数字签名和鉴别协议

Digital Signatures & Authentication Protocols

message authentication

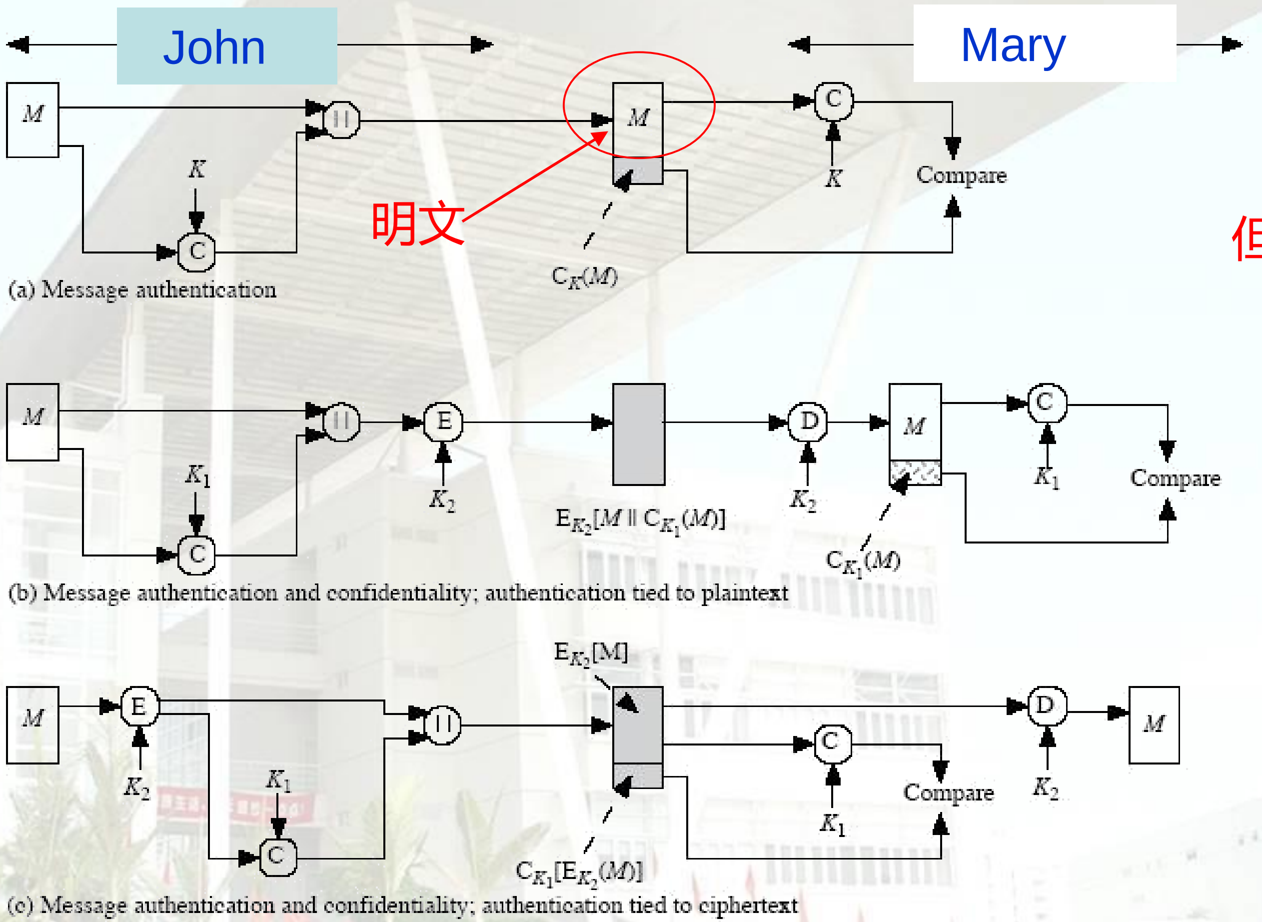


Figure 11.4 Basic Uses of Message Authentication Code (MAC)

防第三方攻击

但不能防双方抵赖

1.伪造M′

2.没发M′′

doesn't address issues of lack of trust

报文鉴别的局限性

- 用于保护通信双方免受第三方攻击
- 无法防止通信双方的相互攻击
  - ✓ 接收方伪造报文
  - ✓ 发送方否认已发送的报文

  诚信问题: 电子银行,股票交易,电子商务, 电子病历
- 引入数字签名

Digital Signatures

To provide the ability:

- verify author, date & time of signature
- authenticate message contents
- be verified by third parties to resolve disputes (解决纠纷)

有手写签名的同等功效!!!

