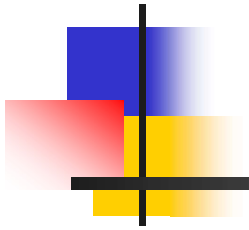
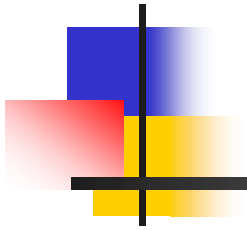
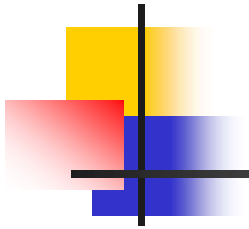


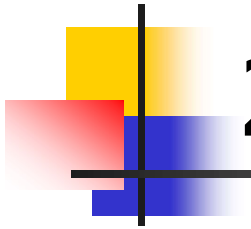
无线网络安全技术



chap2 网安基础

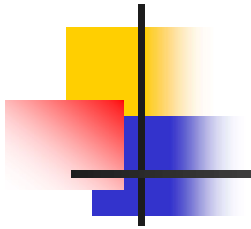


- 身份认证
- 数据机密性
- 数据完整性
- 不可否认
- 访问控制
- 安全协议举例



2.1 认证/鉴别

- 定义：证实客户的真实身份与其所声称的身份是否相符的过程。
- 依据：
  - Something the user know (所知）
    - 密码、口令等
  - Something the user possesses （拥有）
    - 身份证、护照、密钥盘等
  - Something the user is (or How he behaves)
    - 指纹、笔迹、声音、虹膜、DNA等



鉴别机制

- 非密码的鉴别机制
- 基于密码算法的鉴别
  - 采用对称密码算法的机制
  - 采用公开密码算法的机制
  - 采用密码校验函数的机制
- 零知识证明协议

- 非密码的鉴别机制

A. 口令机制

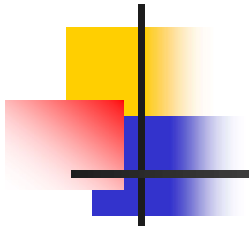
B. 一次性口令机制

C. 基于个人特征的机制

1）指纹识别；

2）声音识别；

3）虹膜等



- 基于密码算法的鉴别
  - 采用对称密码算法的机制
  - 采用公开密码算法的机制
  - 采用密码校验函数的机制

零知识证明技术

- 零知识证明技术可使信息的拥有者无需泄露任何信息就能够向验证者或任何第三方证明它拥有该信息。

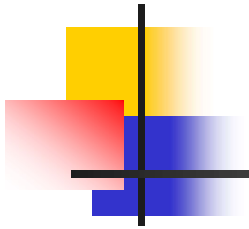
- 双方没有事先建立的任何安全架构，双方是一种动态的认证机制。

- 例如：

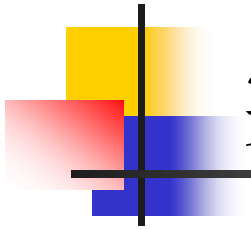
  1）A要向B证明自己拥有某个房间的钥匙，假设该房间只能用钥匙打开锁，而其他任何方法都打不开。这时有2个方法：

  传统方法：（一）A把钥匙出示给B，B用这把钥匙打开该房间的锁，从而证明A拥有该房间的正确的钥匙。

  新的方法：（二）B确定该房间内有某一物体，A用自己拥有的钥匙打开该房间的门，然后把物体拿出来出示给B，从而证明自己确实拥有该房间的钥匙。后面这个方法属于零知识证明。好处在于在整个证明的过程中，B始终不能看到钥匙的样子，从而避免了钥匙的泄露。



- 2）A拥有B的公钥，A没有见过B，而B见过A的照片，偶然一天2人见面了，B认出了A，但A不能确定面前的人是否是B，这时B要向A证明自己是B，也有2个方法。
- 方法一：B把自己的私钥给A，A用这个私钥对某个数据加密，然后用B的公钥解密，如果正确，则证明对方确实是B。
- 方法二：A给出一个随机值，B用自己的私钥对其加密，然后把加密后的数据交给A，A用B的公钥解密，如果能够得到原来的随机值，则证明对方是B。后面的方法属于零知识证明。



实体鉴别分类

- 实体鉴别可以是单向的也可以是双向的。
  - 单向鉴别是指通信双方中只有一方向另一方进行鉴别。
  - 双向鉴别是指通信双方相互进行鉴别。

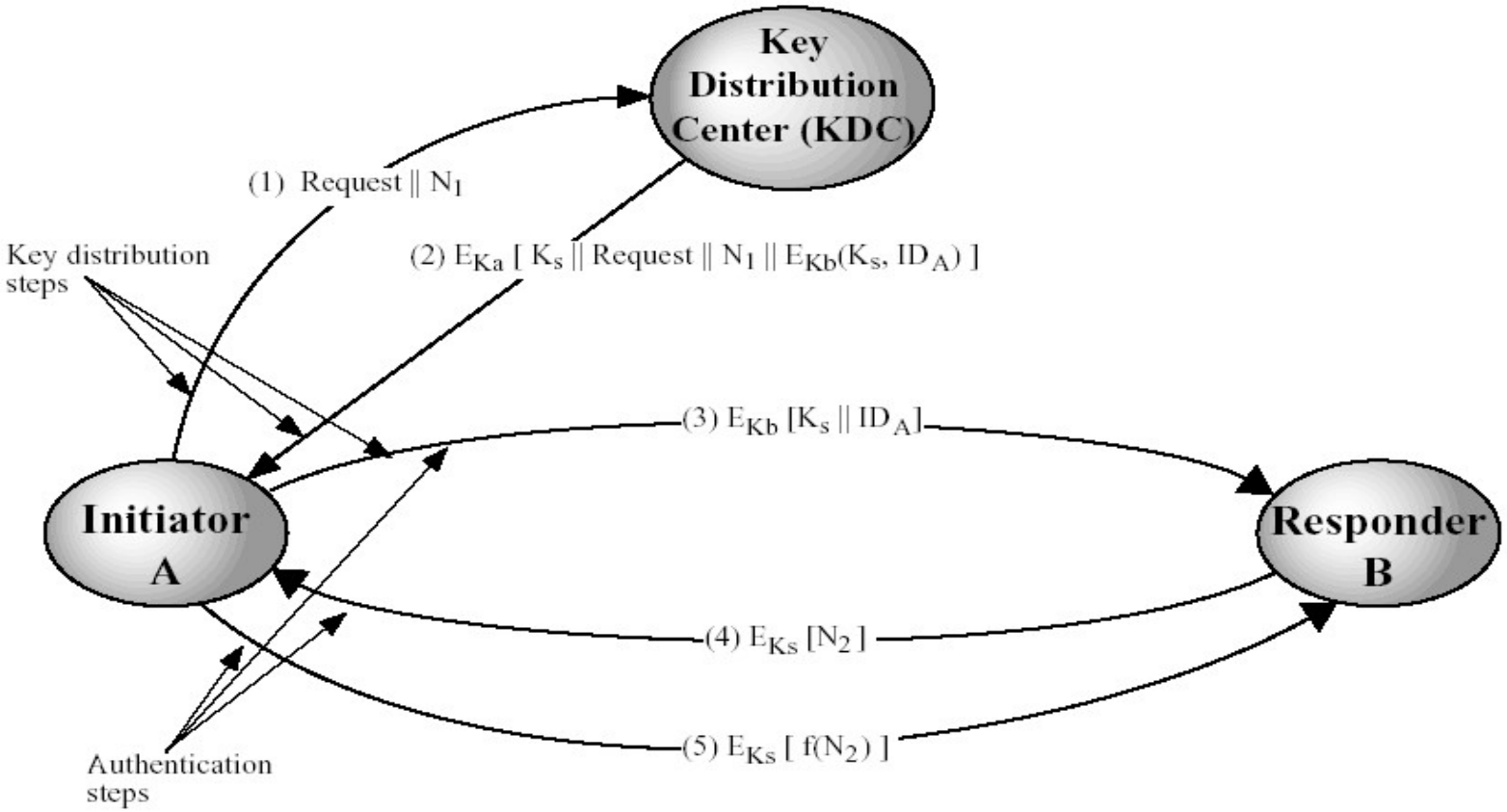
双向鉴别协议

- 最常用的协议。该协议使得通信各方互相认证鉴别各自的身份，然后交换会话密钥。
- 基于鉴别的密钥交换核心问题有两个：
  - 保密性
  - 实效性

（1）为了防止伪装和防止暴露会话密钥，基本鉴别和会话密码信息必须以保密形式通信，这就要求预先存在保密或公开密钥供实现加密使用。

（2）第二个问题也很重要，因为涉及防止消息重放攻击。

Needham/Schroeder Protocol



2.2 数据机密性

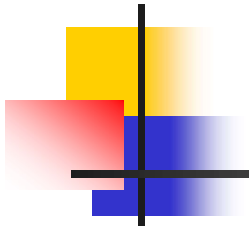
- 受限制的（**restricted)**算法

  算法的保密性基于保持算法的秘密（古典）

- 基于密钥（**key-based)**的算法

  算法的保密性基于对密钥的保密

  （近代：对称+非对称）



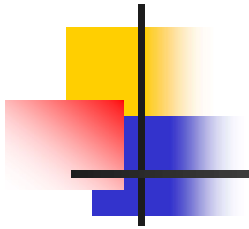
密钥数量

对称密码算法（symmetric cipher)

- 加密密钥和解密密钥相同，或实质上等同，即从一个易于推出另一个
- 又称秘密密钥算法或单密钥算法

非对称密钥算法（asymmetric cipher)

- 加密密钥和解密密钥不相同，从一个很难推出另一个
- 又称公开密钥算法（public-key cipher)
- 公开密钥算法用一个密钥进行加密, 而用另一个进行解密
- 其中的加密密钥可以公开,又称公开密钥（public key)，简称公钥。解密密钥必须保密,又称私人密钥（private key)私钥，简称私钥



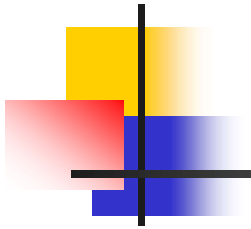
明文处理方式

- **分组密码（block cipher)**

将明文分成固定长度的组，用同一密钥和算法对每一块加密，输出也是固定长度的密文。

- **流密码（stream cipher)**

又称序列密码。序列密码每次加密一位或一字节的明文。

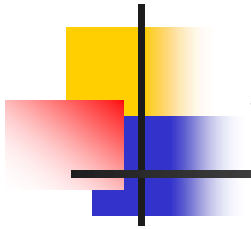


加密方案的安全性

- 无条件安全：无论提供的密文有多少，如果由一个加密方案产生的密文中包含的信息不足以唯一地决定对应的明文
- 除了一次一密的方案外，没有无条件安全的算法
- 安全性体现在任意一条：
  - 破译的成本超过加密信息的价值
  - 破译的时间超过该信息有用的生命周期

密钥搜索所需平均时间

密钥大小（位）	密码算法	密钥个数	每纳秒执行一次解密所需的时间	每纳秒执行一万次解密所需的时间
56	DES	2^56 ≈ 7.2*10^16	2^55 ns ≈ 1.125年	1小时
128	AES	2^128 ≈ 3.4*10^38	2^127 ns ≈ 5.3*10^21年	5.3*10^17年
168	Triple DES	2^168 ≈ 3.7*10^50	2^167 ns ≈ 5.8*10^33年	5.8*10^29年
192	AES	2^192 ≈ 6.3*10^57	2^191 ns ≈ 9.8*10^40年	9.8*10^36年
256	AES	2^256 ≈ 1.2*10^77	2^255 ns ≈ 1.8*10^60年	1.8*10^56年



对称密码算法

- DES秘钥过短，56bit；
- 1997年9月12日，美国联邦登记处公布了正式征集AES候选算法的通告。对AES的基本要求是：比三重DES快、至少与三重DES一样安全、数据分组长度为128比特、密钥长度为128/192/256比特。

