

计算机信息安全

— 密钥管理

唐飞龙

Email : tang-fl@cs.sjtu.edu.cn



密钥管理技术



- 密钥管理的原则
- 密钥的层次结构
- 密钥的生命周期
- 秘密密钥的分发和协商
- 公开密钥的分发
- 密钥管理框架



1. 密钥管理技术

在密码学中引入密钥的好处：

- (1) 密钥作为密码变换的参数，起到“钥匙”的作用，通过加密变换操作，可以将明文变换为密文，或者通过解密变换操作，将密文恢复为明文
- (2) 在一个加密方案中不用担心算法的安全性，即可以认为算法是公开的，只要保护好密钥就可以了，很明显，保护好密钥比保护好算法要容易得多；
- (3) 可以使用不同的密钥保护不同的秘密，这意味着当有人攻破了个密钥时，受威胁的只是这个被攻破密钥所保护的信息，其他的秘密依然是安全的，由此可见密钥在整个密码算法中处于十分重要的中心地位。

密钥管理原则

(1) 区分密钥管理的策略和机制

- 策略是密钥管理系统的高级指导。策略着重原则指导，而不着重具体实现
- 密钥管理机制是实现和执行策略的技术机构和方法。

(2) 全程安全原则

必须在密钥的产生、存储、备份、分发、组织、使用、更新、终止和销毁等的全过程中对密钥采取妥善的安全管理。

(3) 最小权利原则

应当只分发给用户进行某一事务处理所需的最小的密钥集合。

(4) 责任分离原则

一个密钥应当专职一种功能，不要让一个密钥兼任几种功能。

密钥管理原则

(5) 密钥分级原则

可减少受保护的密钥的数量，又可简化密钥的管理工作。
一般可将密钥划分为三级：主密钥，二级密钥，初级密钥。

(6) 密钥更新原则

密钥必须按时更新。否则，即使是采用很强的密码算法，使用时间越长，敌手截获的密文越多，破译密码的可能性就越大。

(7) 密钥应当有足够的长度

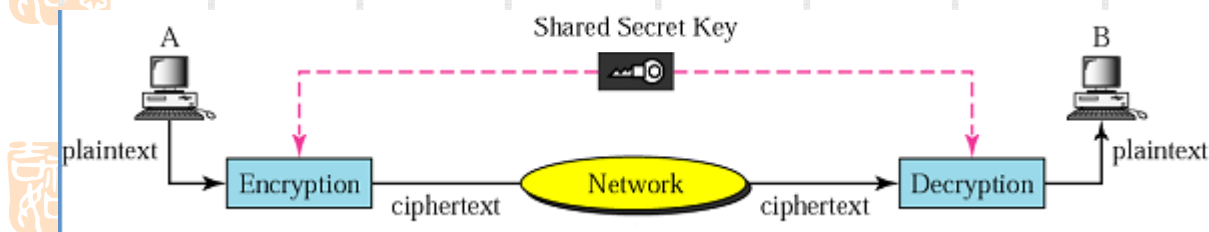
密码安全的一个必要条件是密钥有足够的长度。密钥越长，密钥空间就越大，攻击就越困难，因而也就越安全。

(8) 密码体制不同，密钥管理也不相同

由于传统密码体制与公开密钥密码体制是性质不同的两种密码，因此它们在密钥管理方面有很大的不同。

密钥分配问题

- 在使用对称密码体制进行保密通信时，需要考虑
 - 如何在通信双方建立共享密钥
 - 如何实现双方经常/定期更新密钥 如何实现双方经常/定期更新密钥
 - 如何给双方分配密钥
- 如果不能解决这些问题，安全体制就不能发挥保密通信的效能。



密钥管理技术



- 密钥管理的原则
- 密钥的层次结构
- 密钥的生命周期
- 秘密密钥的分发和协商
- 公开密钥的分发
- 密钥管理框架



2. 密钥的层次结构



按照ANSI X.17，密钥可分为以下三个层次：

1. 初级密钥

我们称用以加解密数据的密钥为初级密钥，也叫数据密钥，它位于整个密钥层次体系的最底层。

2. 一般密钥加密密钥

一般密钥加密密钥通常简称为密钥加密密钥，也称为二级密钥

3. 主密钥

主密钥是密钥层次体系中的最高级密钥，主密钥主要用于对密钥加密密钥进行保护



2. 密钥的层次结构



层次化的密钥结构的优点：

1. 安全性强

一般情况下，位于层次化密钥结构中越底层的密钥更换得越快，最底层密钥可以做到每加密一份报文就更换一次。另外，在层次化的密钥结构中，下层的密钥被破译将不会影响到上层密钥的安全。



2. 可实现密钥管理的自动化

除了主密钥需要由人工装入以外，其他各层的密钥均可以设计由密钥管理系统按照某种协议进行自动地分发更换、销毁等。



密钥管理技术



- 密钥管理的原则
- 密钥的层次结构
- 密钥的生命周期
- 秘密密钥的分发和协商
- 公开密钥的分发
- 密钥管理框架



3. 密钥的生命周期



□ 密钥的产生

1.好的密钥应当具有良好的随机性和密码特性、避免弱密钥的出现

2.不同的密码体制，其密钥的具体生成方法一般不同，与相应的密码体制或标准相联系

3.不同级别的密钥产生的方式一般也不同：

(1) 主密钥通常采用掷硬币、骰子或使用物理噪声发生器的方法来产生。

(2) 密钥加密密钥可以采用伪随机数生成器、安全算法或电子学噪声源产生。

(3) 初级密钥可以在密钥加密密钥的控制下通过安全算法动态的产生。



密钥的生命周期



□ 密钥的存储和备份

- 密钥的安全存储就是要确保密钥在存储状态下的秘密性、真实性和完整性
- 安全可靠的存储介质是密钥安全存储的物质条件，安全严密的访问控制机制是密钥安全存储的管理条件
- 不同级别的密钥应当采用不同的存储方式
- 为了进一步确保密钥和加密数据的安全，对密钥进行备份是必要的。目的是一旦密钥遭到毁坏，可利用备份的密钥恢复出原来的密钥或被加密的数据，避免造成损失