必须能够验证作者及其签名的日期时间

必须能够认证签名时刻的内容

签名必须能够由第三方验证，以解决争议

数字签名模仿传统签名的要点

- 传统签名的基本特点
  - ✓ 能与被签的文件在物理上不可分割
  - ✓ 签名者不能否认自己的签名
  - ✓ 签名”不能”被伪造
  - ✓ “容易”被验证
- 数字签名是传统签名的数字化模仿形式
  - ➢ 能与所签文件“绑定”
  - ➢ 签名者不能否认自己的签名
  - ➢ 签名不能被伪造
  - ➢ 容易被自动验证

数字签名的设计要求

- 签名必须是依赖于被签名信息
- 签名必须使用对发送者是唯一的信息，以防止双方的伪造与否认
- 必须相对容易生成
- 必须相对容易识别和验证
- 伪造该数字签名在计算上具有不可行性，

  对一个已有的数字签名构造新的消息，

  对一个给定消息伪造数字签名
- 在存储器中可保存数字签名副本

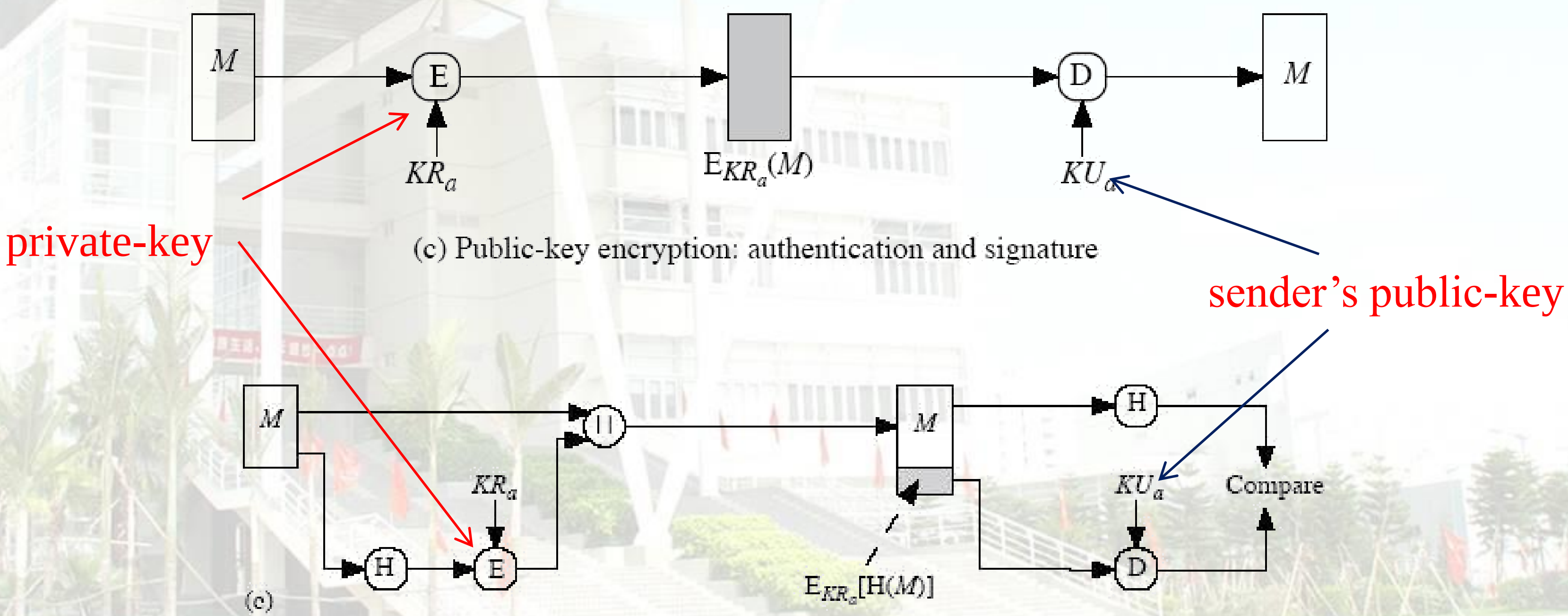
数字签名分类

签名方式

- 直接数字签名direct digital signature
- 仲裁数字签名arbitrated digital signature

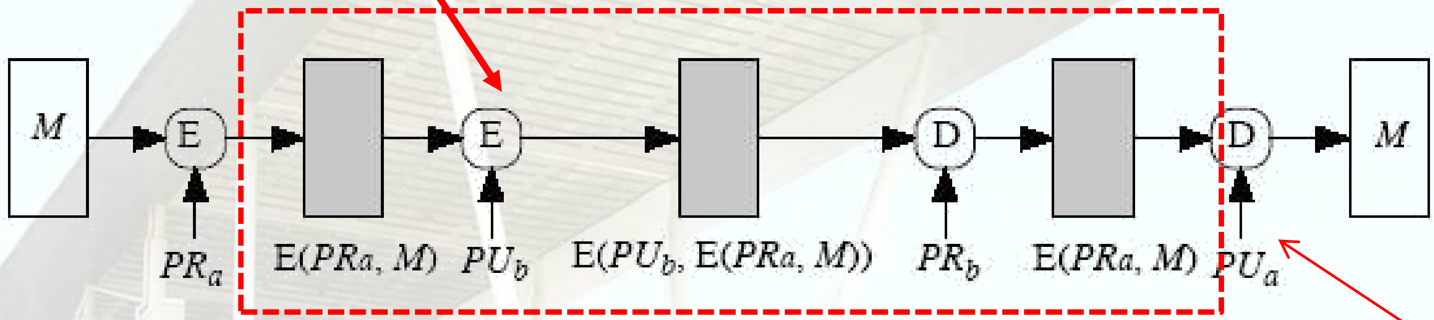
Direct Digital Signatures-DDS

- involve only sender & receiver
- assumed receiver has sender's public-key