电子密码本模式Electronic Codebook, ECB

- 明文分成64的分组进行加密，必要时填充，每个分组用同一密钥加密，同样明文分组加密得相同密文

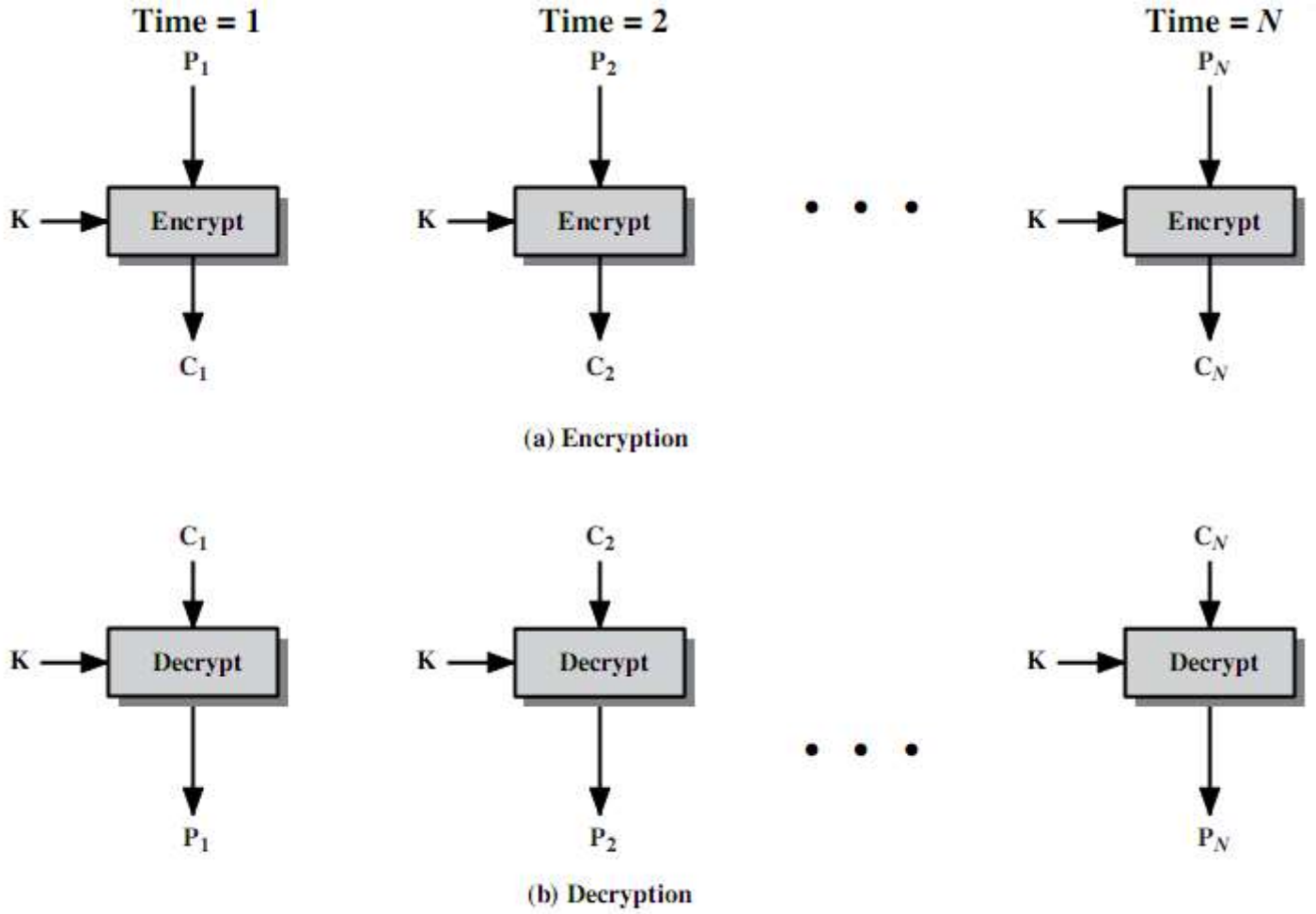


Figure 6.3 Electronic Codebook (ECB) Mode

密文分组链接模式Cipher Block Chaining (CBC)

- 加密输入是当前明文分组和前一密文分组的异或，形成一条链，使用相同的密钥， 这样每个明文分组的加密函数输入与明文分组之间不再有固定的关系

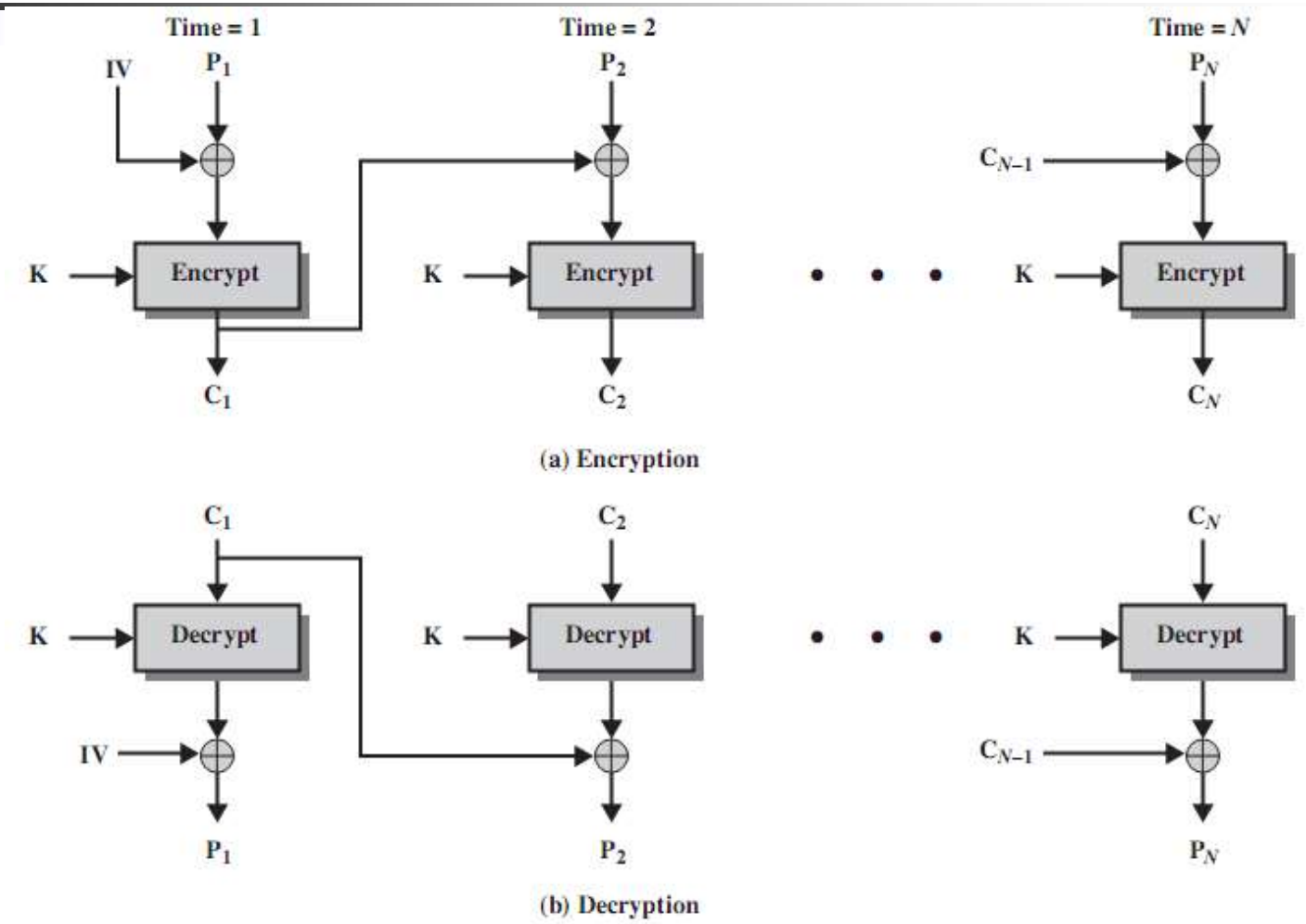


Figure 6.4 Cipher Block Chaining (CBC) Mode

公钥密码学

- 公开密钥算法是非对称算法，即密钥分为公钥和私钥，因此称双密钥体制
- 双钥体制的公钥可以公开，因此也称公钥算法
- 基于数学函数而不是代替和换位，密码学历史上唯一的一次真正的革命
- 公钥密码学在鉴别系统和密钥交换等安全技术领域起着关键的作用