密钥的生命周期



□ 密钥的终止和销毁

- 当密钥的使用期限到期时，必须终止使用该密钥，并更换新密钥
- 终止使用的密钥，一般并不要求立即销毁，而需要再保留一段时间然后再销毁。这是为了确保受其保护的其他密钥和数据得以妥善处理
- 只要密钥尚未销毁，就必须对其进行保护，丝毫不能疏忽大意
- 密钥销毁要彻底清除密钥的一切存储形态和相关信息，使得重复这一密钥成为不可能。这里既包括处于产生、分发、存储和工作状态的密钥及相关信息，也包括处于备份状态的密钥和相关信息



密钥管理技术



- 密钥管理的原则
- 密钥的层次结构
- 密钥的生命周期
- 秘密密钥的分发和协商
- 公开密钥的分发
- 密钥管理框架



4. 秘密密钥的分发和协商

◆ 秘密密钥的分发

1. 离线分发方式

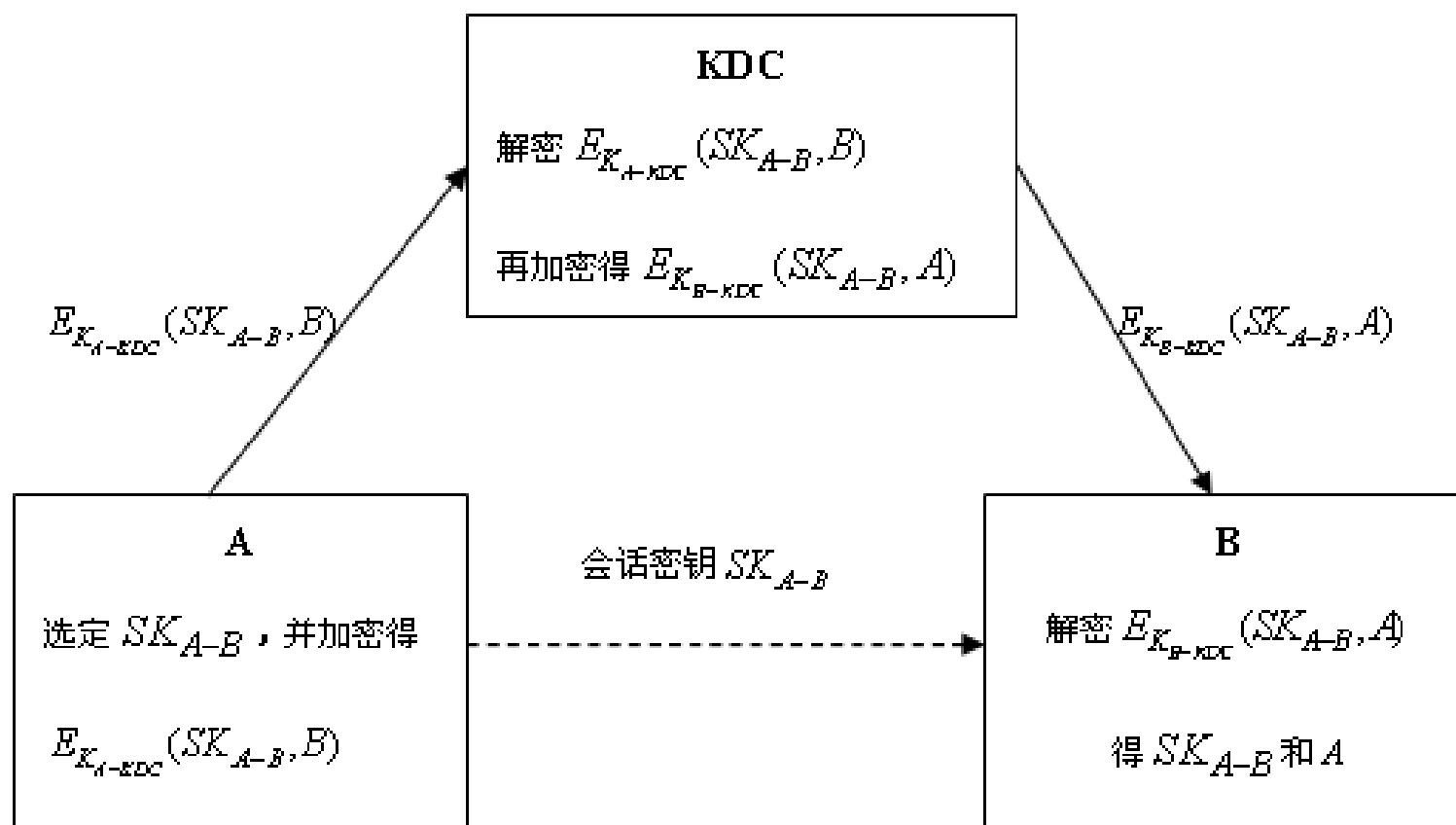
通过非通信网络的可靠物理渠道携带密钥分发给互相通信的各用户。但这种方法有很多缺点，主要包括：随着用户的增多和通信量的增大，密钥量大大增加；密钥的更新很麻烦等

