

密码学中的 CATCH-22 问题

在保密通信中，通信双方在进行秘密通信之前，他们首先得**共享密钥**，即需要首先进行一次分发密钥的秘密通信，但“分发密钥的秘密通信”是否是安全的不能得到证明

似乎进行“秘密的分发密钥”是不可避免的，两千年来，它一直被认为是密码学的一个定理。

对称密码体制的缺陷：

- **密钥管理困难** (n 个用户互相通信，系统中共有 $n(n-1)/2$ 个密钥。如 $n=100$ 时，共 4,995 个密钥。密钥爆炸。)
- **无法实现非否认** (因为解密者也可以加密)
- **密钥分发困难** (与“CATCH-22 问题”有关(如何安全地分发密钥)密钥分发不仅要耗费巨大的成本，而且也很容易成为保密通信中的薄弱环节！)

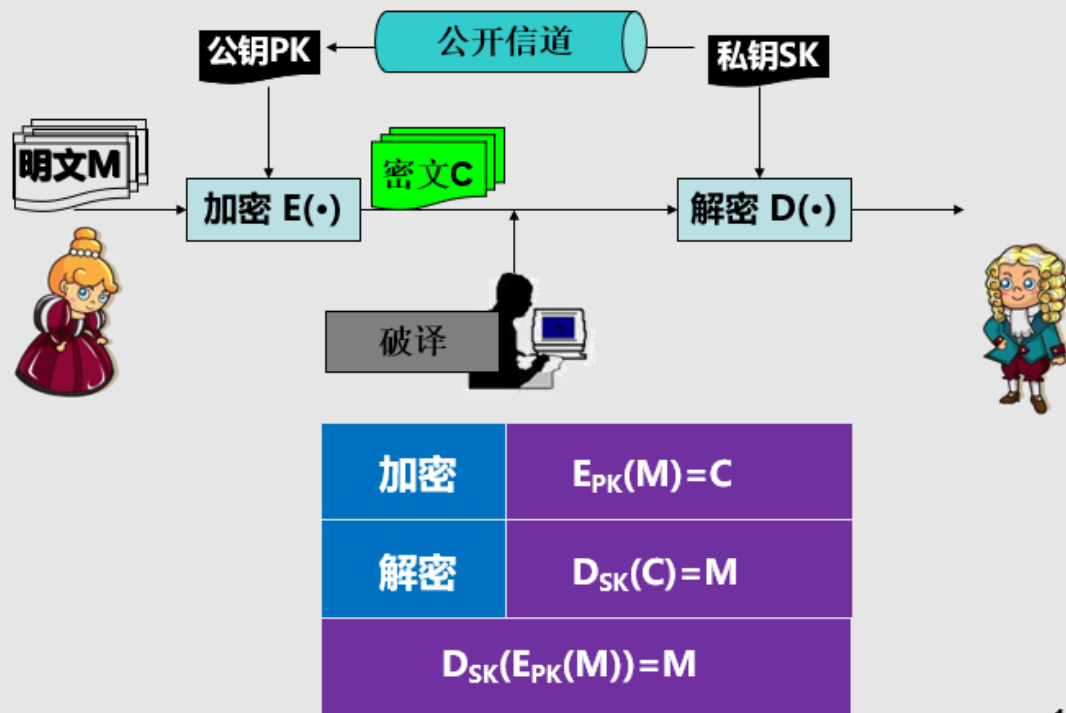
分发管理困难，无法实现非否认

公钥密码学

- **基本思想：**
 - Alice 有一个 **加密密钥 PK**，以及一个**解密密钥 SK**
 - PK 公开，SK 保密 (要求 PK 的公开不能影响 SK 的安全)
 - Bob 要向 Alice 发送明文 M 时，可先查 Alice 的加密密钥 PK，并用 PK 加密得密文 C
 - Alice 收到 C 后，用只有她自己才掌握的解密密钥 SK 解密 C 得到 M。
 - A 有加密密钥和解密密钥，B 通过加密密钥得到密文，A 收到密文后只能用自己掌握的解密密钥来解密密文
- **特点：**
 - 加密与解密由不同的密钥完成
 - 公钥：用于加密，任何人都可以知道(包括攻击者)
 - 私钥：用于解密，必须保密
 - 加解密的非对称性，故而又称 **非对称密码**