- digital signature made by sender signing entire message or hash with private-key



Direct Digital Signatures (2)

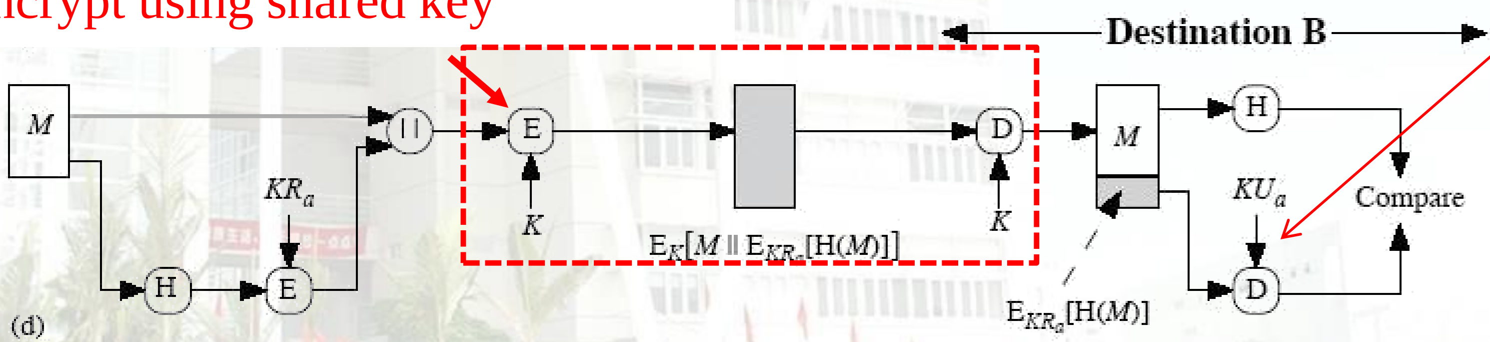
- encrypt using receivers public-key



(d) Public-key encryption: confidentiality, authentication, and signature

sender's public-key

- encrypt using shared key



- Order is very important
- Important : sign first then encrypt message
- Security: depends on sender's private-key

Direct Digital Signatures (3)

security depends on sender's private-key

Problems ?!

Direct Digital Signatures (4)

验证模式依赖于发送方的私有密钥

- 发送方抵赖发送某一消息，声称其私有密钥丢失或被窃，而他人伪造了他的签名 “I lost my private-key !!!”