2. 在线分发方式

通过通信与计算机网络的密钥在线、自动分发方式，主要是通过建立一个密钥分发中心（KDC）来实现。在这种方式中，每个用户将与KDC共享一个保密的密钥，KDC可以通过该密钥来鉴别某一个用户

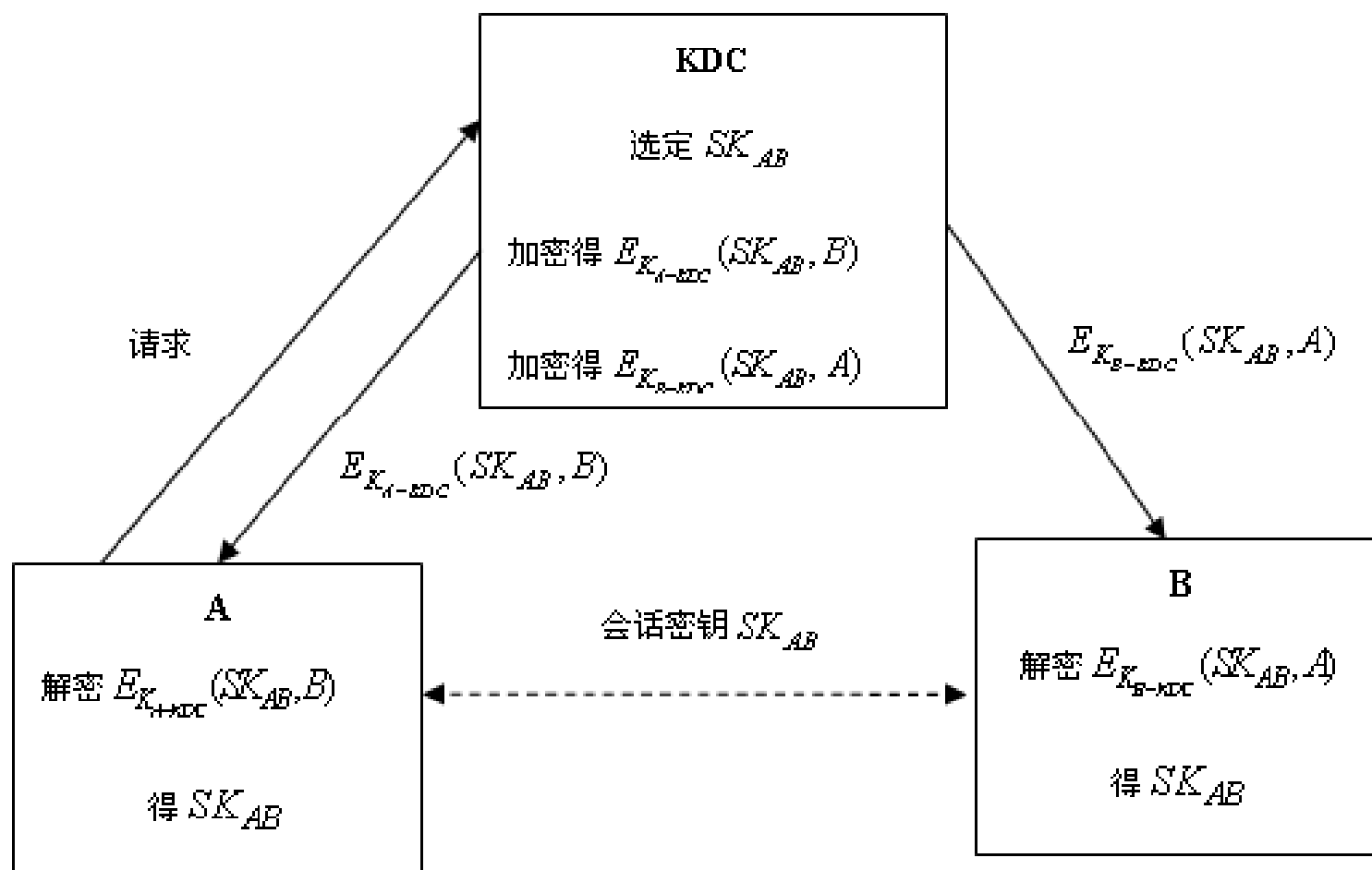
□ 在线密钥分发

(1) 会话密钥由通信发起方生成





□ 在线密钥分发
(2) 会话密钥由KDC生成



◆秘密密钥的协商

1. Diffie-Hellman密钥交换

该协议是Diffie和Hellman在1976年提出的，它是第一个公开发表的公开密钥密码算法，算法的安全性是**基于有限域中计算离散对数的困难性**。
算法如下：

准备：A和Bob协商好一个大素数 q 和 q 的一个本原元 a ， $1 < a < q$ ， q 和 a 为系统的公开参数。

应用：在A与B要通信时，他们可以通过下列步骤协商通信密钥：

(1) A选取大的随机数 X_a ，计算 $Y_a = a^{X_a} \bmod q$ 将 Y_a 传送给B；

(2) B选取大的随机数 X_b ，并计算 $Y_b = a^{X_b} \bmod q$ 将 Y_b 传送给A；

(3) A计算： $K_{ab} = Y_b^{X_a} \bmod q = (a^{X_b} \bmod q)^{X_a} \bmod q = a^{X_a X_b} \bmod q$

