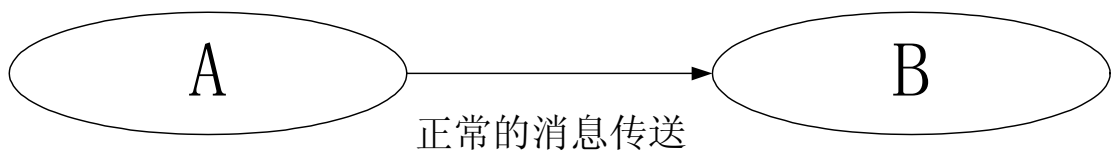


公钥密码技术讲义

1.问题的引入

1.1 攻击类型

根据攻击的不同方式，攻击被分为被动攻击和主动攻击。



图表 1 消息的正常传送

被动攻击

获得正在传送的信息。其特点是：偷听或监视传送。攻击的手段是：泄露消息内容和通信量分析。（绘图说明）

主动攻击

主动攻击主要涉及到数据流的修改或创建错误流。攻击手段是：伪装、重放、修改消息和拒绝服务。

1.2 安全服务

(简要说明)

- A. 保密性
- B. 验证（鉴别）
- C. 完整性
- D. 不可抵赖性（不可否认性）
- E. 访问控制
- F. 可用性

1.3 常规加密的缺陷

尽管对称密码技术有一些很好的特性，但它也存在着明显的缺陷，主要在于其密钥的管理：

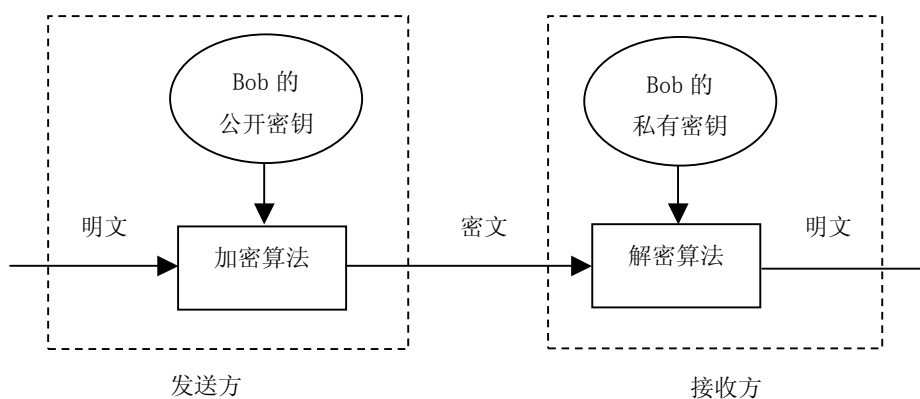
- A. **进行安全通信前需要以安全方式进行密钥交换。**这一步骤，在某种情况下是可行的，但在某些情况下会非常困难，甚至无法实现。
- B. **密钥规模复杂。**举例来说，A 与 B 两人之间的密钥必须不同于 A 和 C 两人之间的密钥，否则给 B 的消息的安全性就会受到威胁。在有 1000 个用户的团体中，A 需要保持至少 999 个密钥（更确切的说是 1000 个，如果她需要留一个密钥给他自己加密数据）。对于该团体中的其它用户，此种情况同样存在。这样，这个团体一共需要将近 50 万个不同的密钥！推而广之，n 个用户的团体需要 $n^2/2$ 个不同的密钥。

2. 公钥密码技术

2.1 基本概念

应用两个不同的密钥：一个是公开的，一个是秘密的。从公开密钥(以下简称为公钥)很难推断出私人密钥(以下简称为私钥)。持有公钥的任何人都可以加密消息，但却无法解密。只有持有私钥的人才能够解密。

2.2 加密/解密基本步骤



图表 2 加密/解密基本步骤

一般的情况下，网络中的用户约定一个共同的公开密钥密码系统，每个用户都有自己的公钥和私钥，并且所有的公钥都保存在某个公开的数据库中，任何用户都可以访问此数据库。这样加密协议如下：

- A. Alice 从公开数据库中取出 Bob 的公开密钥。
- B. Alice 用 Bob 的公开密钥加密她的消息，然后传送给 Bob
- C. Bob 用他的私钥解密 Alice 的消息。