例：改进的方式: add time-stamp

被签名的信息包含一个时间戳（日期与时间）

要求将已暴露的密钥报告给授权中心

- 发送方私有密钥确实在时间T被窃取

攻击方可伪造其签名早于或等于时间T的时间戳

Arbitrated Digital Signatures 仲裁数字签名

- involves use of arbiter A -> third party

  引入仲裁者—公证人

- Send any signed message to “A” first

  签名消息首先送到仲裁者“A”

  - validates any signed message “A”验证其来源和内容
  - dated and sent to recipient “A”加上日期一起发给接收方

- requires suitable level of trust in arbiter
- be implemented with either private or public-key algorithms
- arbiter may or may not see message

  所有的参与者必须相信仲裁机制

Table 13.1 Arbitrated Digital Signature Techniques

(a) Conventional Encryption, Arbiter Sees Message
(1) X → A: $M \| E_{K_{xa}}[ID_X \| H(M)]$
(2) A → Y: $E_{K_{ay}}[ID_X \| M \| E_{K_{xa}}[ID_X \| H(M)] \| T]$
(b) Conventional Encryption, Arbiter Does Not See Message
(1) X → A: $ID_X \| E_{K_{xy}}[M] \| E_{K_{xa}}[ID_X \| H(E_{K_{xy}}[M])]$
(2) A → Y: $E_{K_{ay}}[ID_X \| E_{K_{xy}}[M] \| E_{K_{xa}}[ID_X \| H(E_{K_{xy}}[M])] \| T]$
(c) Public-Key Encryption, Arbiter Does Not See Message
(1) X → A: $ID_X \| E_{KR_x}[ID_X \| E_{KU_y}(E_{KR_x}[M])]$
(2) A → Y: $E_{KR_a}[ID_X \| E_{KU_y}[E_{KR_x}[M]] \| T]$

Notation:

X = sender
Y = recipient
A = Arbiter

M = message
T = timestamp

仲裁数字签名－单密钥加密方式1

计算消息M的hash值

(1) $X \rightarrow A$： $M \| E_{K_{xa}}[ID_x \| \boxed{H(M)}]$

(2) $A \rightarrow Y$： $E_{K_{ay}}[ID_x \| M \| E_{K_{xa}}[ID_x \| H(M)] \| T]$

数字签名

X与A之间共享密钥K_{xa}，Y与A之间共享密钥K_{ay}；

X：准备消息M **计算M的hash值**H(M)，用X的标识符ID_x **及hash值形成**签名，并将消息及签名经K_{xa}加密后发送给A；

A：解密签名，用H(M)验证消息M，然后将ID_x，M，签名，和时间戳一起经K_{ay}加密后发送给Y；

Y：解密A发来的信息，并可将M和签名保存起来。

A可以看到X给Y的所有信息

仲裁数字签名－单密钥加密方式1

在这种模式下Y不能直接验证X的签名，Y认为A的消息正确，只因为它来自A。因此，双方都需要高度相信A：

- X必须信任A没有暴露K_{xa}，并且没有生成错误的签名

$$E_{K_{xa}}[ID_x \| H(M)]$$

- Y必须信任A **仅当hash值正确并** 签名确实是X产生的情况下才发送的 $E_{K_{ay}}[ID_x \| M \| E_{K_{xa}}[ID_x \| H(M)] \| T]$
- 双方都必须信任A

只要A遵循上述要求，则X相信没有人可以伪造其签名；Y相信X不能否认其签名。

仲裁数字签名 - 单密钥加密方式2

(1) $X \rightarrow A$: $ID_x \parallel E_{K_{xy}}[M] \parallel E_{K_{xa}}[ID_x \parallel H(E_{K_{xy}}[M])]$

(2) $A \rightarrow Y$: $E_{K_{ay}}[ID_x \parallel E_{K_{xy}}[M] \parallel E_{K_{xa}}[ID_x \parallel H(E_{K_{xy}}[M])] \parallel T]$

在这种情况下，X与Y之间共享密钥K_{xy}，

X：将标识符ID_x ,密文 $E_{K_{xy}}[M]$，以及对ID_x和密文消息的 hash值,用K_{xa}加密后形成签名发送给A。

A：解密签名，用hash值 验证消息，这时A只能验证消息的密文而不能读取其内容。然后A将来自X的所有信息加上时间戳并用K_{ay}加密后发送给Y。

问题

A和发送方联手可以否认签名的信息；

A和接收方联手可以伪造发送方的签名；

方式1和2均存在

仲裁数字签名－双密钥加密方式3

(1) $X \to A$: $ID_X \| E_{KR_X}[ID_X \| E_{KU_Y}(E_{KR_X}[M])]$

(2) $A \to Y$: $E_{KR_a}[ID_X \| E_{KU_Y}[E_{KR_X}[M]] \| T]$

X：对消息**M**双重加密：首先用**X**的私有密钥**KRx**，然后用**Y**的公开密钥**KUy**。形成一个签名的、保密的消息。然后将该信息以及**X**的标识符一起用**KRx**签名后与**IDx** 一起发送给**A**。这种内部、双重加密的消息对**A**以及对除**Y**以外的其它人都是安全的。