B计算： $K_{ba} = Y_a^{X_b} \bmod q = (a^{X_a} \bmod q)^{X_b} \bmod q = a^{X_a X_b} \bmod q$

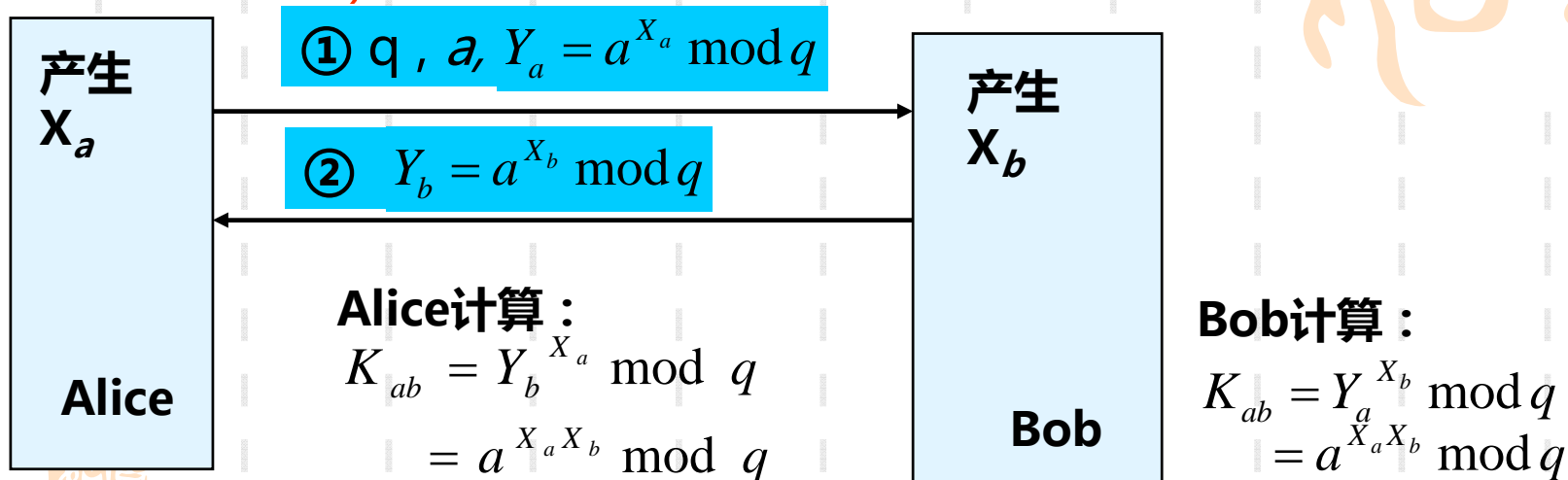
显然， $K_{ab} = K_{ba} = a^{X_a X_b} \bmod q = K_s$

