数字签名是什么？

如果发送者A和接收者B使用不同的密钥，例如A发送消息的时候使用私钥对消息进行加密，B接收消息的时候使用公钥对消息进行解密。因为消息只能由A的私钥进行加密，所以这个签名一定是由A签发的，这样就没有否认的问题了。这个就是数字签名(digital signature)。

签名的生成和验证

生成消息签名这一行为是由发送者A来完成的，也称为对消息进行签名。生成签名就是根据消息内容计算签名值，生成签名意味着A认可这个消息的内容。

验证数字签名可以由消息接受者B来完成，也可以由第三方来完成。验证成功意味着这个消息是由A发出，失败则表示这个消息不是A发出的。

在数字签名的过程中，消息发送者A和消息接受者B使用不同的密钥来进行签名和验证。这里使用的不同的密钥就是公钥和私钥。

数字签名其实就是公钥密码的反向应用，下面我们看看两种的不同：



数字签名的方法

通常来说数字签名一般有两种方式：

直接对消息进行签名

对消息的hash值进行签名

直接对消息进行签名

直接对消息签名包含如下几个步骤：

发送者A用自己的私钥对消息进行加密生成签名。

A将加密后的签名和消息发送给B。

B用A的公钥对消息签名进行解密，从而得到签名之前的消息M1。

B将M1和A直接发送过来的消息M2进行对比，两者一致则签名成功，否则失败。

这里我们注意一下第四个步骤，签名的目的是保证消息是由只持有该密钥的人生成的，而并不是要保证消息传递的机密性。也就是说数字签名本身并不是用来保证机密性的。如果要保证机密性则可以将消息加密之后再发出去。

对消息的hash值进行签名

上面的直接对消息进行签名，看起来非常简单，但是在实际应用中很少用到。因为要对整个消息进行签名时一个非常耗时的操作，所以通常我们会使用单向散列函数对消息进行处理得出一个hash值，然后对这个hash值进行签名：

A用单向散列函数对消息进行计算hash值。

A用自己的私钥对hash值进行签名。

A将消息和签名发给B

B用A的公钥对签名进行解密得到hash值。

B使用单向散列函数对消息进行计算，将结果和4进行比对。

数字签名无法解决的问题

使用数字签名我们可以防止伪造和篡改，也无法防止否认。但是还需要一个大前提就是验证签名的公钥必须是真正属于发送者。

这里我们就需要一个可信任的机构来为我们颁发可信任的公钥，这里就涉及到证书和PKI的知识了。