切地说，DH算法是一个密钥协商算法，双方最终协商出一个共同的密钥，而这个密钥不会通过网络传输。

如果我们把a看成甲的私钥，A看成甲的公钥，b看成乙的私钥，B看成乙的公钥，DH算法的本质就是双方各自生成自己的私钥和公钥，私钥仅对自己可见，然后交换公钥，并根据自己的私钥和对方的公钥，生成最终的密钥secretKey，DH算法通过数学定律保证了双方各自计算出的secretKey是相同的。

质数又称素数。一个大于1的自然数，除了1和它自身外，不能被其他自然数整除的数叫做质数。

MOD取模[运算符](https://baike.baidu.com/item/%E8%BF%90%E7%AE%97%E7%AC%A6/7752795)，[算法](https://baike.baidu.com/item/%E7%AE%97%E6%B3%95/209025)和取余运算(REM)相似。例如a mod b=c，表明a除以b余数为c。