由(3)知A和B已获得了相同的秘密值 K_s ，双方以 K_s 作为加解密钥以传统对称密钥算法进行保密通信

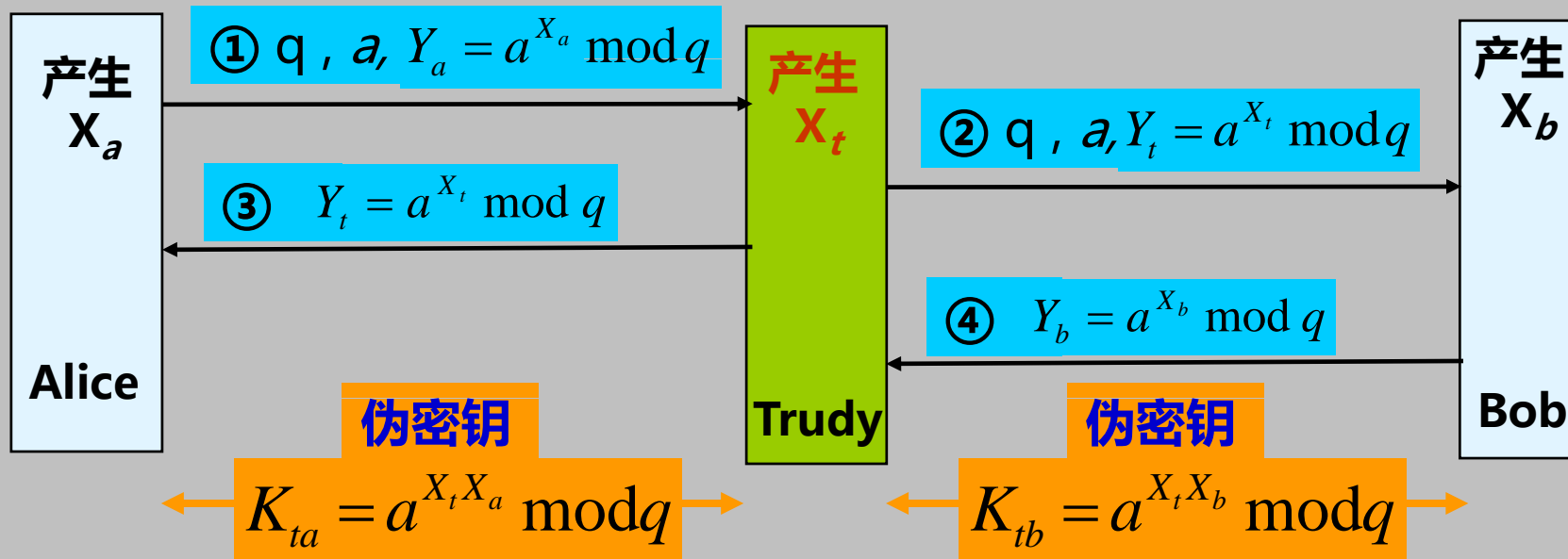
Diffie-Hellman密钥交换协议的安全问题

算法本身的安全性依赖于有限域上计算离散对数的困难性。但协议在实际应用时，一定要引入某种鉴别机制，否则容易受到中间人攻击(man-in-the-middle attack)

正常密钥交换



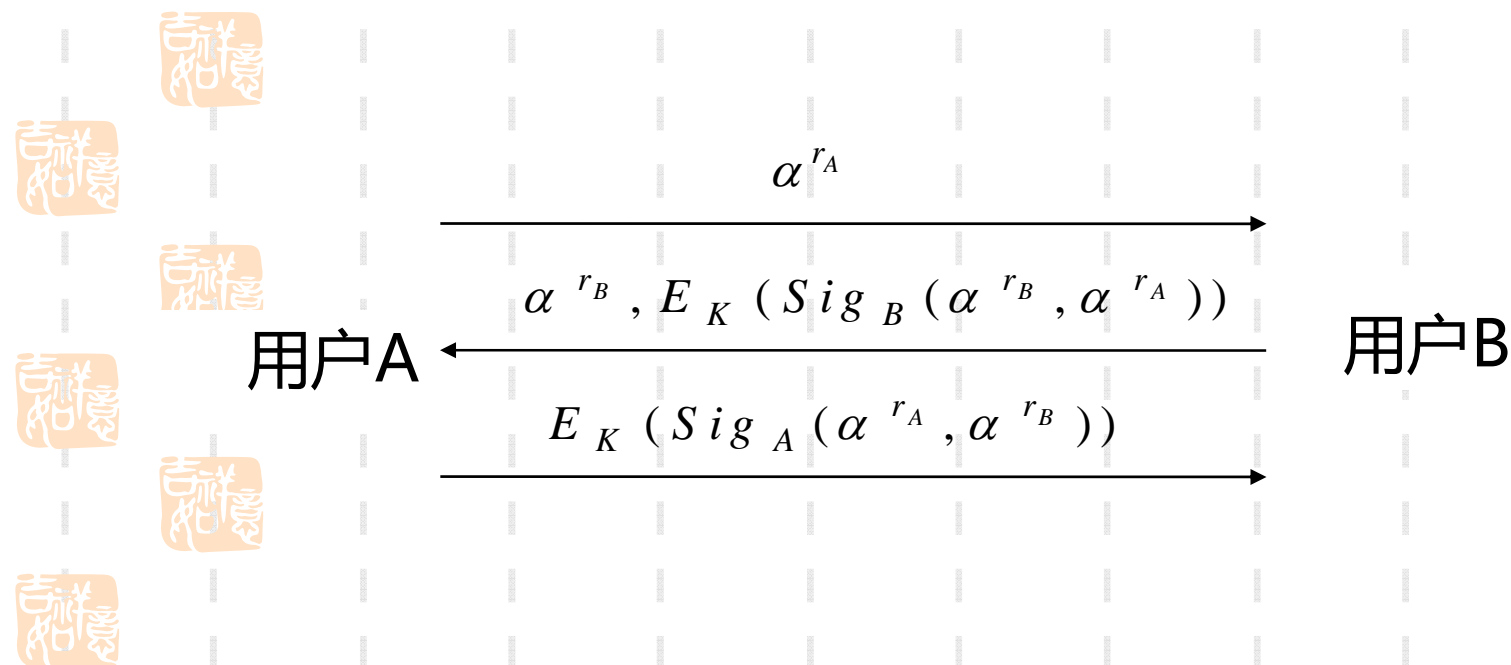
中间人攻击



◆ 秘密密钥的协商

2. 引入鉴别机制的端-端密钥协商协议

端-端协议(STS, Stop-To-Stop)是对Diffie-Hellman密钥交换协议进行修改后得到的。用户A与B之间的消息交换过程可以图解如下:



密钥管理技术



- 密钥管理的原则
- 密钥的层次结构
- 密钥的生命周期
- 秘密密钥的分发和协商
- 公开密钥的分发
- 密钥管理框架




5. 公开密钥的分发




5.1 公开密钥的分发方式

- (1) 公钥的公开发布
- (2) 建立公钥目录
- (3) 带认证的公钥分发(在线服务器方式)

 在建立公钥目录的基础上增加认证功能，这是一种在线服务器式公钥分发解决方案。该方案的缺点是可信服务器必须始终在线。

- (4) 使用数字证书的公钥分发(离线服务器方式)