公开密钥密码:保密和认证

$C = KR_a(KU_b(M))$ $M = KR_b(KU_a(C))$

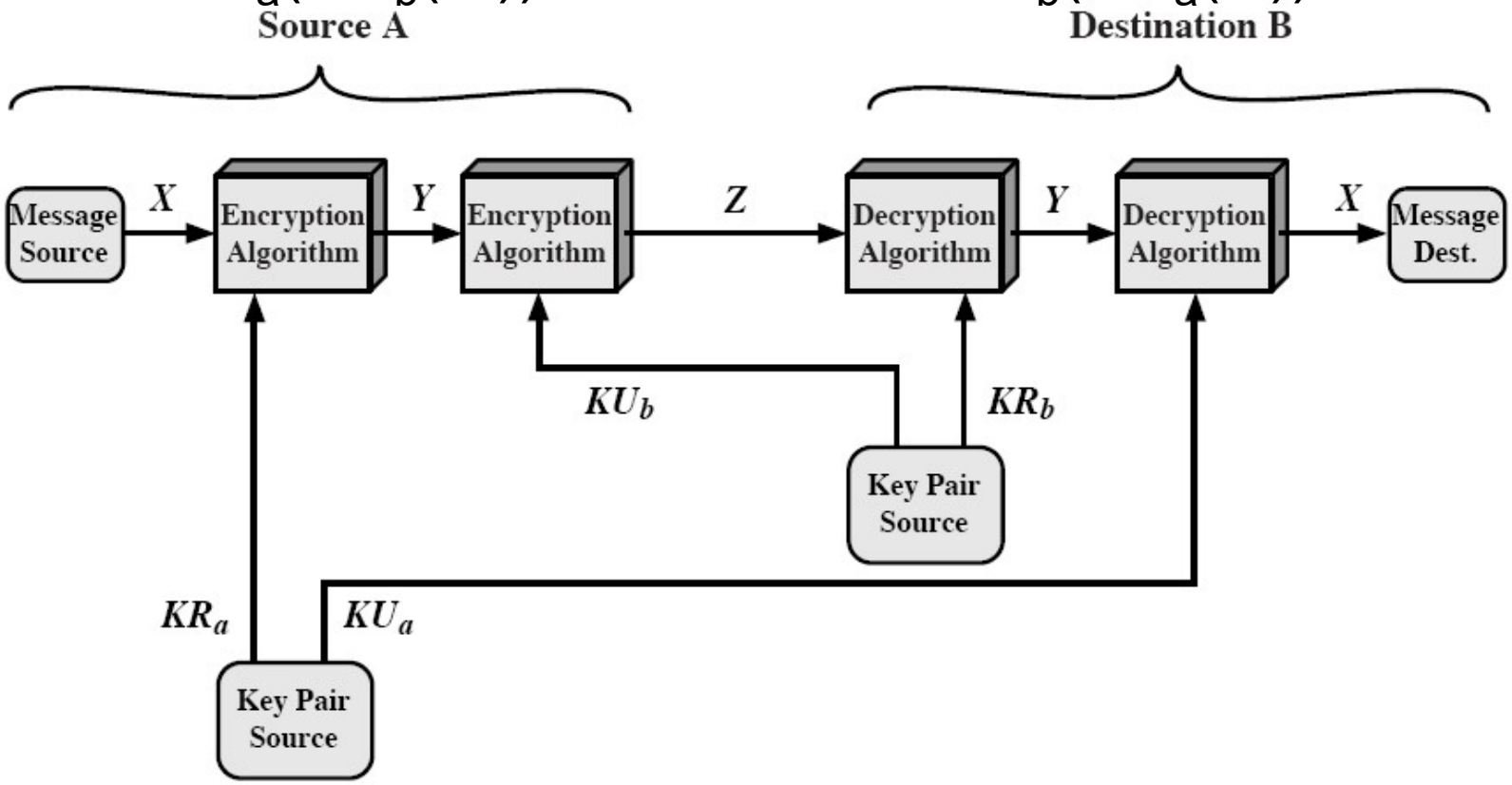
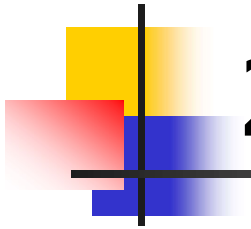
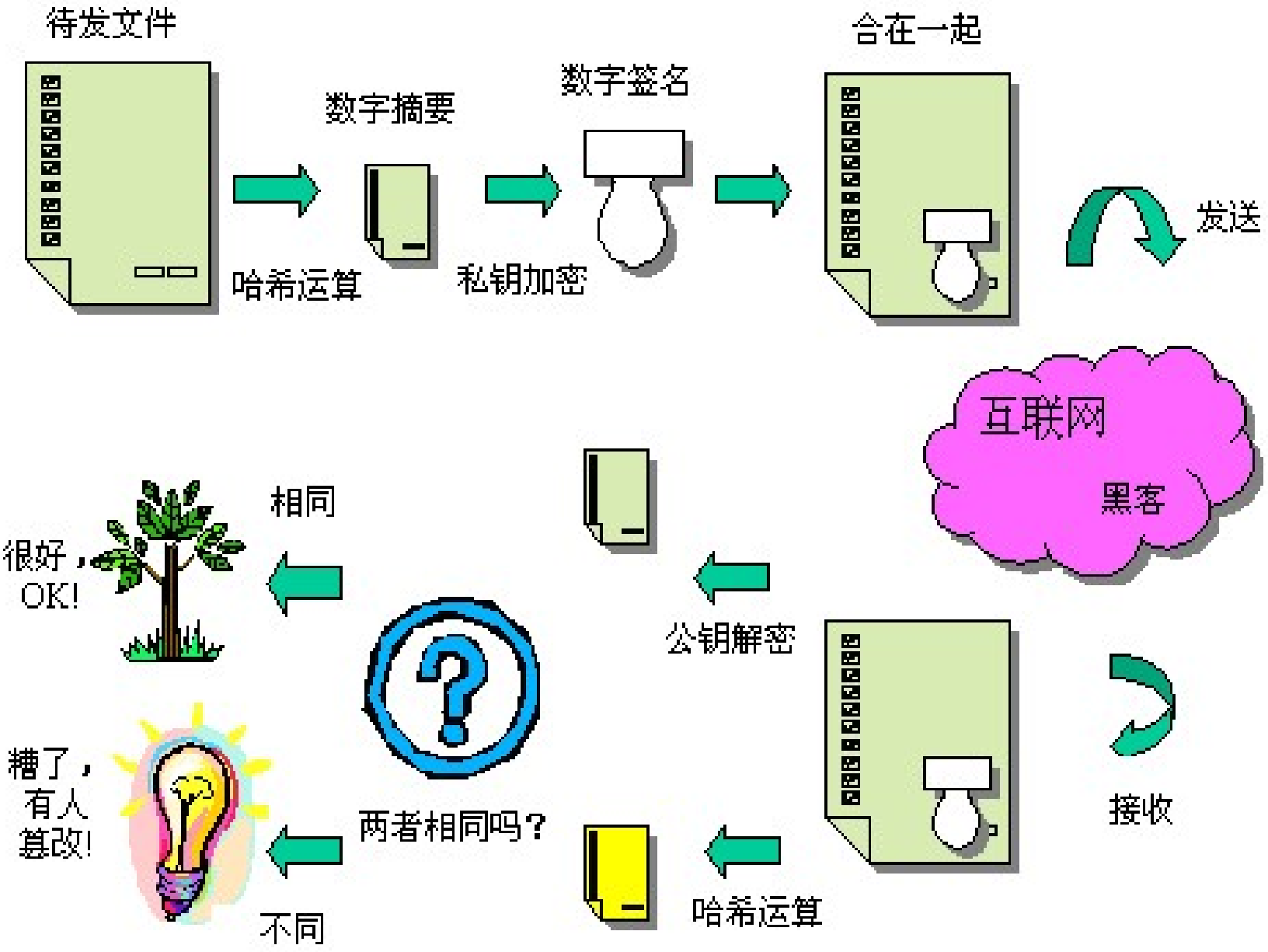


Figure 9.4 Public-Key Cryptosystem: Secrecy and Authentication



2.3 数据完整性

- **MAC**
- **Hash**


待发文件
数字摘要
数字签名
合在一起
哈希运算
私钥加密
发送
互联网
黑客
相同
很好，
OK!
公钥解密
糟了，
有人
篡改!
两者相同吗？
接收
不同
哈希运算

SHA：是FIPS(联邦信息处理标准)所认证的五种安全散列算法

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message digest size	160	224	256	384	512
Message size	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block size	512	512	512	1024	1024
Word size	32	32	32	64	64
Number of steps	80	64	64	80	80

2.4 不可否认（不可抵赖）

- 数字签名是一种认证机制，它使得消息的产生者可以添加一个起签名作用的码字。通过计算消息的散列值并用产生者的私钥加密散列值来生成签名。签名保证了消息的来源和完整性。

(1) $A \rightarrow B: E_{KRa}[M];$

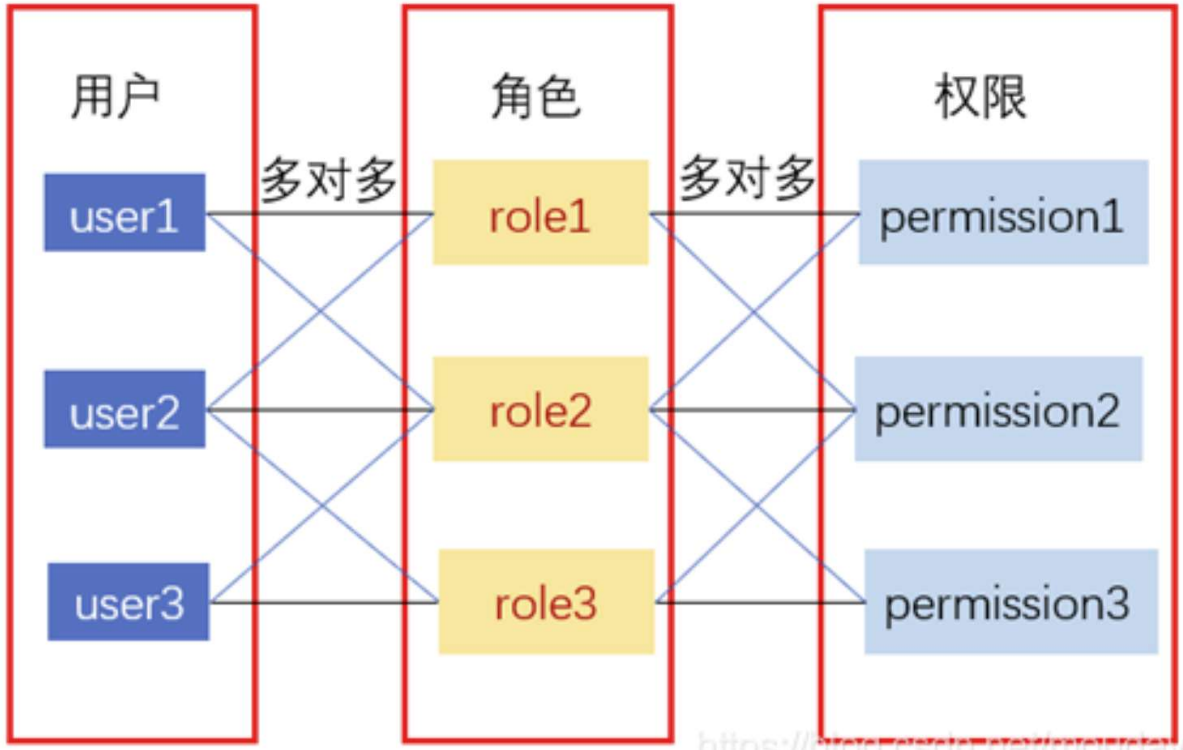
(2) $A \rightarrow B: E_{KUb}\ [E_{KRa}(M)]$

(3) $A \rightarrow B: M||E_{KRa}[H(M)]$

(4) $A \rightarrow B: E_{K}[M||E_{KRa}[H(M)]]:$

提供保密性、鉴别和数字签名

2.5 访问控制


用户
角色
权限
多对多
多对多
user1
user2
user3
role1
role2
role3
permission1
permission2
permission3