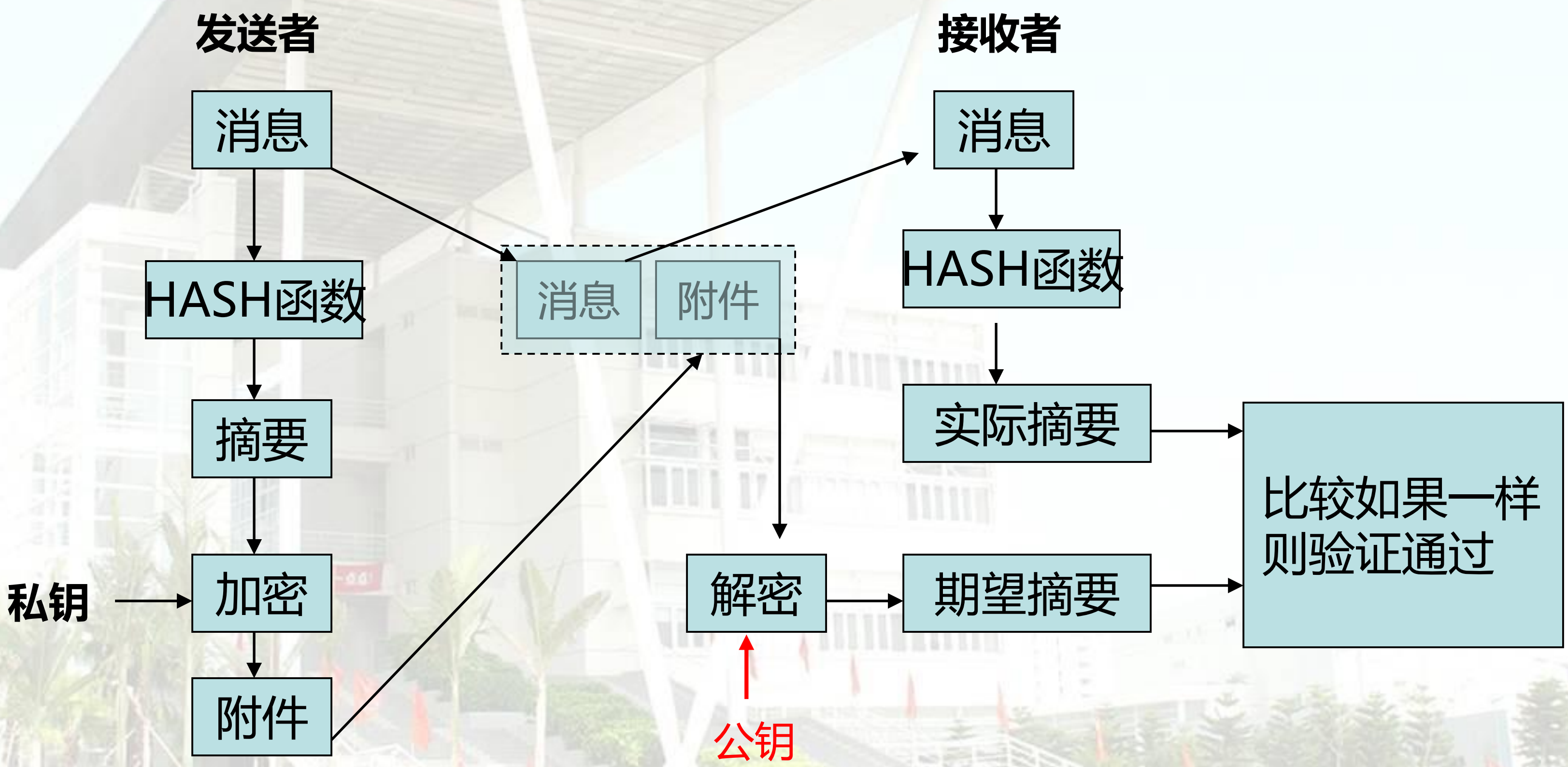
A：检查**X**的公开/私有密钥对是否仍然有效，是，则确认消息。并将包含**IDx**、双重加密的消息和时间戳构成的 消息用**KRa**签名后发送给**Y**。