 采用数字签名技术来确保公钥的真实性和完整性：服务器将用户的标识符和用户的公钥一起签名，便将用户的标识符和用户的公开钥绑定在一起。



公开密钥的分发



□ 公钥证书

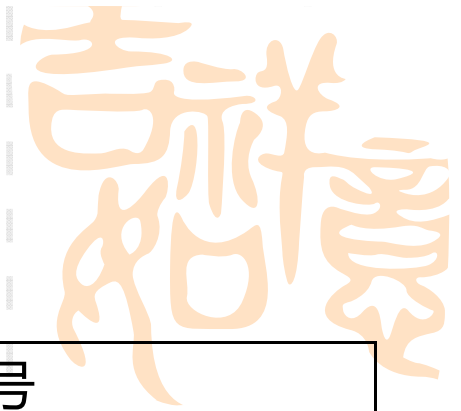
公钥证书是一种包含持证主体标识、持证主体公钥等信息，并由可信任的CA签署的信息集合。公钥证书主要用于确保公钥及其与用户绑定关系的安全。公钥证书的内容主要包括用户的名称、用户的公钥，证书的有效日期和CA的签名等。

由于公钥证书不需要保密，可以在互联网上分发，从而实现公钥的安全分发。又由于公钥证书有CA的签名，攻击者不能伪造合法的公钥证书。因此，只要CA是可信的，公钥证书就是可信的。

使用公钥证书的主要好处是，用户只要获得CA的公钥，就可以安全地获得其他用户的公钥。



公开密钥的分发



□ X.509公钥证书

X.509标准最早于1988年颁布。在此之后又于1993年和1995年进行过两次修改,X.509标准在Internet环境中得到广泛应用。

(1) X.509证书的格式

| |
|-------------------|
| 版本号 |
| 证书序列号 |
| 签名算法标识符 |
| 颁发者的名称 |
| 有效期(不早于/不晚于) |
| 主体名称 |
| 主体的公钥信息(算法标识、公钥值) |
| 颁发者惟一标识符(可选) |
| 主体惟一标识符(可选) |
| 扩展项(可选) |
| 颁发者的签名 |



公开密钥的分发



(2) 数字证书的管理

① 数字证书的签发

由认证机构CA负责

② 数字证书的更新

认证机构可以定期更新所有用户的证书，或者根据用户请求来更新特定用户的证书。

③ 数字证书的查询

④ 数字证书的作废

数字证书都有一定的使用期限；在某些特殊情况下(如对应的私钥已经失密、名字更改、主体与认证机构的关系已经改变等)要求提前作废该证书，认证机构通过周期性地发布并维护证书撤销列表(CRL, Certificate Revocation List)来实现该功能



密钥管理技术



- 密钥管理的原则
- 密钥的层次结构
- 密钥的生命周期
- 秘密密钥的分发和协商
- 公开密钥的分发
- 密钥管理框架



密钥分配



□ 密钥分配方式

- 1. 由A选定，物理地传送给B
- 2. 第三方选定，物理地传送给A和B
- 3. 使用A和B共有的密钥加密后传送给另一方
- 4. A和B都有到第三方的加密连接，则由第三方用加密连接传送给A和B

□ 分析

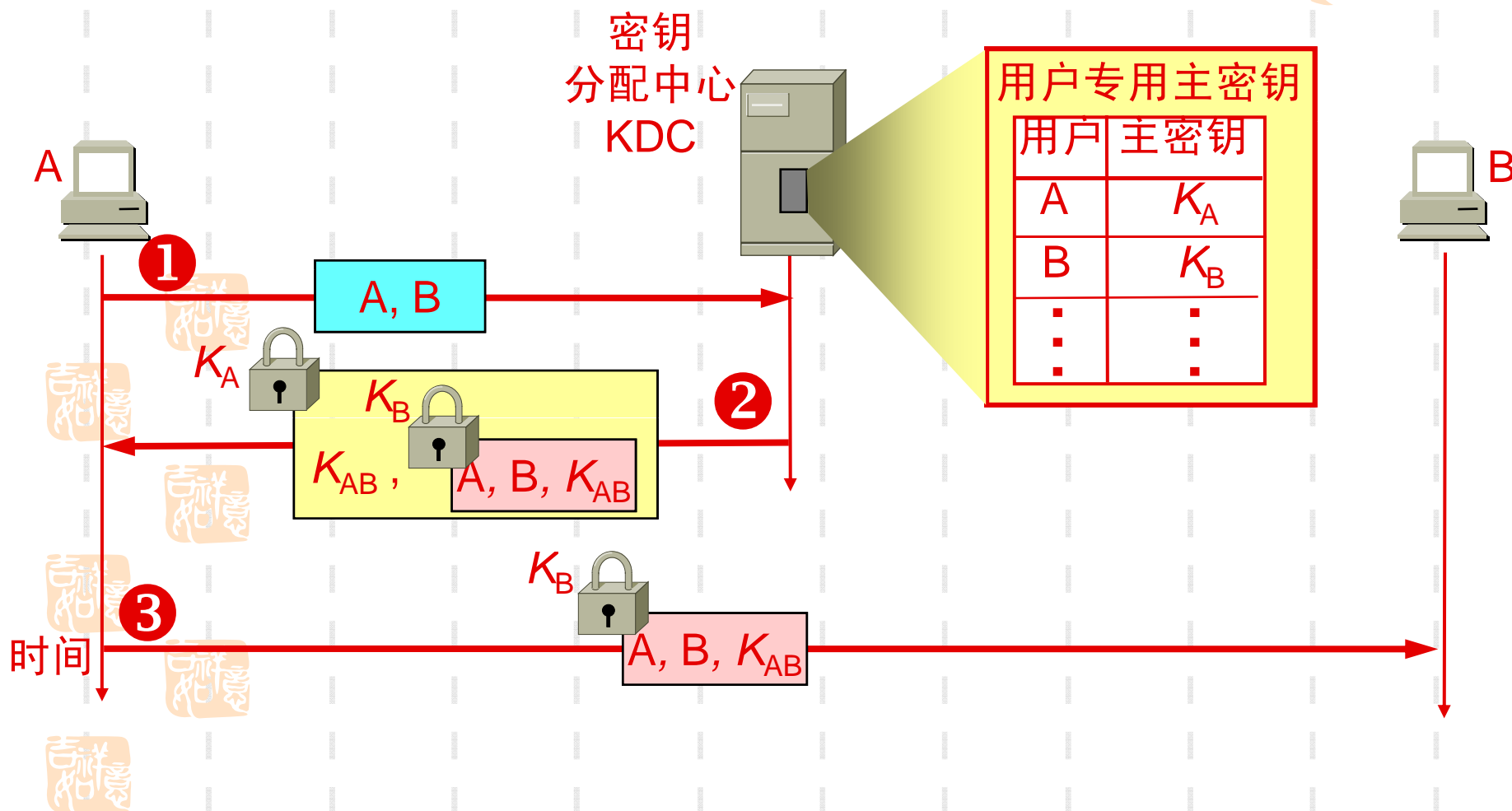
- 1, 2需人工传递，对于链路加密是合理的，对于端到端加密，密钥分配难度比较大
- 3 一旦暴露一个密钥，后续密钥都暴露了
- 4 适合于端到端加密，有许多变体