2.6 网络安全模型

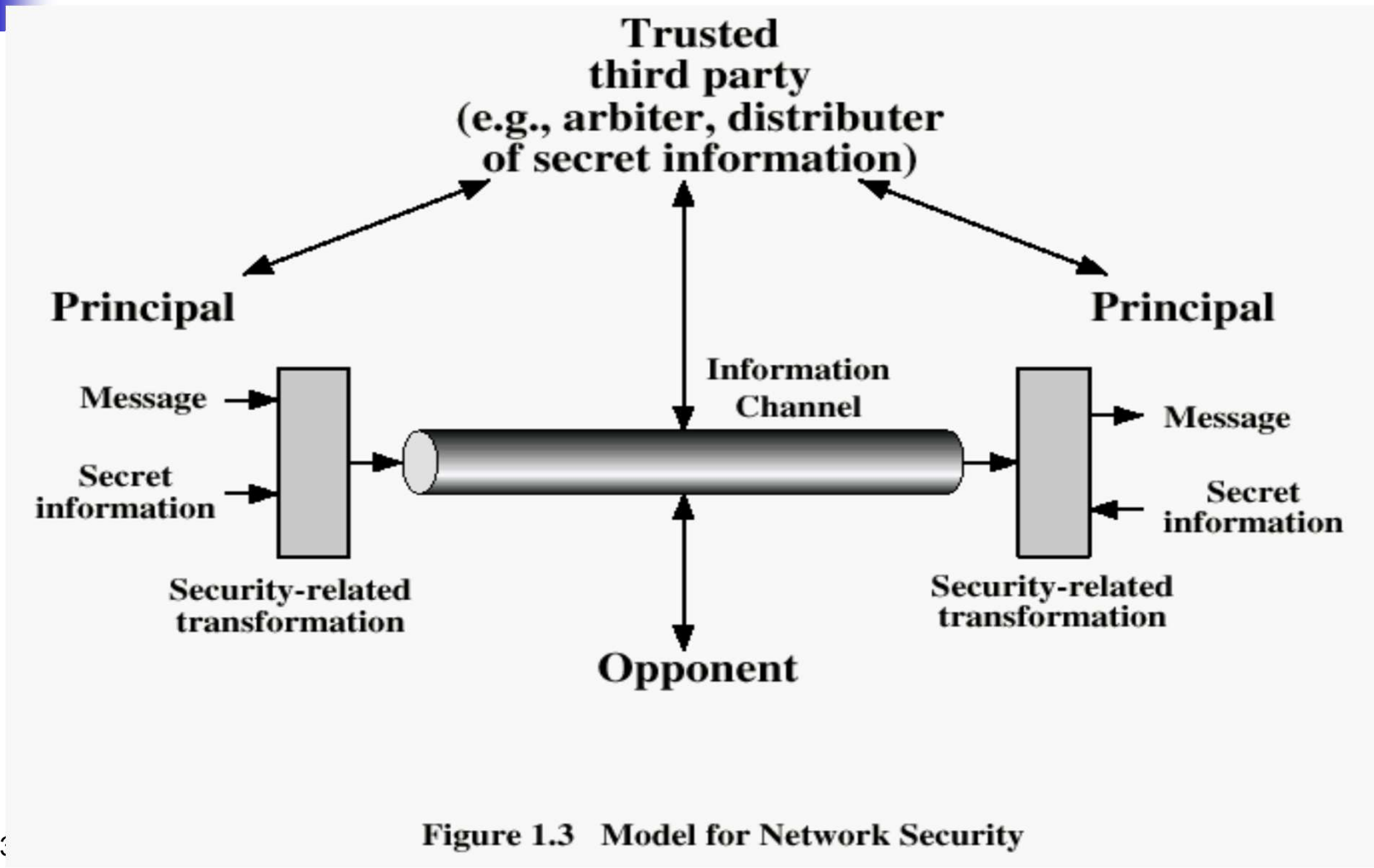


Figure 1.3 Model for Network Security

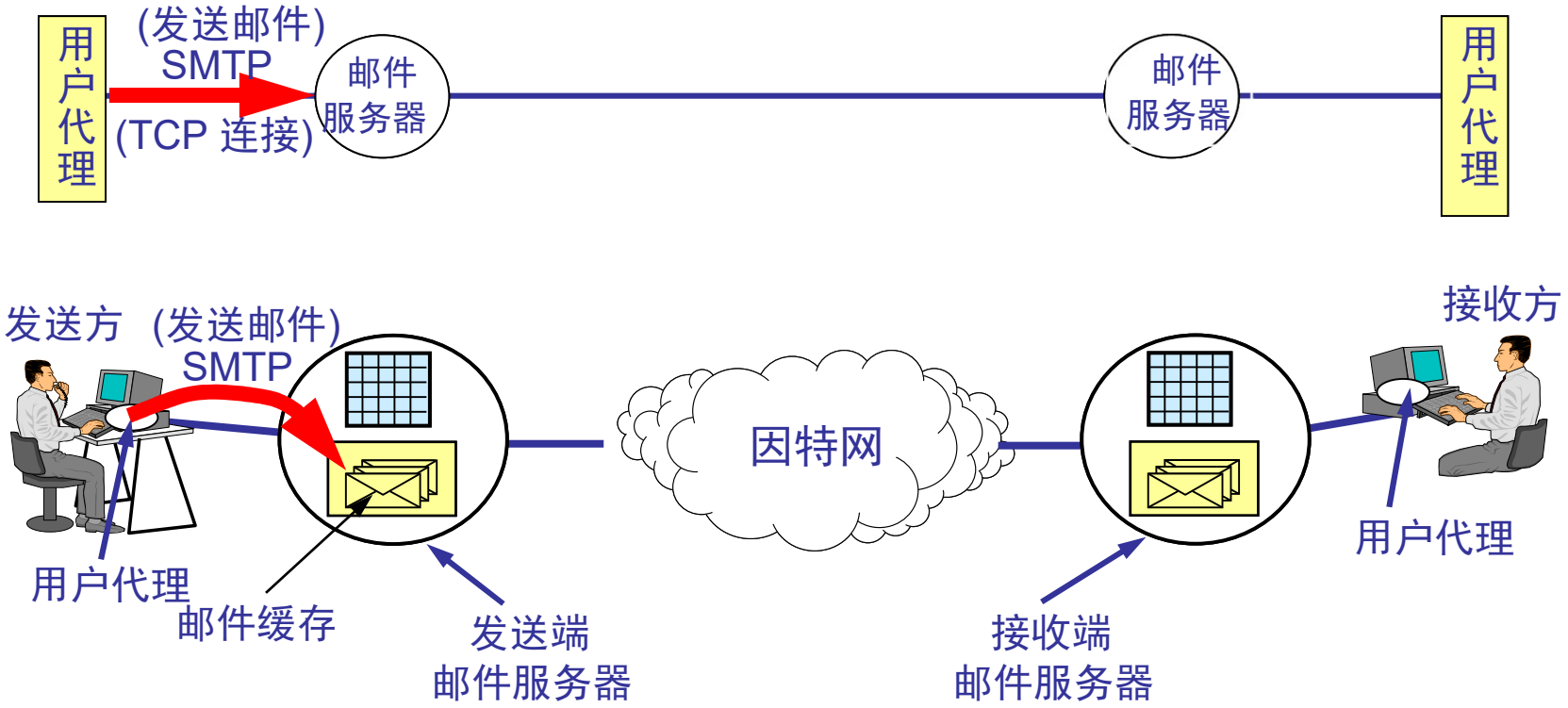
模型说明：

- 设计目的：为了实现安全传输。
- 设计安全服务应包括以下4个方面的内容：
  1. 设计执行安全相关传输的算法，该算法应是攻击者无法攻破的；
  2. 产生算法所用的秘密信息；
  3. 设计分配和共享秘密信息的方法；
  4. 指明通信双方使用的协议，该协议利用安全算法和秘密信息实现安全服务。

安全协议举例

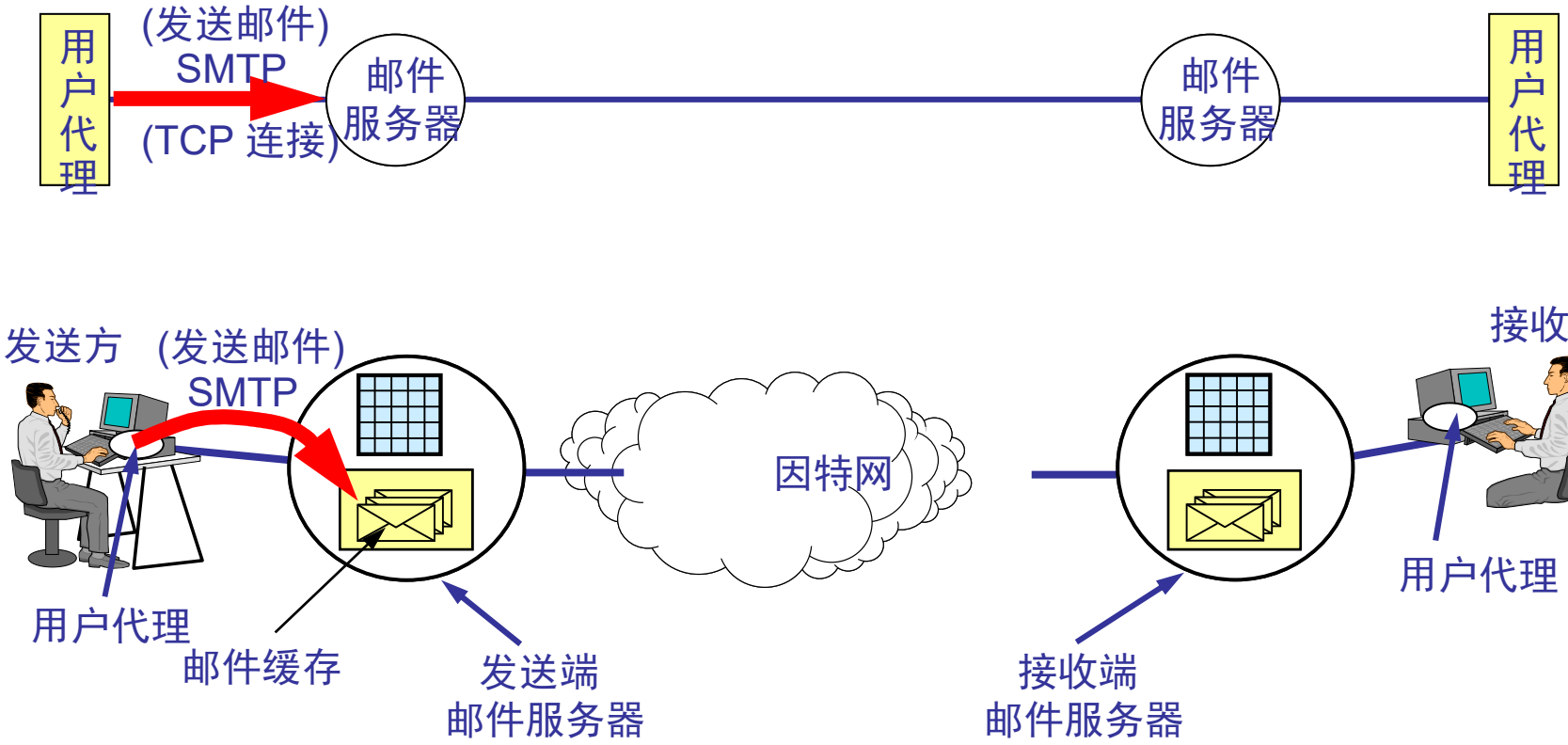
电子邮件的安全（PGP协议）

❖ 电子邮件的收取和发送过程(1)



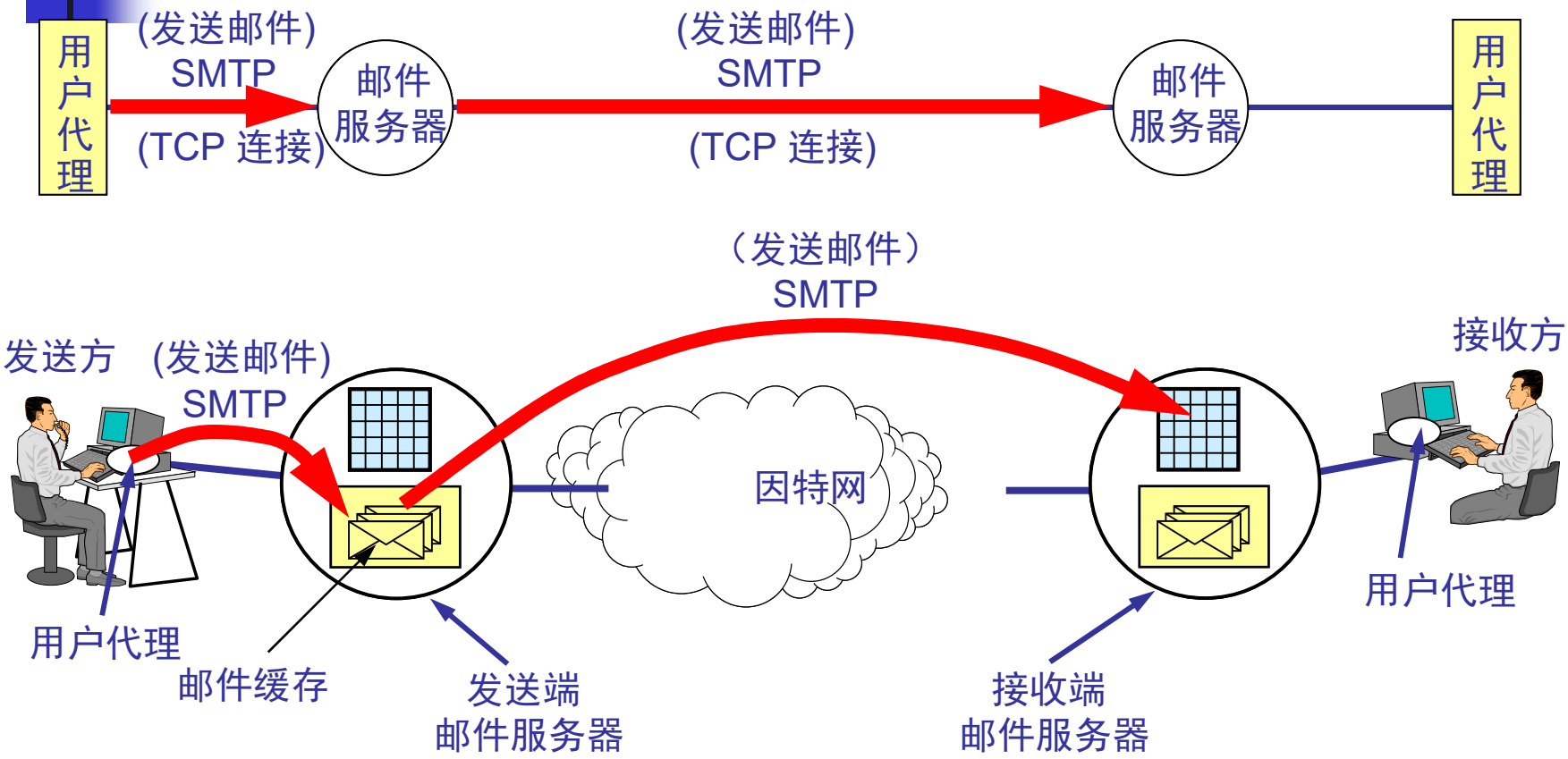
(1) 发信人调用用户代理来编辑要发送的邮件。用户代理用 SMTP 把邮件传送给发送端邮件服务器。