公钥密码学：加密密钥公开，解密密钥私有。故又称非对称密钥

公钥密码学下的秘密通信模型



13

Q: 从未见过面的两个人如何实现秘密通信？

- 公钥密码的提出，首先是解决 从未见过面的两个人如何实现秘密通信的问题。实质是，如何在公开信道上(肯定不安全)，实现对称密钥的秘密分发。
- Diffie 和 Hellman 提出的“Diffie-Hellman 密钥交换协议”正是为解决这一问题
- 他俩设想公钥加密的可能性，但没能提出解决方案

单向函数的概念是公钥密码学的中心概念

单向函数的特点

特点 计算容易，求逆计算上不可行



已知 x ，计算 $f(x)$ 很容易

已知 $f(x)$ ，计算 x 很难

- 现实世界中的例子——烧掉纸
 - 把纸烧掉很容易
 - 从纸灰恢复出原来的纸，却非常难

单向函数计算容易求解难

Q: 单向函数在密码学中很有用，但它们是否存在呢？

没有人证明单向函数是否真的存在，也没有实际的证据能够构造出真正的单向函数

但是，有很多函数看起来像单向函数，我们能有效地计算它们，且至今还不知道有什么办法能容易地求出它们的逆

有很多函数像单向函数但是也没有实际的证据能够构造出真正的单向函数


单向函数的作用

作用：单向函数不能用于加密，因为用单向函数加密的明文，没人能解开

陷门单向函数

特点：计算很容易，求逆计算上不可行。但知道秘密陷门，求逆很容易

陷门单向函数



特点	计算很容易，求逆计算上不可行。但知道秘密陷门，求逆很容易
<p>陷门单向函数满足下列条件：</p> <ul style="list-style-type: none">①给定x，计算$y=f(x)$很容易②给定y，计算x使$y=f(x)$是计算上不可行的③给定e，对给定的任何y，若相应的x存在，则计算x使$y=f(x)$是容易的	
<p>仅满足①、②两条的函数是单向函数</p> <p>第②条性质暗示仅由y推测x是计算上不可行的</p> <p>第③条称为陷门性，e称为陷门信息</p>	

29

注意 不是所有的困难问题都能转化成密码体制

- 一个困难问题可以转化为密码体制必须满足以下条件：

能将陷门信息 成功且安全 地嵌入到该问题中，使得只有用该陷门信息才可以有效求解

困难问题转化为密码体制必须满足能将陷门信息成功且安全的嵌入到该问题中，使得只有用该陷门信息才可以有效求解。

候选的单向函数：

- 大整数分解问题（已知 n 是两个大 p 和 q 的乘积对 n 进行分解求出 p 和 q ）
- 离散对数问题
- Diffie-Hellman 问题（
- 给定 (p, g, g^x, g^y) , p 是素数, $g \in \mathbb{Z}_p^*$ 是生成元
- 计算 Diffie-Hellman 问题 (CDH)
- 计算 g^{xy}
- 判定 Diffie-Hellman 问题 (DDH)
- 给定 $T \in \mathbb{Z}_p^*$, 判断 T 是否等于 g^{xy})

它们是否真是单向函数, 并未得到证明。但到目前为止, 还没找到多项式时间算法能解决这些问题

他们是困难的只是一个假设（困难假设）

解决从未见过面的两个人如何实现秘密通信

- 如何在公开信道上(肯定不安全), 实现对称密钥的秘密分发。
 - Diffie-Hellman 密钥交换协议正是为解决这一问题而提出的
- 基于的困难问题被称为 Diffie-Hellman 问题

