



信息安全导论

第四章 公钥密码与散列函数

黄 玮

中国传媒大学



温故

- 密码学简史
- 密码学基本概念
- 流密码
- 分组密码
- 对称密码应用



知新

- 公钥密码
- 散列函数
- 密码学相关应用



本章内容提要

- 公钥密码
- 散列函数
- 密码学相关应用



公开密钥算法

- 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
- 双密钥体制的公钥可以公开，因此也称公钥算法
- 公钥算法的出现，给密码的发展开辟了新的方向。公钥算法虽然已经历了20多年的发展，但仍具有强劲的发展势头，在鉴别系统和密钥交换等安全技术领域起着关键的作用



公开密钥算法的提出

- 公钥密码学是1976年由Diffie和Hellman在其“密码学新方向”一文中提出的，见文献：
—W.Diffie and M.E.Hellman, New Directions in Cryptography, IEEE Transaction on Information Theory, V.IT-22.No.6, Nov 1976, PP.644-654



公开密钥算法的提出

- RSA公钥算法是由Rivest, Shamir和Adleman在1978年提出来的
- 参见Communications of the ACM. Vol.21.No.2. Feb. 1978, PP.120-126
- 该算法的数学基础是初等数论中的Euler (欧拉)定理, 并建立在大整数因子的困难性之上



公开密钥算法的基本要求

- 加密与解密由不同的密钥完成
 - 加密: $X \rightarrow Y: Y = E_{KU}(X)$
 - 解密: $Y \rightarrow X: X = D_{KR}(Y) = D_{KR}(E_{KU}(X))$
- 产生公私密钥对、加密与解密的计算量可接受
- 知道加密算法，从加密密钥得到解密密钥在计算上是不可行的
 - 知道密文和公钥，恢复明文在计算上是不可行的
- 两个密钥中任何一个都可以作为加密而另一个用作解密（不是必须的）
 - 加解密次序可以交换

单向性



单向陷门函数

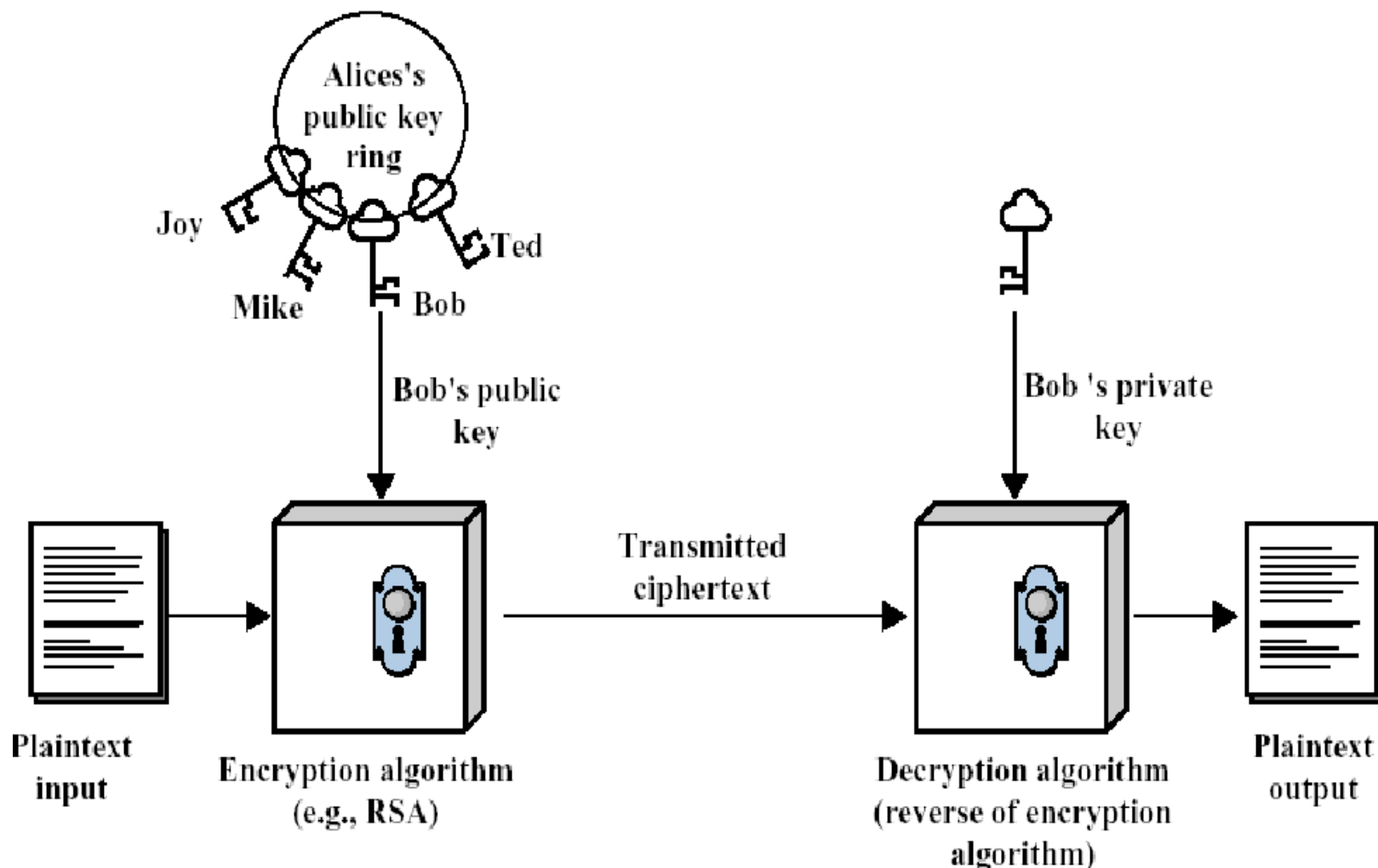
- 给定 x ，计算 $y=f(x)$ 是容易的
- 给定 y ，计算 $x=f^{-1}(y)$ 是困难的
——计算量大，耗时极长
- 存在 δ ，已知 δ 时，对给定的任何 y ，若相应的 x 存在，则计算 x 使 $x=f^{-1}(y)$ 是容易的
—— f 公开，相当于公钥
—— δ 保密，相当于私钥