2.3 优点

从以上的介绍中可以看出，与对称密码技术相比较，利用非对称密码技术进行安全通信，有以下优点：

- A. 通信双方事先不需要通过保密信道交换密钥
- B. 密钥持有量大大减少。在 n 个用户的团体中进行通信，每一用户只需要持有自己的私钥，而公钥可放置在公共数据库上，供其它用户取用。这样，整个团体仅需拥有 n 对密钥，就可以满足相互之间进行安全通信的需求。（实际中，因安全方面的考虑，每一用户可能持有多个密钥，分别用于数字签名、加密等用途。此种情况下，整个团体拥有的密钥对数为 n 的倍数。但即使如此，与使用对称密码技术时需要 $n^2/2$ 个不同的密钥相比，需要管理的密钥数量仍显著减少。）
- C. 非对称密码技术还提供了对称密码技术无法或很难提供的服务：如与哈希函数联合运用可生成数字签名（下面介绍），可证明的安全伪随机数发生器的构造，零知识证明等。

2.4 公钥密码系统提供的安全服务

- A. 加密/解密：发送方可以用接收方的公钥加密消息。
- B. 数字签名：发送方用其私钥“签署”消息，通过对消息或作为消息函数的小块数据应用加密算法来进行签署。
- C. 密钥交换：两方互相合作可以进行会话密钥的交换。

2.5 理论基础

一个公开密钥密码系统必须满足的条件是：

- A. 通讯双方 A 和 B 容易通过计算产生出一对密钥（公开密钥 K_1 ，私钥密钥 K_2 ）
- B. 在知道公开密钥 K_1 和待加密报文 M 的情况下，对于发送方 A，很容易通过计算产生对应的密文：
- C. $C = Ek_1 M$
- D. 接收方 B 使用私有密钥容易通过计算解密所得的密文以便恢复原来的报文：
- E. $M = Dk_2 C = Dk_2[Ek_1 M]$
- F. 除 A 和 B 以外的其他人即使知道公钥 k_1 ，要确定私钥 K_2 在计算上也是不可行的。
- G. 除 A 和 B 以外的其他人即使知道公钥 k_1 和密文 C ，要想恢复原来的明文 C 在计算上也是不可行的。

这些要求最终可以归结到设计一个单向陷门函数。

单向函数：

一个单向函数是满足下列条件的函数：它将一个定义域映射到值域，使得每个函数值有一个唯一的原像，同时还要满足下列条件：函数值计算很容易，而逆计算是不可行的。

单向陷门函数：

所谓单向陷门函数是这样的函数，即除非知道某种附加的信息，否则这样的函数在一个方向上容易计算，而在另外的方向上要计算是不可行的。有了附加的信息，函数的逆就可以在多项式时间内计算出来。

一个实用的公开密钥密码系统的建立和发展依赖于找到一个单向陷门函数。

2.6 公开密钥密码分析

攻击公开密钥密码系统的方法有如下几种：

穷举法 - 对此防范措施应为：使用长的密钥。但是由于公钥算法依赖于单向陷门函数，计算函数的复杂性与密钥的长度的关系可能会增长得更快。因而密钥大小必须足够大，以保证安全性，但是又要足够小以便加密解密使用。

根据公钥计算私钥 - 在数学上证明，对任何公钥算法，这种分析可能成功。因而对任何公钥算法，对这种攻击方法都需要测试

3.常用加密算法

3.1 Diffie-Hellman 密钥交换

DH 算法是 W.Diffie 和 M.Hellman 提出的。此算法是最早的公钥算法。它实质是一个通信双方进行密钥协定的协议：两个实体中的任何一个使用自己的私钥和另一实体的公钥，得到一个对称密钥，这一对称密钥其它实体都计算不出来。DH 算法的安全性基于有限域上计算离散对数的困难性。离散对数的研究现状表明：所使用的 DH 密钥至少需要 1024 位，才能保证有足够的中、长期安全。

3.2 RSA

RSA 算法是 R.Rivest、A.Shamir 和 L.Adleman 于 1977 年在美国麻省理工学院开发，于 1978 年首次公布。

RSA 公钥密码算法是目前网络上进行保密通信和数字签名的最有效的安全算法之一。RSA 算法的安全性基于数论中大素数分解的困难性，所以，RSA 需采用足够大的整数。因子分解越困难，密码就越难以破译，加密强度就越高。