Diffie-Hellman 密钥交换协议

- 系统建立:
 - 随机选择素数 p
 - 随机选择生成元 $g \in \mathbb{Z}_p^*$

$Y_A = g^x$

$Y_B = g^y$

$(Y_B^x) \bmod p = K$

$(g^{xy}) \bmod p = K$

$(Y_A^y) \bmod p = K$

$(g^{xy}) \bmod p = K$

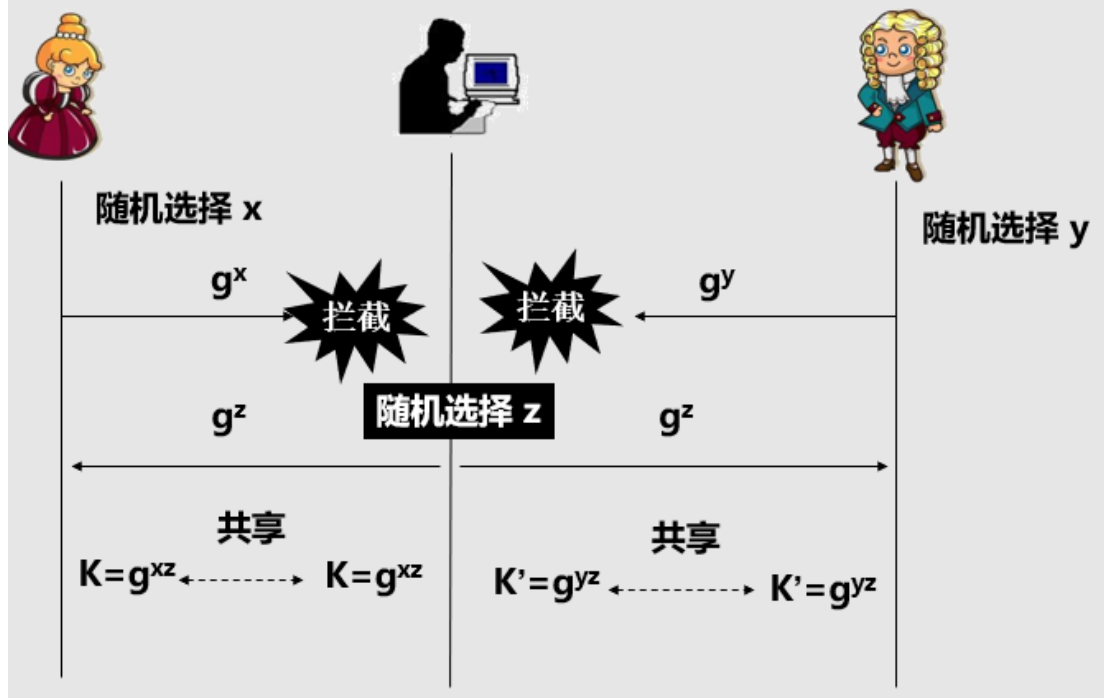
- 注意

该方案不能用于交换任意实际的信息

仅允许安全地共享一个密钥, 用于后续通讯中使用对称密码体制

- 目前, p 的长度至少应为 512 比特

Diffie-Hellman密钥交换协议 中间人攻击



如何防止中间人攻击

- 需要对传送的信息进行认证保护，以使通信双方知道到底在与谁进行密钥交换

RSA 加密方案

1. 流程

- 系统建立
 - 随机选择大素数 p 、 q ，计算 $n = pq$
 - 随机选取 $e < \phi(n)$ ，且 $\gcd(e, \phi(n)) = 1$
 - 计算 d ，使 $ed \equiv 1 \pmod{\phi(n)}$
 - (e, n) 为公钥， d 为私钥
- 加密： $c = m^e \bmod n$ ， $m \in \mathbb{Z}_n^*$
- 解密： $m = c^d \bmod n$

2. 解密过程的正确性

- 根据欧拉定理有：
 - $m^{k\phi(n)+1} \equiv m \pmod{n}$
- 在 RSA 中
 - 有 e 和 d 满足 $ed \equiv 1 \pmod{\phi(n)}$
 - 因此，存在 k 使得 $ed = 1 + k\phi(n)$
- 因此
 - $c^d \equiv (m^e)^d \equiv m^{1+k\phi(n)} \equiv m \pmod{n}$
- 欧拉定理** 设 $n \geq 2$ ，如果 $\gcd(a, n) = 1$ ，则 $a^{\phi(n)} \equiv 1 \pmod{n}$

RSA 加密方案 RSA 中的计算问题

① 如何产生两个大素数？

- 素性检测

② 如何计算 d ？

- d 满足 $ed \equiv 1 \pmod{\phi(n)}$, 且 $\gcd(e, \phi(n))=1$
- 实际上, d 和 e 在模 $\phi(n)$ 下互为逆元
- 用扩展的欧几里得算法计算

