第五章 系统安全分析与设计

1.信息系统安全属性

安全属性：

保密性：最小授权原则(只给应用维持其基本职能最基本的权限)、防暴露(如在命名时可以将名称设置为乱码)、信息加密(防止信息在传输时被截获后破解)、物理保密

完整性(即保证数据在传输过程中不受损耗)：安全协议、校验码、密码校验、数字签名、公证

可用性(只允许合法用户使用这些资源)：综合保障(IP过滤、业务流控制、路由控制选择、审计跟踪)

不可抵赖性：数字签名

第二节.对称加密技术与非对称加密技术

常见的非对称密钥加密算法　又称公开密钥加密：

1.RSA 512位

2.ELgamal 3.ecc 4 背包算法 Rabin d-h

常见的对称加密技术（又称共享密钥加密）：

1.DES 56位

2.3DES 112 位 三次加密 两个密钥　3DES更难破解；优点：加密速度快、效率高。缺点：加密强度不高、密钥分发困难

3.AES 256位

4.RC-5 5.IDEA 128位

理解：

每个人都有自己的公钥，而公钥是明文公开的，大家都知道的，而私钥则是发送方与接受方各自独有且不能被别人知道的，如甲的公钥加密的信息只能由甲的私钥来解密，甲想要把信息发送给乙则可以用乙的公钥加密后再发送给乙，乙收到后再用自己的私钥来解密；该加密方式一般采用1024位加密，因此极为庞大的信息若采用该加密方式则极难解密。

前提：公钥公开透明，私钥只有各自知道。从甲方 到 乙方 甲的公钥加密的data只能有甲用私钥解密 甲想把信息发给乙 只能用乙的公钥加密后再发给乙 乙收到后再用自己的私钥来解密

第三节.信息摘要与数字签名

信息摘要(防止信息被篡改)

1.概念：在信息中，摘要是信息的特征值，原始信息发生变化则特征值会改变。

2.应用：在信息传输中，若A与B进行信息传输，内容被丙截获，进而篡改后再发送给乙，则会造成信息错误，因此，若A在将信息传输给乙时可以同时传输一个摘要，B收到信息后将摘要翻译出并与之比较，则能够判断信息是否被篡改。

3.算法：信息摘要采用的算法是单向散列函数(单向Hash函数),即将明文转换为摘要。而摘要不能转换为明文，常用的信息摘要算法还有MD5、SHA等，市场上广泛使用的MD5，SHA算法的散列值分别为128和160位，由于SHA通常采用的密钥长度较长，因此安全性高于MD5。

解析加密、数字签名和数字证书

https://blog.csdn.net/TheSkyLee/article/details/108699243?ops_request_misc=%257B%2522request%255Fid%2522%253A%252216634023931680018274411 2%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=166340239316800182744112&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-108699243-null-null.142^v47control,201^v3control_1&utm_term=%E6%95%B0%E5%AD%97%E7%AD%BE%E5%90%8D&spm=1018.2226.3001.4187

什么是加密：

加密就是对明文数据按某种特殊算法进行处理，使其成为不可读的一段代码，通常称为“密文“， 密文通过”密钥“解密后还原出原来的明文，通过这样的途径可以达到保护数据不被非法人窃取、阅读的目的。

私钥在非对称技术中用来解密，在数字签名中则用来签名，B在收到A的信息后用A的公钥解密即验证，则信息发自A是不可抵赖的

举例说明：

甲 and 乙 窃密者：eve

第一回合：

甲发了一封邮件给乙　我们知道消息在网络都是明文传输的　所以eve可以很简单的篡改获取甚至冒充甲

eve截获可以修改明文再发给乙　所以对邮件进行窃听 对内容进行了篡改 还对甲进行了伪装

就会!! 毫无安全感!

第二回合：

如果不能进行明文传输 甲和乙就商量好密钥 对文件进行加密就好了 然后由乙解密就行了

甲和乙提前商量好的密钥加密后进行传输

由于eve 没有密钥 就算截取了也无法获取里面的信息

因为篡改后的数据必须使用密钥再次加密乙才能正确解密

所以只要保证密钥不泄露就可以了

如果密钥被泄露 就相当于明文通信了

存在一个问题：对方怎样安全的交换密钥呢？ 如果交换时必须明文通信 则一切泡汤

第三回合：　甲—bob 乙-alice

非对称加密！！ 甲和乙生成一对公私钥

首先 Alice 需要先生成一对公私钥，私钥只能 Alice 自己知道，公钥是可以让任何人都知道的，因此可将公钥直接发送给 Bob，就算被截获也无所谓。

Bob 使用 **Alice** 的公钥加密邮件内容，加密后的内容只能由 **Alice** 的私钥解密，所以就算 Eve 截获也是徒劳。

反之，如果 Alice 想给 Bob 回信，就需要用 **Bob** 的公钥加密后发送。

这就解决了密钥交换问题，也保证了邮件内容不会泄露。也就是说现在可以防窃听。

第四回合：如何证明甲是本人甲？

不知道你注意到没有，这里也存在另外一个问题：

Eve 也可以使用 **Alice** 的公钥冒充 **Bob** 给 **Alice** 发邮件啊，因为 **Alice** 的公钥本来就是公开的，任何人都可以获得。