❖ 电子邮件的收取和发送过程(2)



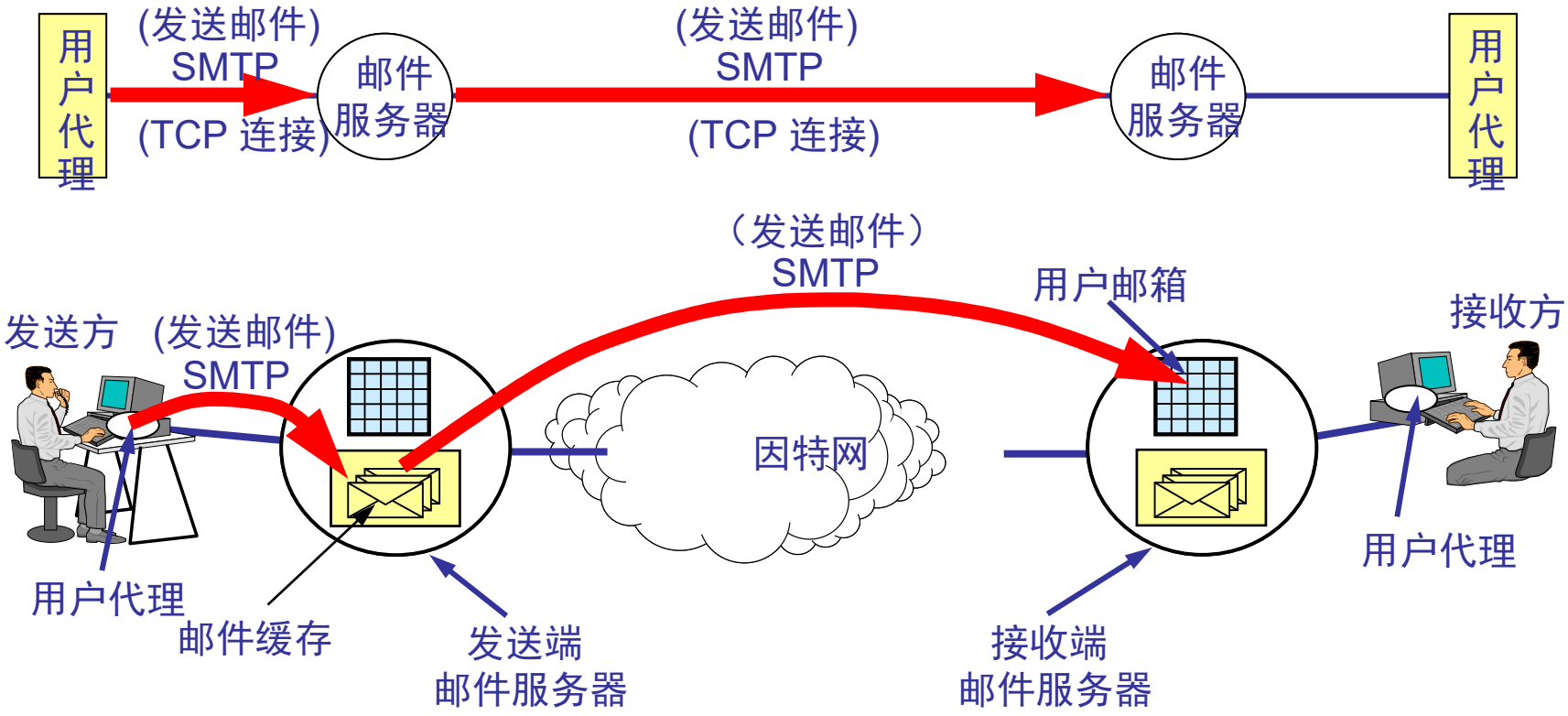
(2) 发送端邮件服务器将邮件放入 邮件缓存队列中，等待发送。

❖ 电子邮件的收取和发送过程(3)



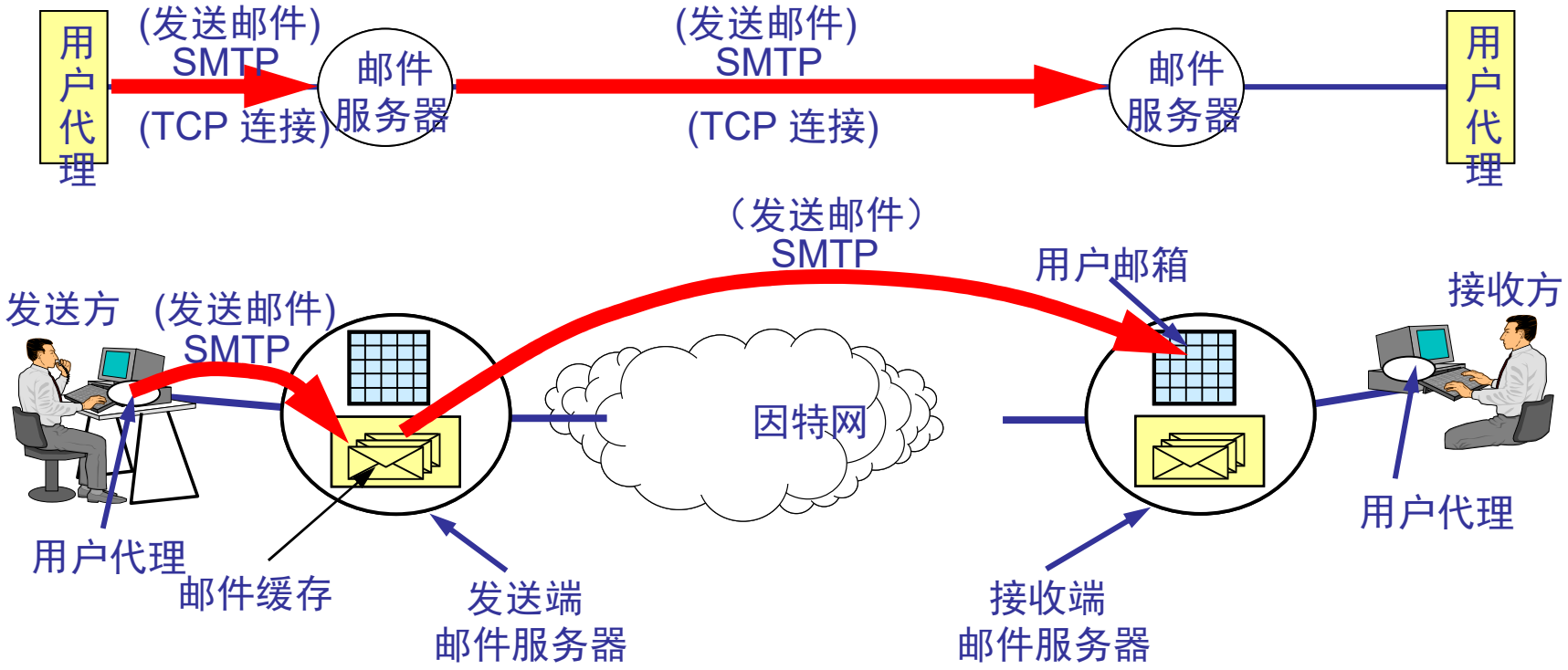
(3)运行在发送端邮件服务器的 SMTP 客户进程，发现在邮件缓存中有待发送的邮件，就向运行在接收端邮件服务器的 SMTP 服务器进程发起 TCP 连接的建立。

❖ 电子邮件的收取和发送过程(4)



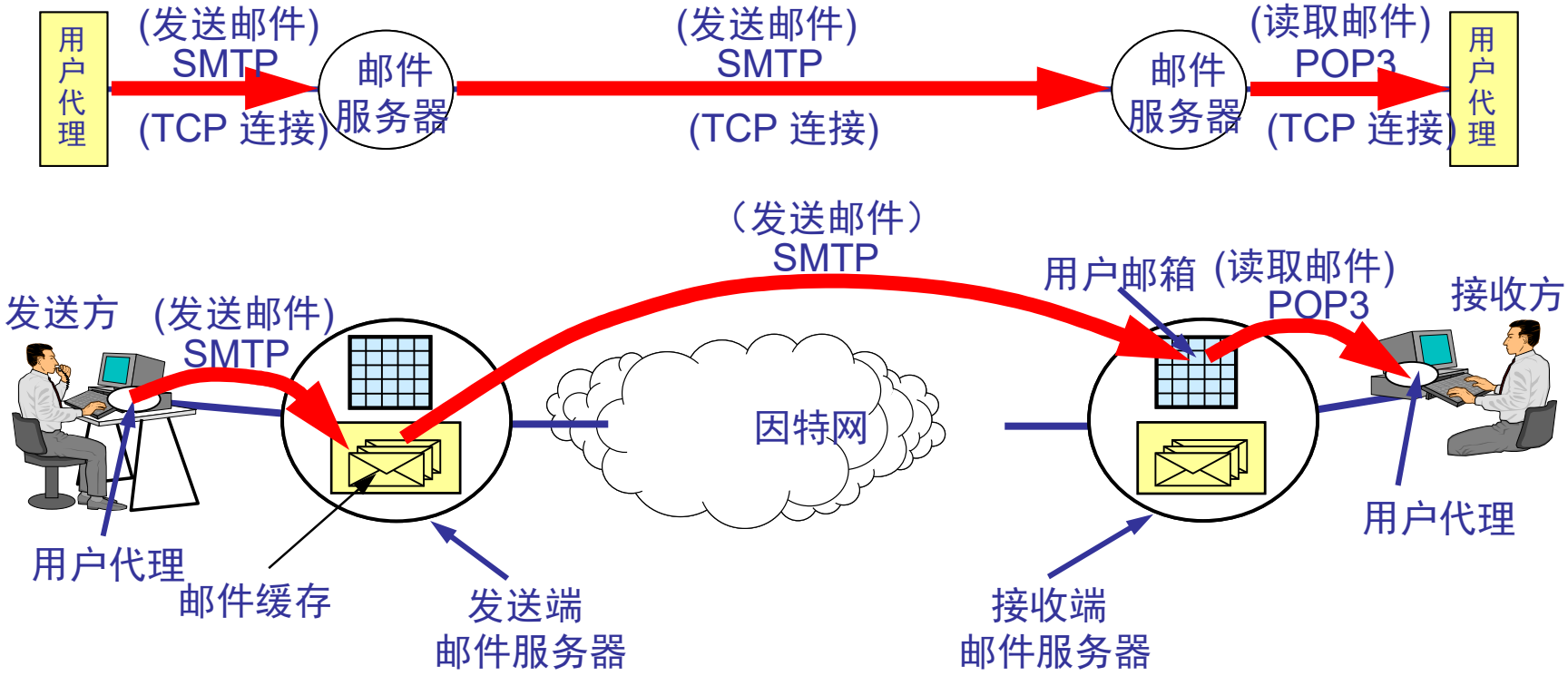
(4) TCP 连接建立后，SMTP 客户进程开始向远程的 SMTP 服务器进程发送邮件。当所有的待发送邮件发完了，SMTP 就关闭所建立的 TCP 连接。

电子邮件的收取和发送过程(5)