③ 加解密

- 实际都是对一个整数的幂运算, 再求模
- 使用 乘法链算法 可以降低计算复杂度

RSA 加密方案 安全性

① 分解模数 n

- 理论上, RSA 的安全性基于分解大整数是困难的这一假设
 - 如果模数 n 被成功地分解为 p 和 q , 则可立即计算出 $\phi(n)$, 从而能确定 e 在 $\phi(n)$ 下的乘法逆元 d , 便可成功破译 (解密密文)
- 但数学上至今未能证明分解 n 就是攻击 RSA 的最佳方法, 也未证明分解大整数就是 NP 问题。
- 随着人类计算能力的不断提高, 原来被认为是不可能分解的大数已被成功分解

② 目前, n 的长度至少介于 1024 到 2048 比特之间

RSA 加密方案 对 RSA 的攻击举例——共模攻击

为方便起见, 可能给每个用户相同的模数 n , 不同的加解密密钥, 但这样做是不安全的

设两个用户的公钥分别为 e_1 和 e_2 , 且 $\gcd(e_1, e_2)=1$, 明文是 m , 密文分别是

$$c_1 = m^{e_1} \pmod{n}, \quad c_2 = m^{e_2} \pmod{n}$$

攻击者截获 c_1 和 c_2 后, 可以这么做:

① 计算两个整数 r 和 t , 满足

$$re_1 + te_2 = 1 \text{ (扩展的欧几里得算法), 其中一个必为负数, 设为 } r$$

① 计算 c_1 在模 n 下的逆元 c_1^{-1}

② 恢复明文 m 如下

$$(c_1^{-1})^{r} c_2^{t} \equiv c_1^{r} c_2^{t} \equiv m^{re_1 + te_2} \equiv m \pmod{n}$$

RSA 加密方案 参数选择

- 为保证算法安全性, 要求 p 和 q 满足以下要求

- ① p 、 q 不能相同, 同时既不要太接近, 又不能差别太大
- ② p 、 q 的长度应该差不多(512 位)
- ③ p 、 q 是安全素数

- **安全素数**: 形如 $p=2p_1+1$ 的素数, p_1 也是素数

- **目的**: 确保 $p-1$ 、 $q-1$ 都有大的素因子, 以增加猜测 $\phi(n)$ 的难度

- ④ $\gcd(p-1, q-1)$ 应当小

- 另外, d 要大于 $1/3n^{1/4}$, 否则易遭到连分式算法(Wiener's attack)的攻击
- e 的选择也要注意
 - 一般选 $e=3, 17, 65537(2^{16}+1)$, 二进制表示只有两个 1
 - 加密前先用随机比特填充 m , 使之与 n 有相同的长度。PEM、PGP 等软件都是这么做的。
 - 这样, 即使多个用户使用相同的 e , 使用这三个值也没有任何安全问题, 且可提高计算速度

RSA 加密方案 缺点

- 安全性问题
 - 还无法证明对 RSA 的破译是否等同于大整数分解问题
 - 只是到目前为止, RSA 似乎是安全的
- 速度问题
 - 增大 p 和 q 将使计算开销指数级增长
- 选择合适的 p 、 q 对于普通用户来说比较困难
 - 因为他们没有相关的知识。