单向性

陷门性



基于公开密钥的加密过程





用公钥密码实现保密

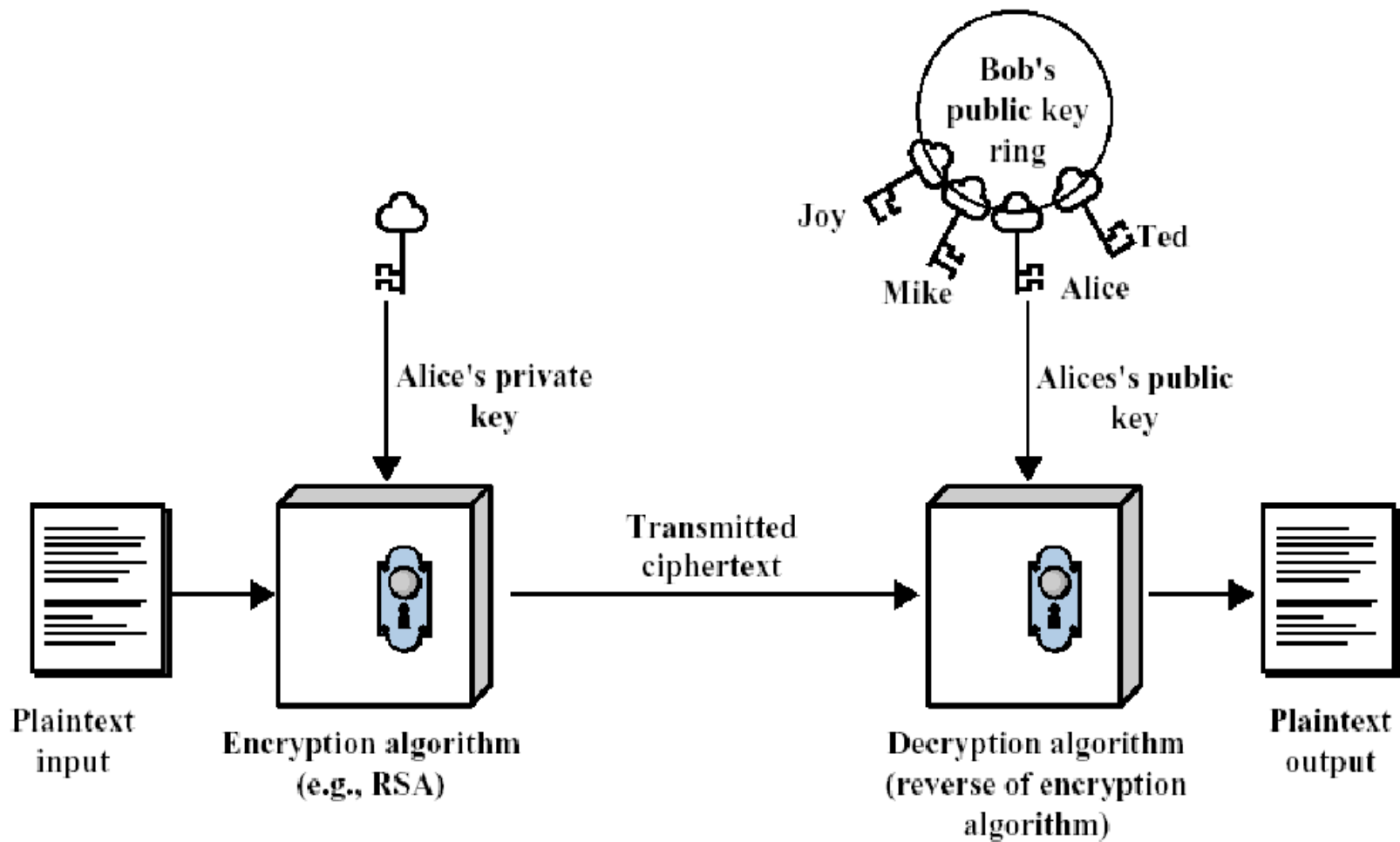
- 用户拥有自己的密钥对(KU,KR)
- 公钥 KU公开，私钥KR保密

$$A \rightarrow B: Y = E_{KU_b}(X)$$

$$B: D_{KR_b}(Y) = D_{KR_b}(E_{KU_b}(X)) = X$$



基于公开密钥的鉴别过程





用公钥密码实现鉴别

- 条件：两个密钥中任何一个都可以用作加密而另外一个用作解密
- 鉴别

$$A \rightarrow ALL: Y = E_{KR_a}(X)$$

$$ALL: D_{KU_a}(Y) = D_{KU_b}(E_{KR_a}(X)) = X$$

- 鉴别 + 保密

$$A \rightarrow B: Z = E_{KU_b}(E_{KR_a}(X))$$

$$B: D_{KU_a}(D_{KR_b}(Z)) = X$$



公开密钥算法

- 公钥算法的种类很多，具有代表性的三种密码：
- 基于整数分解难题 (IFP) 的算法体制
- 基于离散对数难题 (DLP) 算法体制
- 基于椭圆曲线离散对数难题 (ECDLP) 的算法体制

教材上有RSA算法和椭圆曲线密码算法，讲授略



本章内容提要

- 公钥密码
- 散列函数
- 密码学相关应用



散列函数的定义

- 散列函数:

$$\text{hash} = H(M)$$

- M: 变长 (消息) 报文
- $H(M)$: 定长的散列值