其算法如下：

- A. 选择两质数 p 、 q
- B. 计算 $n = p * q$
- C. 计算 n 的欧拉函数 $\Phi(n) = (p - 1)(q - 1)$
- D. 选择整数 e ，使 e 与 $\Phi(n)$ 互质，且 $1 < e < \Phi(n)$
- E. 计算 d ，使 $d * e = 1 \text{ mod } \Phi(n)$

其中，公钥 $KU = \{e, n\}$ ，私钥 $KR = \{d, n\}$

加密/解密过程:

利用 RSA 加密, 首先需将明文数字化, 取长度小于 $\log_2 n$ 位的数字作为明文块。

对于明文块 M 和密文块 C , 加/解密的形式如下:

$$\text{加密: } C = M^e \bmod n$$

$$\text{解密: } M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

RSA 的安全性基于大数分解质因子的困难性。因为若 n 被分解为 $n = p * q$, 则 $\Phi(n)$ 、 e 、 d 可依次求得。目前, 因式分解速度最快的方法的时间复杂性为 $\exp(\sqrt{\ln(n)\ln\ln(n)})$ 。统计数据表明, 在重要应用中, 使用 512 位的密钥已不安全, 需要采用 1024 位的密钥。

3.3 椭圆曲线密码体制(ECC)

原理:

1985 年, N. Koblitz 和 V. Miller 分别独立提出了椭圆曲线密码体制(ECC), 其依据就是定义在椭圆曲线点群上的离散对数问题的难解性。

为了用椭圆曲线构造密码系统, 首先需要找到一个单向陷门函数, 椭圆曲线上的数量乘就是这样的单向陷门函数。

椭圆曲线的数量乘是这样定义的: 设 E 为域 K 上的椭圆曲线, G 为 E 上的一点, 这个点被一个正整数 k 相乘的乘法定义为 k 个 G 相加, 因而有

$$kG = G + G + \dots + G \quad (\text{共有 } k \text{ 个 } G)$$

若存在椭圆曲线上的另一点 $N \neq G$, 满足方程 $kG = N$ 。容易看出, 给定 k 和 G , 计算 N 相对容易。而给定 N 和 G , 计算 $k = \log_G N$ 相对困难。这就是椭圆曲线离散对数问题。

离散对数求解是非常困难的。椭圆曲线离散对数问题比有限域上的离散对数问题更难求解。对于有理点数有大素数因子的椭圆离散对数问题, 目前还没有有效的攻击方法。

建立椭圆曲线密码体制

选取适当的有限域 F_q 和椭圆曲线 E , 在 $E(F_q)$ 中选一个周期很大的点, 如选了一个点 $P=(x_p, y_p)$, 它的周期为一个大的素数 n , 记 $[n](P)=n(\text{素数})$

注意: 在这个密码体制中, 具体的曲线及点 P 和它的 n 都是公开信息。密码体制的形式采用 ElGamal 体制, 是完全类比过来。

加密/解密协议

a) 密钥的生成

- A. Bob (使用者) 执行了下列计算:
- B. 在区间 $[1, n-1]$ 中随机选取一个整数 d
- C. 计算点 $Q := dP$ (d 个 P 相加)
- D. Bob 公开自己的公开密钥 -- $(E(F_q), P, n, Q)$
- E. Bob 的私钥为整数 d

Alice 要发送消息 m 给 Bob, Alice 执行:

- A. 查找 Bob 的公钥 $(E(F_q), P, n, Q)$
- B. 将 m 表示成一个域元素 $P_m \in F_q$
- C. 在区间 $[1, n-1]$ 内选取一个随机数 k
- D. 依据 Bob 的公钥计算点 $(x_1, y_1) := kP$ (k 个 P 相加)
- E. 计算点 $(x_2, y_2) := kQ$, 如果 $x_2 = 0$, 则回到第 C 步
- F. 生成密文: $C_m = \{kP, P_m + kQ\}$
- G. 传送加密数据 $\{kP, P_m + kQ\}$ 给 Bob

b) Bob 的解密过程

Bob 接收到 $\{kP, P_m + kQ\}$,

Bob 用这一对中的第一个点乘以 B 的秘密密钥, 再从第二个点中减去这个值:

$$P_m + kQ - d(kP) = P_m + k(dQ) - d(kP) = P_m$$

4. 数字信封和数字签名

公钥密码体制在实际应用中包含数字签名和数字信封两种方式。

数字信封(Digital Envelop)的功能类似于普通信封。普通信封在法律的约束下保证只有收信人才能阅读信的内容; 数字信封则采用密码技术保证了只有规定的接收人才能阅读信息的内容。