对称密码和公钥密码的比较

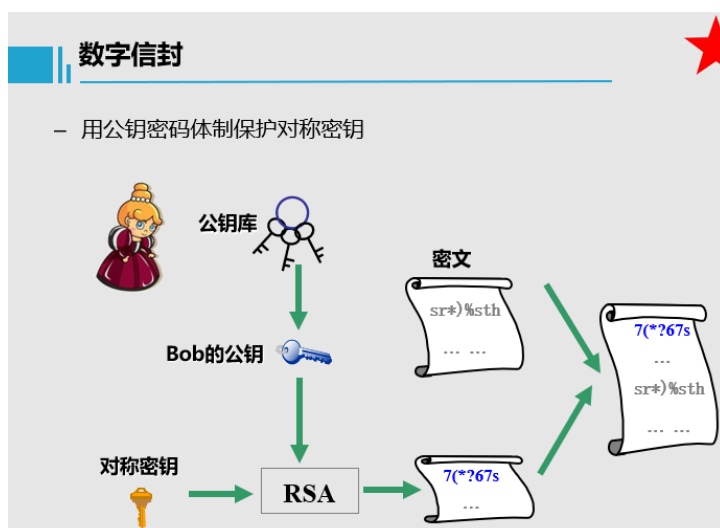
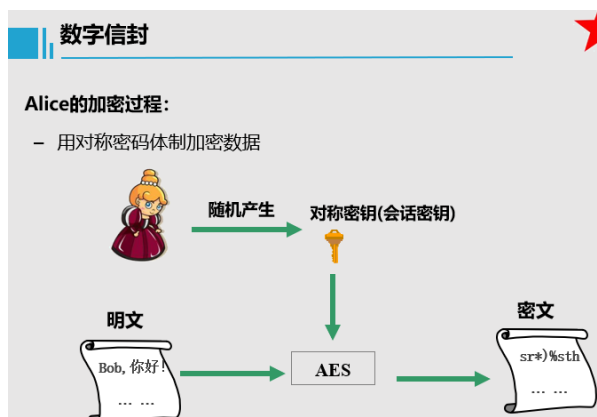
对称密码	公钥密码
<p>• 一般要求:</p> <ol style="list-style-type: none"> 1、加密解密用相同的密钥 2、收发双方必须共享密钥 <p>• 安全性要求:</p> <ol style="list-style-type: none"> 1、密钥必须保密 2、没有密钥, 解密不可行 3、知道算法和若干密文不足以确定密钥 	<p>• 一般要求:</p> <ol style="list-style-type: none"> 1、加密解密使用不同的密钥 2、发送方拥有公钥或私钥, 而接收方拥有另一个密钥 <p>• 安全性要求:</p> <ol style="list-style-type: none"> 1、私钥必须保密, 公钥可以公开 2、没有私钥, 解密不可行 3、知道算法和公钥, 以及若干密文不能确定私钥

公钥密码小结

- 缺点
 - 计算速度慢
 - 密钥长
 - 应用历史短
- 误区
 - 公钥密码更安全
 - 对称密码体制已经过时
 - RSA 最快的情况也比 DES 慢上 100 倍，速度一直是公钥密码体制的缺陷。
 - 公钥分发十分简单

数字信封

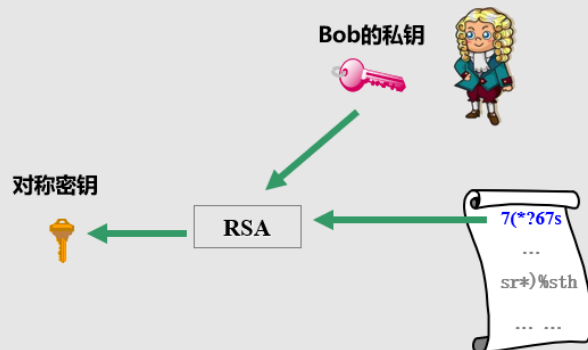
- 对称密码、公钥密码各有长短
 - 对称密码体制：高效，适用于加密大量数据，但密钥分发困难
 - 公钥密码体制：效率低，不适于加密大量数据，但灵活
- 数字信封(Digital Envelope)
 - 目的：取长补短，利用对称密钥加密数据，用接收者公钥保护该对称密钥



数字信封

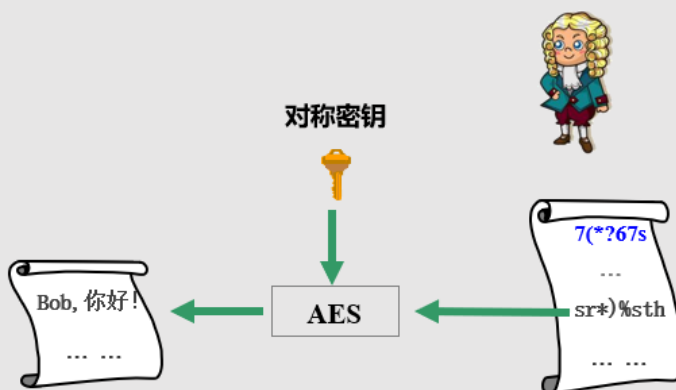
- Bob在接收端的解密过程:

- 用私钥恢复对称密钥



数字信封

- 用对称密钥恢复明文



数字信封 总结

发送方

- ① 随机生成对称密钥，用该密钥加密明文
- ② 用接收方的公钥加密该对称密钥
- ③ 将①、②的结果发送给接收方

- 接收方

- ① 用自己的私钥恢复对称密钥
- ② 用该对称密钥解密密文

- 考虑数据完整性的话，还应加入完整性技术