仲裁数字签名 - 双密钥加密方式3

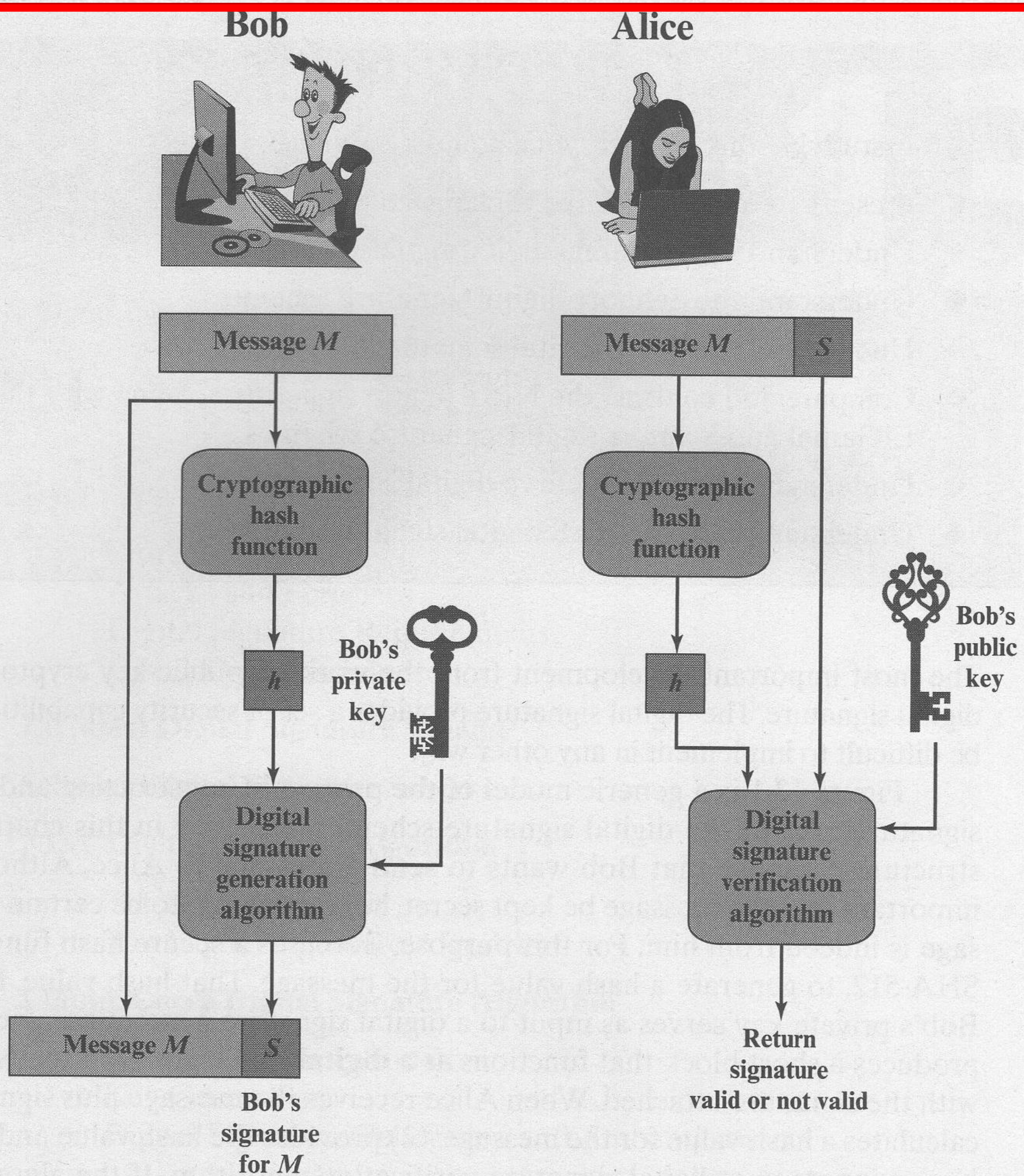
1、在通信之前各方之间无须共享任何信息，从而避免了联手作弊；
2、即使KRx 暴露，只要KRa 未暴露，不会有错误标定日期的消息被发送；
3、从X发送给Y的消息的内容对A和任何其他人是保密的。

常用数字签名技术- RSA法

- 用于证实消息的真实来源，并可以解决消息发送者和解收者之间的争端。以下是使用HASH函数的数字签名方案



Essential Elements of Digital Signature Process



(a) Bob signs a message

(b) Alice verifies the signature

Digital Signature