数字信封中采用了单钥密码体制和公钥密码体制。信息发送者首先利用随机产生的对称密码加密信息, 再利用接收方的公钥加密对称密码, 被公钥加密后的对称密码被称之为数字信封。在传递信息时, 信息接收方要解密信息时, 必须先用自己的私钥解密数字信封, 得到对称密码, 才能利用对称密码解密所得到的信息。这样就保证了数据传输不可抵赖性。

数字签名(Digital Signature)是指用户用自己的私钥对原始数据的哈希摘要进行加密所得的数据。信息接收者使用信息发送者的公钥对附在原始信息后的数字签名进行解密后获得哈希摘要, 并通过与自己用收到的原始数据产生的哈希摘要对照, 便可确信原始信息是否被篡改。这样就保证了消息来源的真实性和数据传输的完整性。

4.1 数字签名原理

在文件上手写签名长期以来被用作作者身份的证明, 或表明签名者同意文件的内容。实际上, 签名体现了以下几个方面的保证:

- A. **签名是可信的。** 签名使文件的接收者相信签名者是慎重地在文件上签名的。
- B. **签名是不可伪造的。** 签名证明是签字者而不是其他的人在文件上签字。
- C. **签名不可重用。** 签名是文件的一部分, 不可能将签名移动到不同的文件上。
- D. **签名后的文件是不可变的。** 在文件签名以后, 文件就不能改变。
- E. **签名是不可抵赖的。** 签名和文件是不可分离的, 签名者事后不能声称他没有签过这个文件。

而在计算机上进行数字签名并使这些保证能够继续有效则还存在一些问题。

首先, 计算机文件易于复制, 即使某人的签名难以伪造, 但是将有效的签名从一个文件剪辑和粘贴到另一个文件是很容易的。这就使这种签名失去了意义。

其次, 文件在签名后也易于修改, 并且不会留下任何修改的痕迹。

有几种公开密钥算法都能用作数字签名, 这些公开密钥算法的特点是不仅用公开密钥加

密的消息可以用私钥解密和，而且反过来用私人密钥加密的消息也可以用公开密钥解密。其基本协议很简单：

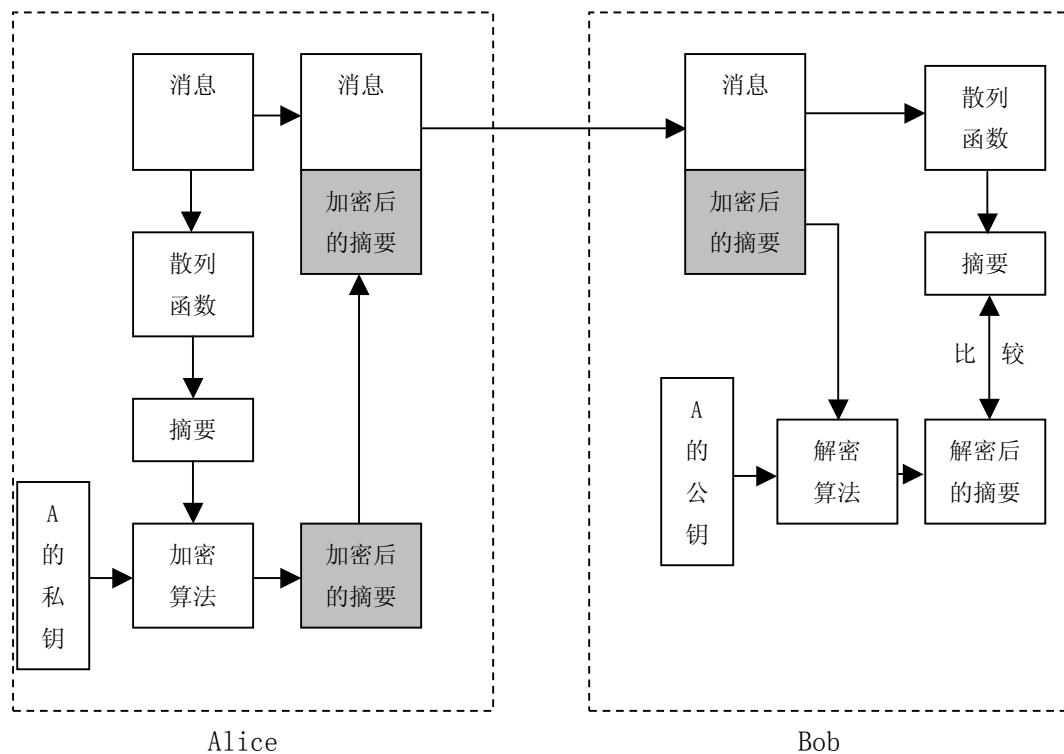
- A. Alice 用她的私钥对文件加密，从而对文件签名。
- B. Alice 将签名后的文件传给 Bob
- C. Bob 用 Alice 的公钥解密文件，从而验证签名。