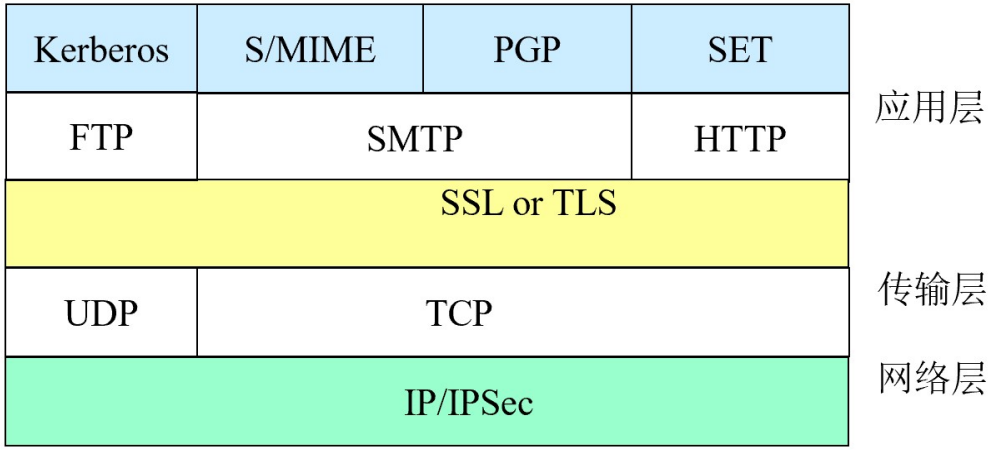
(5)运行在接收端邮件服务器中的 SMTP 服务器进程收到邮件后，将邮件放入收信人的用户邮箱中，等待收信人在方便时进行读取。

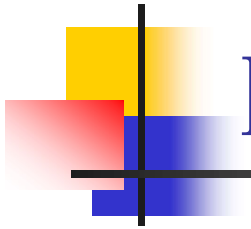
电子邮件的收取和发送过程(6)



(6)收信人在打算收信时，调用用户代理，使用 POP3（或 IMAP）协议将自己的邮件从接收端邮件服务器的用户邮箱中的取回（如果邮箱中有来信的话）。

- PGP - Pretty Good Privacy
  - 作者：Phil Zimmermann
  - 提供可用于电子邮件和文件存储应用的保密与鉴别服务。





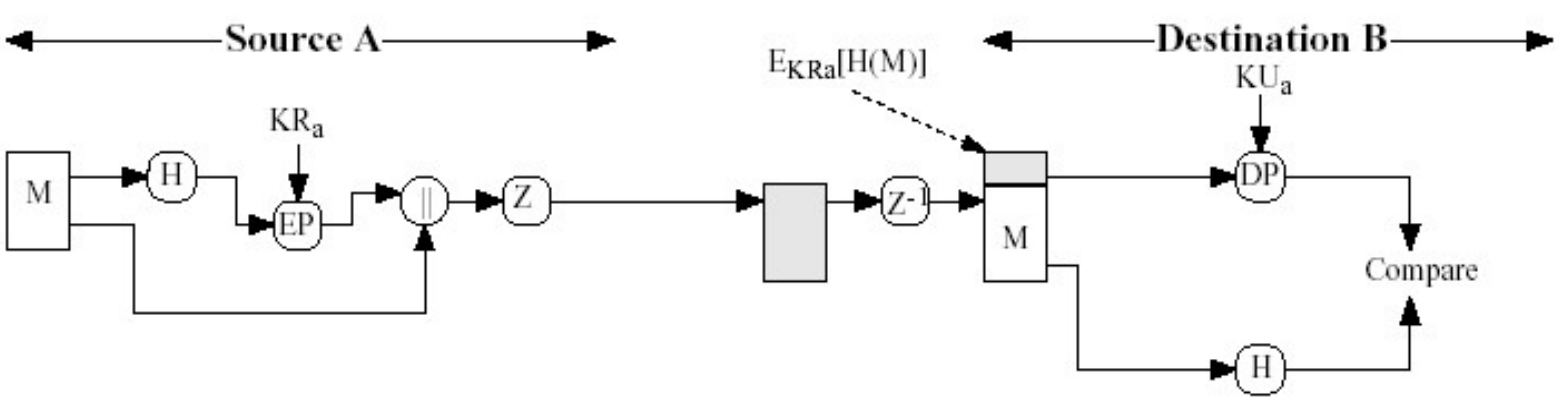
PGP安全业务

- 数字签名
  - DSS/SHA或RSA/SHA
- 完整性
  - RSA、MD5
- 消息加密
  - CAST-128或IDEA或3DES + Diffie-Hellman或RSA
- 数据压缩
  - ZIP
- 邮件兼容
  - Radix 64
- 数据分段

记号说明：

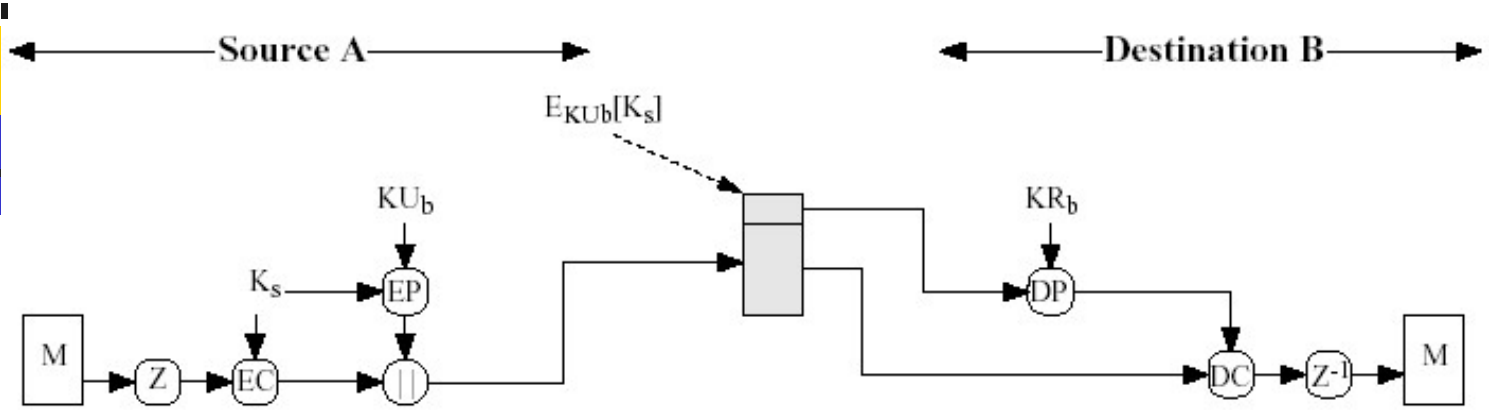
Ks	**: session key**
KRa	**: 用户A的私钥**
KUa	**: 用户A的公钥**
EP	**: 公钥加密**
DP	**: 公钥解密**
EC	**: 常规加密**
DC	**: 常规加密**
H	**: 散列函数**
‖	**: 连接**
Z	**: 用ZIP算法数据压缩**
R64	**: 用radix64转换到ASCII格式**

PGP——功能：身份鉴别



- 发送方
  - 产生消息M
  - 用SHA-1对M生成一个160位的散列码H
  - 用发送者的私钥对H加密，并与M连接
- 接收方
  - 用发送者的公钥解密并恢复散列码H
  - 对消息M生成一个新的散列码，与H比较。如果一致，则消息M被鉴别。

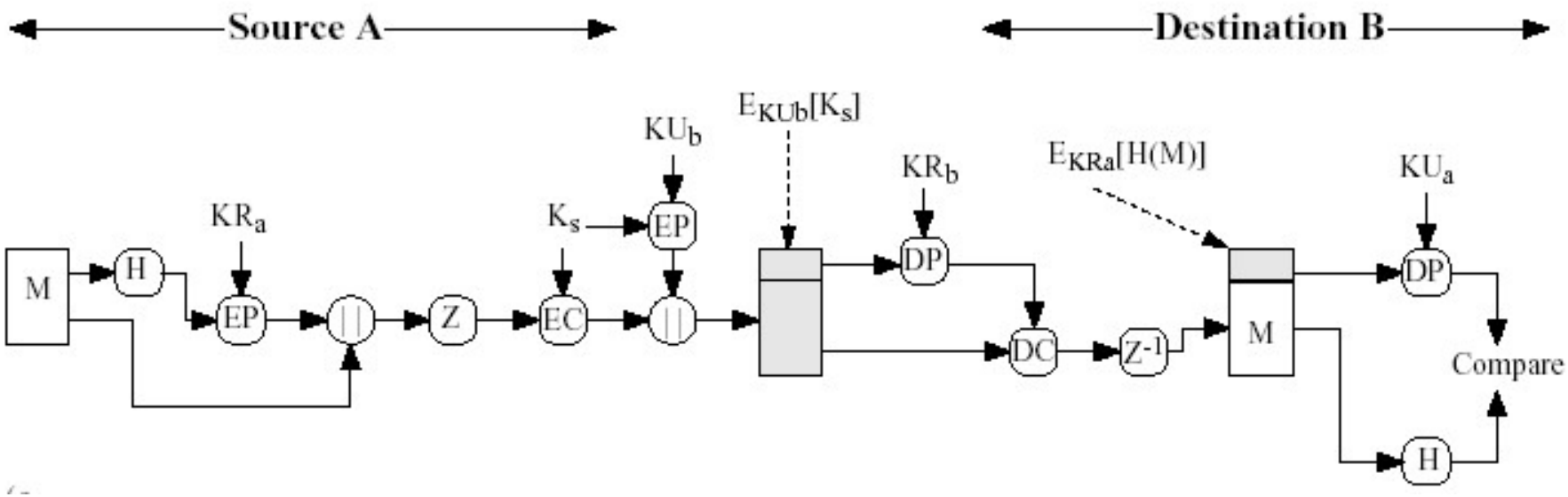
PGP —— 保密性

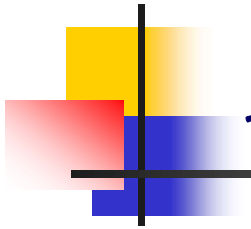


- 发送方
  - 生成消息M并为该消息生成一个随机数作为会话密钥。
  - 用会话密钥加密M
  - 用接收者的公钥加密会话密钥并与消息M结合
- 接收方
  - 用自己的私钥解密恢复会话密钥
  - 用会话密钥解密恢复消息M

保密与鉴别同时运用

- 发送者先用自己的私钥签名，然后用会话密钥加密，再用接收者的公钥加密会话密钥。





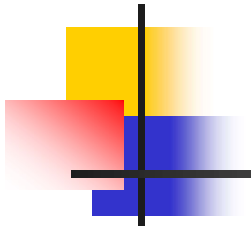
数据压缩

- 压缩的位置：发生在签名后、加密前。
- 在加密前压缩：压缩的报文更难分析，可以加强密码的安全性。

- 对邮件传输或存储都有节省空间的好处。

PGP的操作－电子邮件兼容性

- PGP将会遇到发送二进制数据的问题 (例如加密的结果)
- 但Email系统设计为仅支持可打印字符
- 因此PGP必须实现二进制流到ASCII字符的转换功能
- 采用基数-64转换算法
  - maps 3 bytes to 4 printable chars
  - also appends a CRC
- 如果内容太大，PGP也提供对消息分段的功能



分段与重组

- **Email**常常受限制于最大消息长度（一般限制在最大**50000**字节）
- 更长的消息要进行分段，每一段分别邮寄。
- **PGP**自动分段并在接收时自动恢复。
- 签名只需一次，在第一段中。