由于 Eve 也可以获得 Alice 公钥，所以没法防止 Eve 伪造和篡改，并且对于 Alice 而言，她无法分辨出邮件到底是 Eve 发的还是 Bob。

所以这个问题的本质就是 「**Alice** 如何确认邮件来自于 **Bob**」。

那么在生活中，我们如何做这件事呢？

那就是让 Bob 在纸上签名并且按手印，因为指纹和字迹是 Bob 独有的，其它人很难伪造。

所以我们需要在计算机中引入类似的机制：

即只有 **Bob** 自己能够产生的独一无二的标志，并且其它人能够验证这个标志确实是属于 **Bob**的。

这就是我们今天要讲的主题—「数字签名」。

还记得什么是 Bob 独有的吗？

对，就是 Bob 自己的私钥，Bob 用自己的私钥对邮件内容计算一个「签名」，将「签名」和邮件内容一起发送出去，接受者 Alice 可以使用 **Bob** 的公钥验证这个签名是否正确，这就叫「验签」。

如果不是 Bob 的私钥计算的签名，那么 Alice 用 Bob 公钥验签将会出错。

可以看到， Eve 试图使用自己的私钥计算签名然后发送给 Alice， 但是 Alice 使用 Bob的公钥进行验签时将会出错！

那么 Eve 可能篡改内容并冒充 Bob 的签名吗？不可能！因为内容发生改变时，对应的签名也需要重新计算，而签名的生成依赖于私钥，只要 **Bob** 的私钥不泄露，签名就不会被冒充。

所以使用数字签名，我们能够鉴别消息的发送者，也就是说黑客无法伪装发送者进行发送数据，也无法篡改。

接受者 Alice 收到后，取下数字签名，同时用 Bob 的公钥解密，得到「摘要1」，证明确实是 **Bob** 发的。

(画外音：如果使用 *Bob* 的公钥验证签名出错，那么签名一定不是 *Bob* 的私钥生成的）

再对邮件内容使用相同的散列函数计算「摘要2」，与上面得到的「摘要1」进行对比，两者一致就说明信息未被篡改。

这样两步分证明发送者身份和保证数据未被篡改。

第四回合：

数字证书　甲的公钥被利用或者篡改 无法验证甲的公钥就是甲的公钥

之前是消息进行数字签名 然后接受者进行验证 现在是公钥是否是甲发的？

为了解决这个问题，就引入了「数字证书」，什么叫数字证书呢？

看了这个描述，是不是感觉还是云里雾里，还是我用大白话来说吧~

只要你理解了前面的数字签名，就能理解这里的数字证书，因为我把数字证书叫做「公钥的数字签名」。

为什么呢？我们引入数字证书的目的是为了保证公钥不被篡改，即使被篡改了也能识别出来。

而防篡改的方法就是数字签名，但是这个签名不能我们自己做，原因说过了，因为我们的公钥还没分发出去，别人无法验证。

所以只能找可信的第三方来帮我们签名，即证书颁布机构（CA），CA 会将：证书的颁布机构、有效期、公钥、持有者(subject)等信息用 CA 的私钥进行签名。

并且将签名结果和这些信息放在一起，这就叫做「数字证书」。

这样，Bob 就可以去 CA 申请一个证书，然后将自己的证书发给 Alice，那么 Alice 如何验证这个证书确实是 Bob的呢？

当然是使用 CA 的公钥进行验签。

收到 Bob 发过来的数字证书后，Alice 使用 CA 的公钥进行验证，验证通过即证明这确实是 Bob 证书，也就可以使用证书中包含的 Bob 的公钥，按照之前讨论的流程进行通信。

那么 Eve 是否可以在中途篡改 Bob 的证书呢？

答案是不行，因为证书的信息使用 CA 的私钥进行签名，只要 Eve 修改了任何一个 Bit 都会导致最后签名验证不通过。

那 Eve 可不可以修改证书信息后自己重新计算一次证书的数字签名呢？

也不行，因为证书的数字签名计算依赖于 CA 的私钥，Eve 是拿不到 CA 的私钥的。

如果拿到了，说明什么？整个世界都是不可信的。

现在我们来回答文章开头提出的一些问题：

1．非对称加密中公私钥都可以加密，那么什么时候用公钥加密，什么时候用私钥“加密”？

- 加密场景，那么肯定希望只有我才能解密，别人只能加密。即公钥加密，私钥解密。
- 签名场景，既然是签名，就希望只能我才能签名，别人只能验证。即私钥签名，公钥验签

2．什么是数字签名，数字签名的作用是什么？

- 数字签名就是使用私钥对数据摘要进行签名，并附带和数据一起发送。
- 可以起到防篡改、防伪装、防否认的作用。

3．为什么要对数据的摘要进行签名，而不是直接计算原始数据的数字签名？

- 数据可能比较大，签名是使用非对称加密算法，比较耗时
- 防止第三方使用公钥解开签名后，拿到原始数据

4．什么是数字证书，数字证书存在解决了什么问题？

- 数字证书就是由 CA 机构使用自己私钥，对证书申请者的公钥进行签名认证。
- 数字证书解决了如何安全分发公钥的问题，也奠定了信任链的基础。

名词：

非对称加密技术

对称加密技术

信息摘要

数据签名

数字验证