在实际过程中，这种做法的准备效率太低了。从节省时间，数字签名协议常常与单向散列函数一起使用。Alice 并不对整个文件签名，而是只对文件的散列值签名。

在下面的协议中，单向散列函数和数字签名算法是事先协商好的：

- A. Alice 产生文件的单向散列值。
- B. Alice 用她的私人密钥对散列加密，以此表示对文件的签名。
- C. Alice 将文件和散列签名送给 Bob
- D. Bob 用 Alice 发送的文件产生文件的单向散列值，同时用 Alice 的公钥对签名的散列解密。如果签名的散列值与自己产生的散列值匹配，签名是有效的。

如下图：



图表 3 数字签名协议原理

由于两个不同的文件具有系统的 160 位散列值的概率为 $1/2^{160}$ ，所以在这个协议中使用散列函数的签名与使用文件的签名是一样安全的。

4.2 数字签名的应用例子

现在 Alice 向 Bob 传送数字信息，为了保证信息传送的保密性、真实性、完整性和不可否认性，需要对要传送的信息进行数字加密和数字签名，其传送过程如下：

- A. Alice 准备好要传送的数字信息（明文）。

- B. Alice 对数字信息进行哈希 (hash) 运算, 得到一个信息摘要。
- C. Alice 用自己的私钥 (SK) 对信息摘要进行加密得到 Alice 的数字签名, 并将其附在数字信息上。
- D. Alice 随机产生一个加密密钥 (DES 密钥), 并用此密钥对要发送的信息进行加密, 形成密文。
- E. Alice 用 Bob 的公钥 (PK) 对刚才随机产生的加密密钥进行加密, 将加密后的 DES 密钥连同密文一起传送给 Bob
- F. Bob 收到 Alice 传送过来的密文和加过密的 DES 密钥, 先用自己的私钥 (SK) 对加密的 DES 密钥进行解密, 得到 DES 密钥。
- G. Bob 然后用 DES 密钥对收到的密文进行解密, 得到明文的数字信息, 然后将 DES 密钥抛弃 (即 DES 密钥作废)。
- H. Bob 用 Alice 的公钥 (PK) 对 Alice 的数字签名进行解密, 得到信息摘要。
- I. Bob 用相同的 hash 算法对收到的明文再进行一次 hash 运算, 得到一个新的信息摘要。
- J. Bob 将收到的信息摘要和新产生的信息摘要进行比较, 如果一致, 说明收到的信息没有被修改过。

5. 常规加密技术和公钥加密技术的比较

常规加密技术的优点: 加密速度快、运行时占用资源少等。

公钥加密技术的优点: 密钥交换。

对两种技术的综合应用。

6. 参考文献

1. 《AES 和椭圆曲线密码算法的研究》顾婷婷 硕士毕业论文
2. 《密码编码学与网络安全: 原理与实践 (第二版)》William Stallings, 电子工业出版社
3. 《公钥密码技术讲义》张巍 四川大学计算机网络与安全研究所讲义