加密密钥和密钥环

PGP使用四种类型的密钥：一次性会话常规密钥，公钥，私钥，基于口令短语的常规密钥。

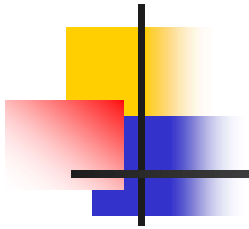
需求：

1、需要一种生成不可预知的会话密钥的手段

2、需要某种手段来标识具体的密钥。

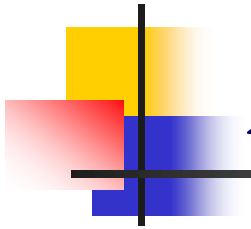
- 一个用户拥有多个公钥/私钥对。（更换，分组）

3、每个PGP实体需要维护一个文件保存其公钥私钥对，和一个文件保存通信对方的公钥。



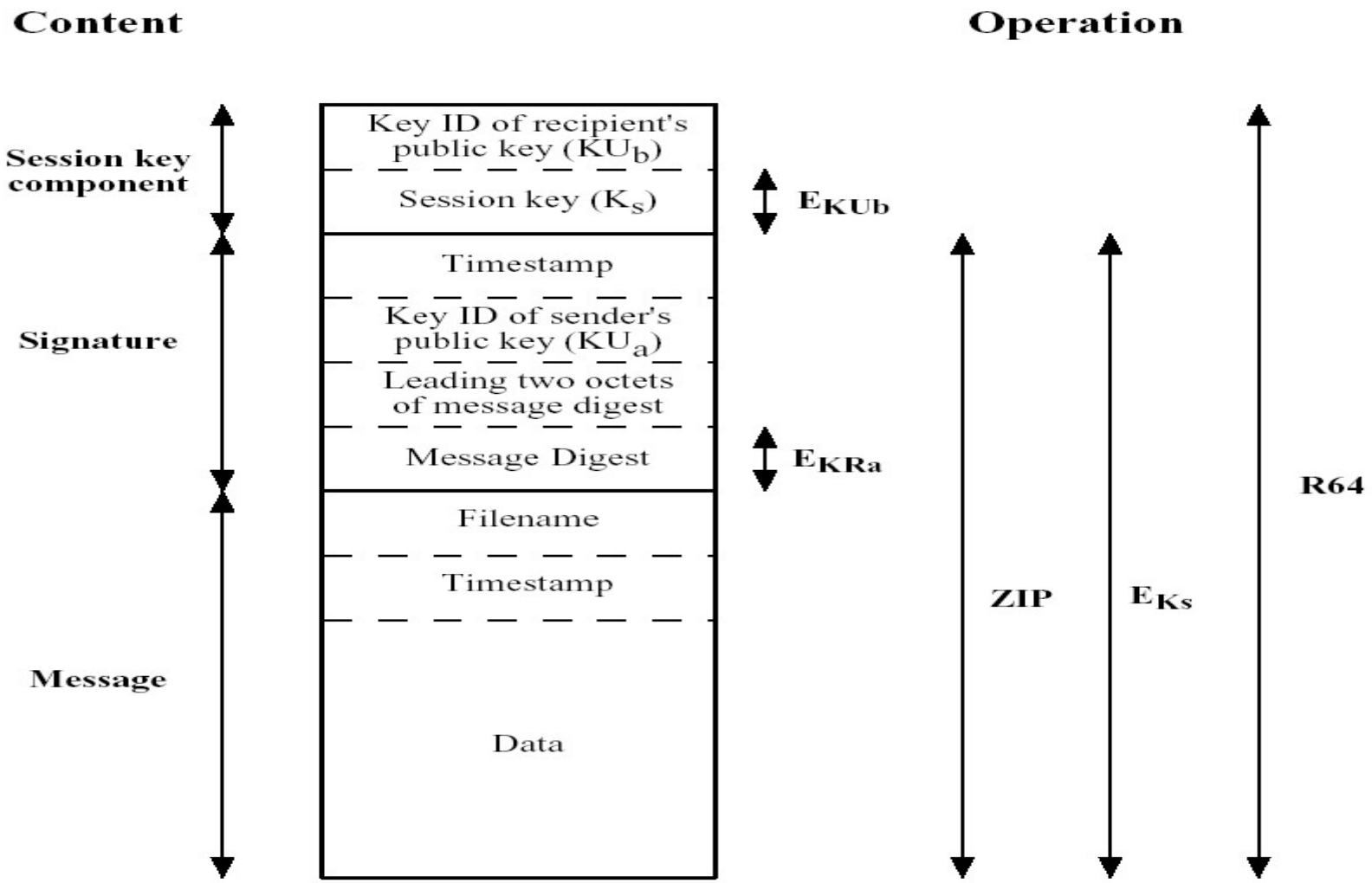
密钥标识符

- 一个用户有多个公钥/私钥对时，接收者如何知道发送者是用了哪个公钥来加密会话密钥？
  - 将公钥与消息一起传送。
  - 将一个标识符与一个公钥关联。对一个用户来说做到一一对应。
- 定义**KeyID** 包括**64**个有效位：**(KUa mod 264)**
- **KeyID**同样也需要**PGP**数字签名。



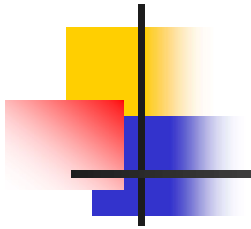
发送消息的格式

- 一个消息包含三部分成员：
  - 报文**message component**
  - 签名**signature (optional)**
  - 会话密钥**session key component (optional)**



Notation:

E_{KUb} = encryption with user b's private key
E_{KRa} = encryption with user a's public key
E_{Ks} = encryption with session key
ZIP = Zip compression function
R64 = Radix-64 conversion function



密钥环

- **PGP**在每一个节点上提供一对数据结构：
  - 公钥环：包括该用户知道的所有其它**PGP**用户的公钥，由密钥标识索引。
  - 私钥环：包括该用户的公钥私钥对，由密钥标识索引，且被杂凑了的用户口令加密。

Private-Key Ring

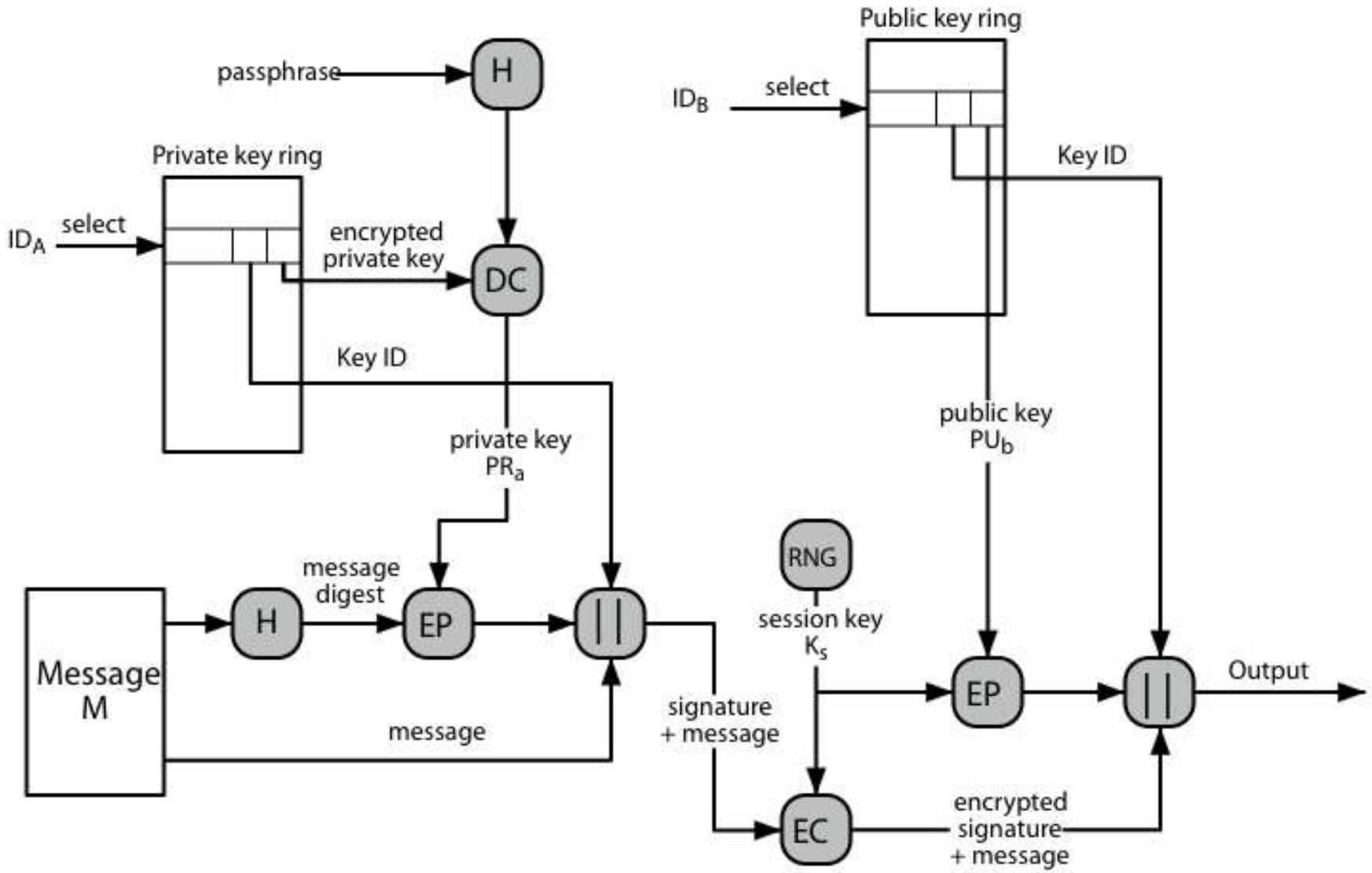
Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	$E_{H(P_i)}[KR_i]$	User *i*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Public-Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	$trust\_flag_i$	$User_i$	$trust\_flag_i$		
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•

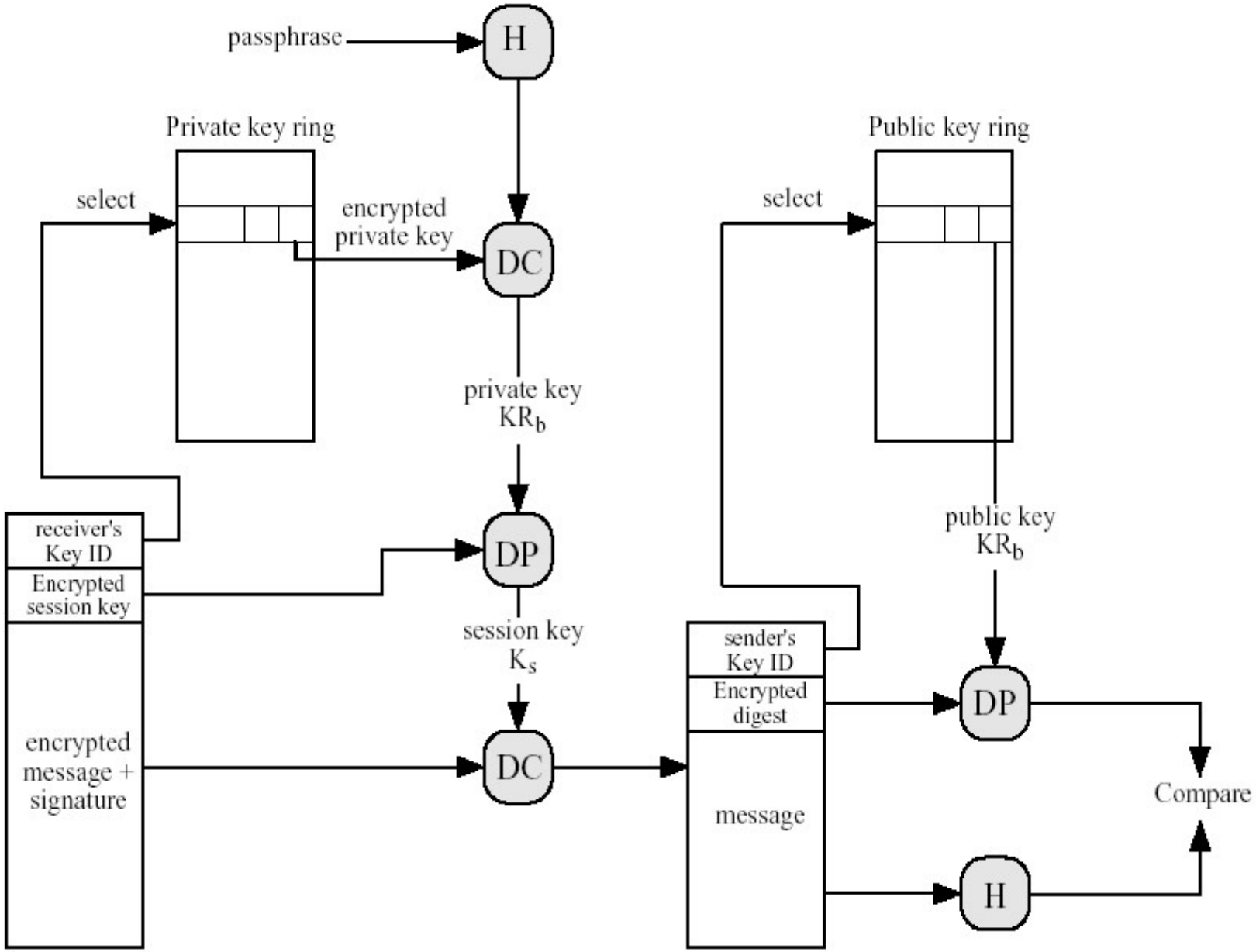
* = field used to index table

Figure 12.4 General Structure of Private- and Public-Key Rings.