1. 对称密钥的分配

- 目前常用的密钥分配方式是设立**密钥分配中心** KDC (Key Distribution Center)。
- KDC 是大家都信任的机构，其任务就是给需要进行秘密通信的用户临时分配一个会话密钥（仅使用一次）。
- 用户 A 和 B 都是 KDC 的登记用户，并已经在 KDC 的服务器上安装了各自和 KDC 进行通信的**主密钥**（master key） K_A 和 K_B 。“主密钥”可简称为“密钥”。

对称密钥的分配



基于KDC的对称密钥分配

- 1. 用户A向密钥分配中心送明文(A, B), 说明想和用户B通信, 即申请会话时所用的会话密钥。
- 2. KDC收到申请后, 从用户专用主密钥文件中找出用户A和B的主密钥KA和KB (主密钥是用来加密会话密钥的), 同时产生为A和B通信用的会话密钥, 然后传送给A:

$KDC \rightarrow A: E_{KA}(B, SK, T, E_{KB}(A, SK, T))$

- 3. 最后A将上式中的最后部分传送给B:
- 4. B收到此报文并用B的主密钥解密后, 即可得到会话密钥SK, 用户A和B即可用会话密钥SK进行保密通信。

$A \rightarrow B: E_{KB}(A, SK, T)$

KDC管理

- 1.KDC可以为每对用户的每次通信产生一个新的会话密钥
- 2.主密钥也不能长期使用而不进行更换(主密钥必须比会话密钥具有更好的保密性)
- 3. K_A 和 K_B 是KDC分别与用户A和B共享的，故当用户A收到 $E_{K_A}(B, SK, T)$ 时，知道此消息来自KDC，是由KDC签发给用户A的用于向用户B证明其真实身份的证书。
- 4.因此可将证书存放一段时间，每次通信时重复使用。(有效期可由时间戳指明)

基于KDC的公钥分配

- 每个用户只保存自己的秘密密钥和KDC的公开密钥PKAS。用户可以通过KDC获得任何其他用户的公开密钥
- 此处的KDC即**认证中心CA** (Certification Authority)
- 证书将公钥与其对应的实体（人或机器）进行**绑定**