passphrase
H
Private key ring
ID_A
select
encrypted
private key
DC
Key ID
private key
PR_a
message
digest
H
EP
||
Message
M
message
signature
+ message
RNG
session key
K_s
EC
Public key ring
ID_B
select
Key ID
public key
PU_b
EP
||
Output
encrypted
signature
+ message

PGP—发送方处理消息的过程

- 签名：
  - 从私钥环中得到私钥，利用**userid**作为索引
  - **PGP**提示输入口令短语，恢复私钥
  - 构造签名部分
- 加密：
  - **PGP**产生一个会话密钥，并加密消息
  - **PGP**用接收者**userid**从公钥环中获取其公钥
  - 构造消息的会话密钥部分


passphrase
H
Private key ring
select
encrypted
private key
DC
private key
KR_b
receiver's
Key ID
Encrypted
session key
DP
session key
K_s
encrypted
message +
signature
DC
Public key ring
select
public key
KR_b
sender's
Key ID
Encrypted
digest
DP
message
H
Compare

PGP报文的接收

PGP—接收方处理消息的过程

- 解密消息
  - PGP用消息的会话密钥部分中的KeyID作为索引，从私钥环中获取私钥
  - PGP提示输入口令短语，恢复私钥
  - PGP恢复会话密钥，并解密消息
- 验证消息
  - PGP用消息的签名部分中的KeyID作为索引，从公钥环中获取发送者的公钥
  - PGP恢复被传输过来的消息摘要
  - PGP对于接收到的消息作摘要，并与上一步的结果作比较

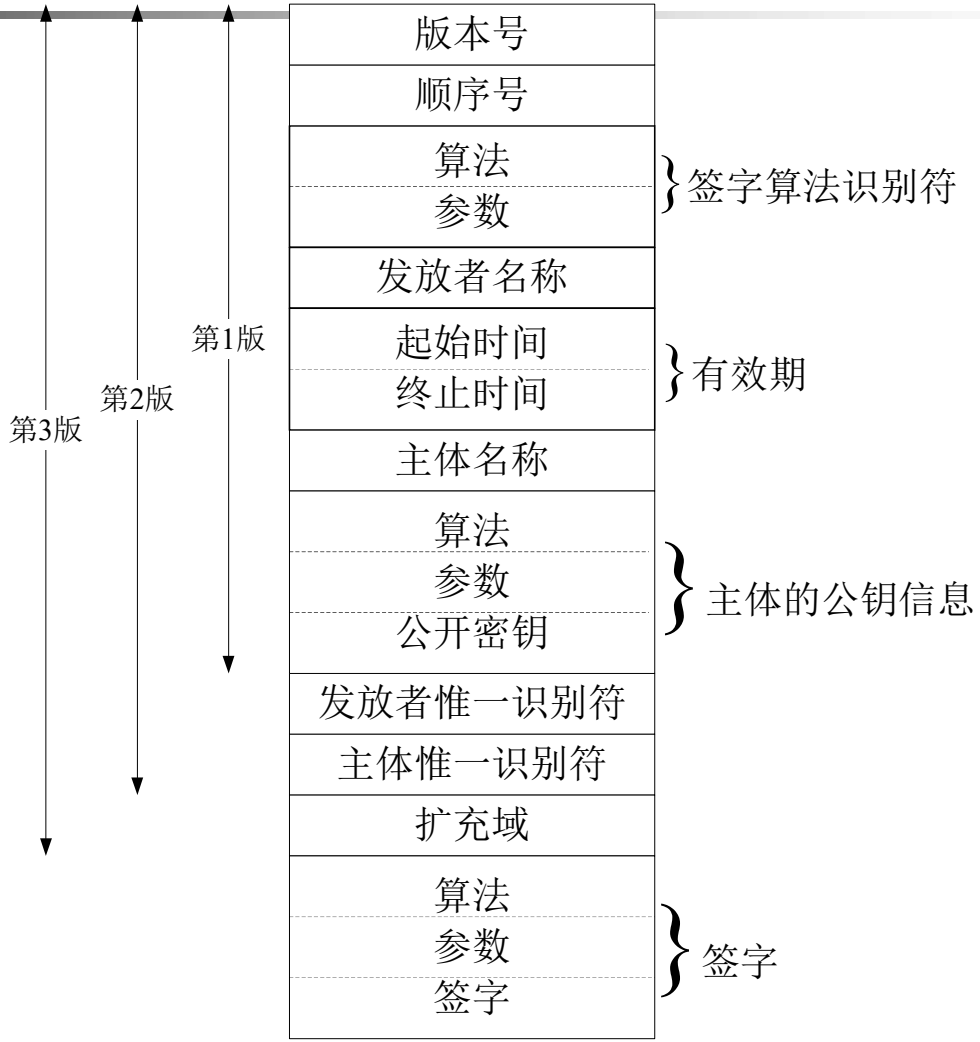
PGP 密钥管理

- 依赖于CA
- 在PGP中每个用户都是自己的 CA
  - 可以为所有知道的用户直接进行签名
- 形成了一个“信任网”
  - 相信密钥已经被签名
  - 如果对他们来说构成了一个签名信任链，则可以相信该密钥
- 密钥环引入了“信任指示”
- 用户可以撤销他们自己的密钥

A获得B公钥的方法

- 物理上从B获得密钥
- 利用电话验证密钥
- 从共同信任的个体D处获得B的公钥
- 从信任的认证机构中获取B的公钥

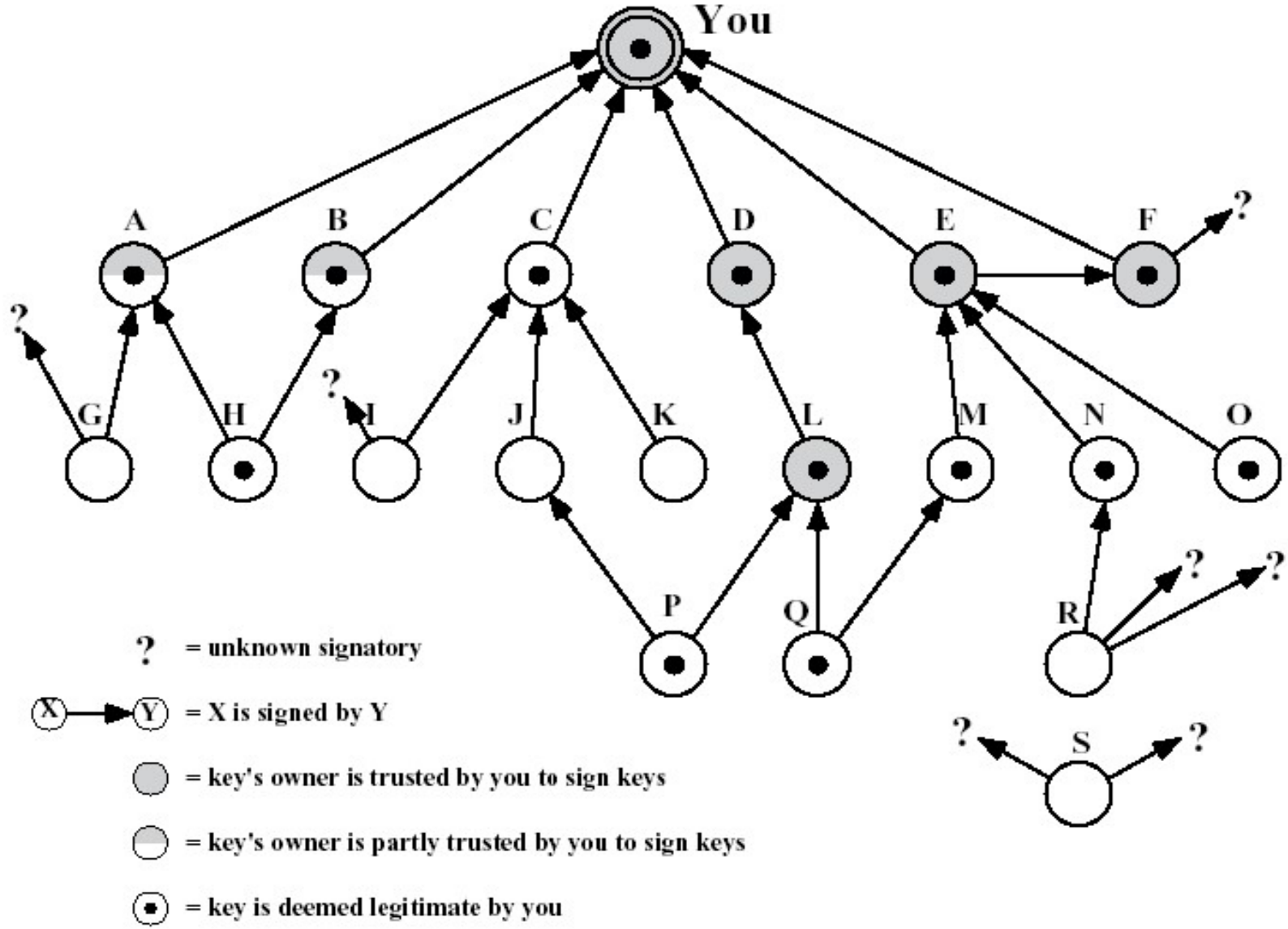
公钥：X.509证书格式

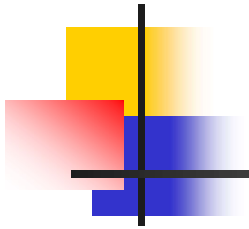

版本号
顺序号
算法
参数
签字算法识别符
发放者名称
起始时间
终止时间
有效期
主体名称
算法
参数
公开密钥
主体的公钥信息
发放者惟一识别符
主体惟一识别符
扩充域
算法
参数
签字
签字
第1版
第2版
第3版

PGP信任管理

- 第一种是集中式信任，比如CA结构，大家都信任CA，于是，Peter是不是Peter，是根据CA是否认识这个Peter，例如数字证书认证。
- 第二种方式没有CA，只信任自己和朋友。每个人都是在一个朋友圈子中，朋友圈子以外的人，则需要一个中间人作为介绍人，这就是现实的生活。
- **PGP基于第二种方式**
- 用户以各自为中心,相互认证公钥,相互签名公钥证书。 这些签名使得用户的公钥彼此相连,形成自然的网状结构,也就是所谓的信任网 **Web Of Trust** 。

PGP —— 信任模型示例


You
A
B
C
D
E
F
?
G
H
I
J
K
L
M
N
O
P
Q
R
S
? = unknown signatory
X
Y
= X is signed by Y
= key's owner is trusted by you to sign keys
= key's owner is partly trusted by you to sign keys
= key is deemed legitimate by you



- 信任值表示可以是定性的(离散的表示)，也可以是定量的(连续的表示)。
- 可以将信任值离散地表示为

untrust, uncertainty, low, medium, high

Trust? Authentication?