□ C

(

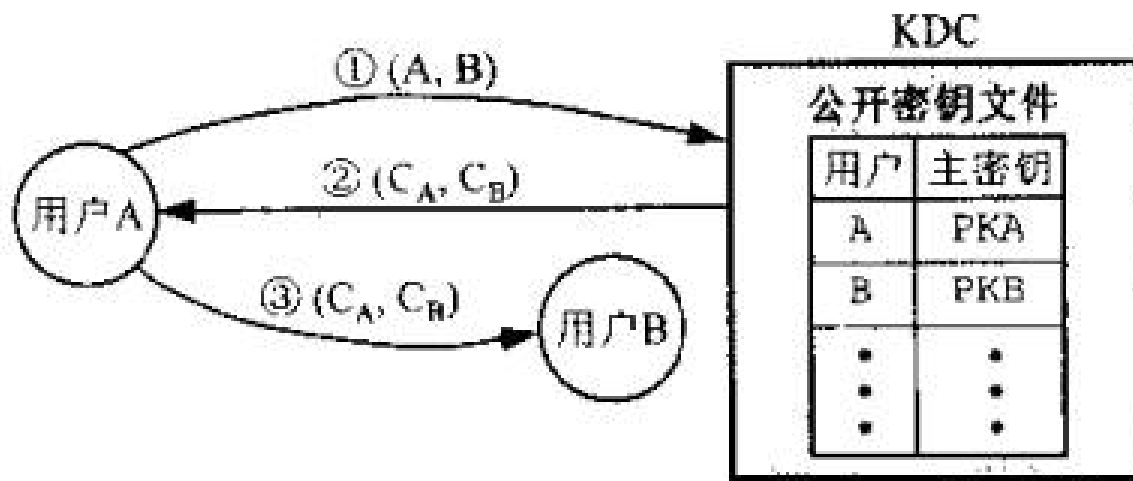
得

其

信

息

。



来的**证书**信息。此证
言的地方获
、钥是否为
业务

图 9-17 公开密钥分配协议

- 首先，A向KDC申请公开密钥。将信息(A, B)发给KDC。KDC给A的响应为：

$KDC \rightarrow A: (C_A, C_B)$

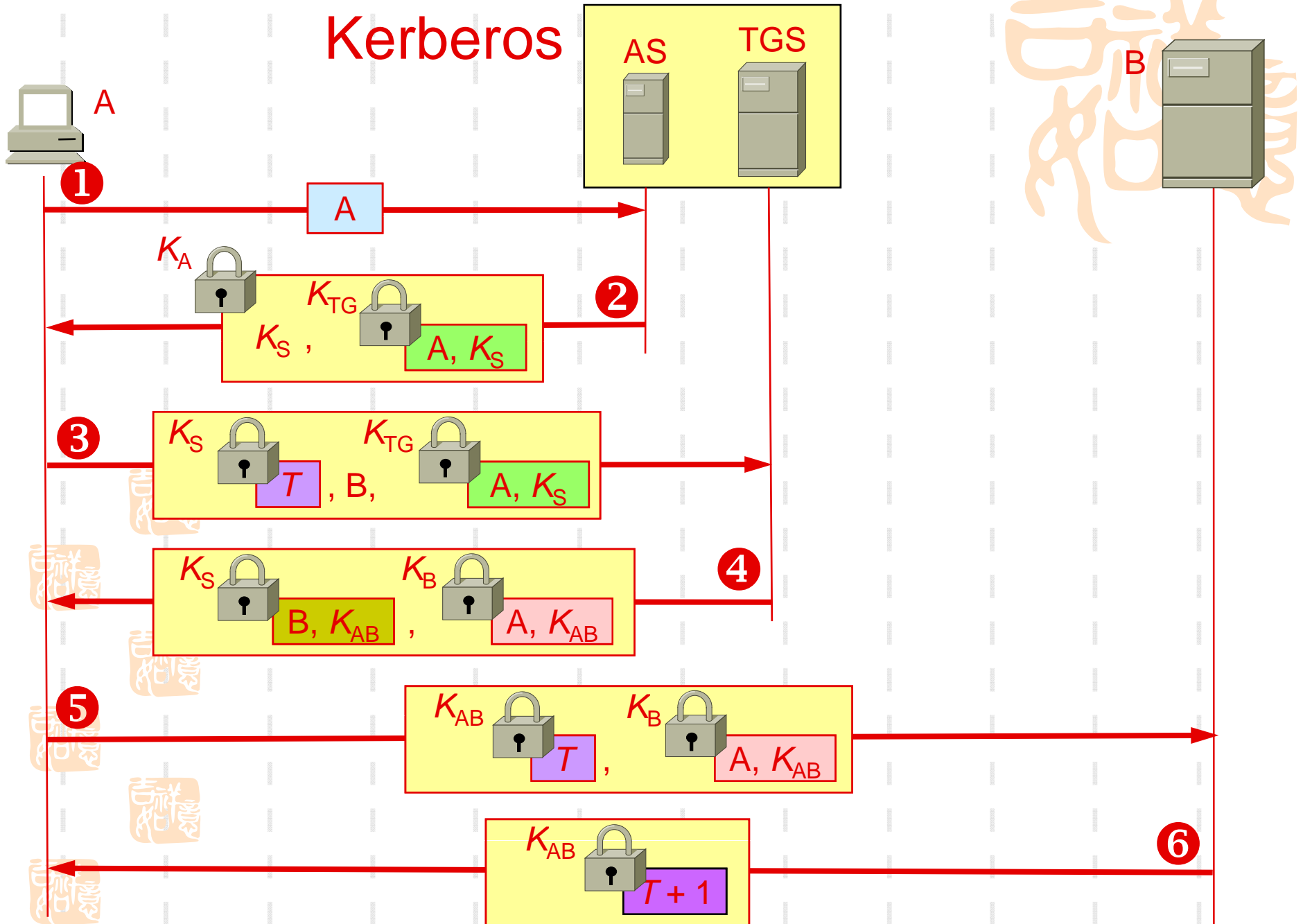
- 这里 C_A 和 C_B 就是前面提到过的证书, 分别含有A和B的公开密钥。

KDC使用其秘密密钥 SK_{AS} 对 C_A 和 C_B 进行了签名，以防止伪造。 C_A 中的 PK_A 提供A核实其公开密钥。

- A从 C_B 中可以获得B的公开密钥 PK_B (时间戳 T_1 和 T_2 的作用仍是防止重放攻击。)
- 最后，A将证书 C_A 和 C_B 传送给B，B获得了A的公开密钥 PK_A ,同时也可以检验自己的公开密钥 PK_B 。

$$C_A = D_{SK_{AS}}(A, PK_A, T_1), \quad C_B = D_{SK_{AS}}(B, PK_B, T_2)$$

Kerberos



层次式密钥分配方案



□ 层次式

- 建立 系列KDC 各个KDC之间存在层次关系
- 最低层的负责某一区域
- 不同区域之间的通信通过上一层KDC进行

□ 优点

- 主密钥的分配工作量减小
- 整个系统鲁棒性强



□ The End!