- 主要用于为文件、报文或其它分组数据产生指纹: hash



散列函数的要求

- H能用于任意大小输入的分组
- H能产生定长输出
- 对任何给定的 x ， $H(x)$ 要相对易于计算，使得硬件和软件实现成为实际可能
- 对任何给定的码 h ，寻找 x 使得 $H(x)=h$ 在计算上是不可行的，即单向性
- 对任何给定的分组 x ，寻找不等于 x 的 y ，使得 $H(x)=H(y)$ 在计算上是不可行的，即弱碰撞性
- 对任何 $x \neq y$ ，使得 $H(x)=H(y)$ 在计算上是不可行的，即强碰撞性
- 输入改变任何一个比特或多个比特， $H(x)$ 会相应变化尽可能多个比特位，即雪崩效应（可选要求）



Hash vs MAC

- MAC提供的是一种消息**可信**鉴别方法
—完整性、真实性
- Hash提供的是一种消息**完整**性保证方法
- Hash是一种直接产生MAC的方法
- MAC的产生方法不局限于Hash



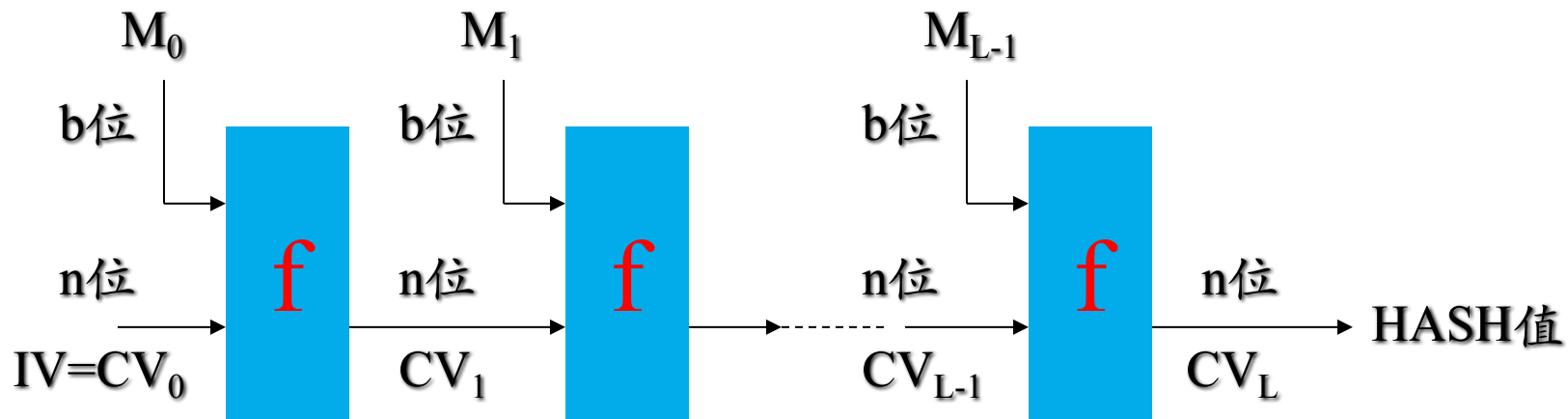
Hash函数通用结构

- 由Ron Rivest于1990年提出MD4
- 几乎被所有hash函数在设计时所借鉴
- 具体做法
 - 把原始消息M分成一些固定长度的块 Y_i
 - 最后一块padding并使其包含消息M的长度值
 - 设定初始值 CV_0
 - 压缩函数 f , $CV_i = f(CV_{i-1}, Y_{i-1})$
 - 最后一个 CV_i 为hash值



安全Hash函数的一般结构

- Merkle提出了安全Hash函数主处理的一般结构
- 对数据压缩，产生Hash码

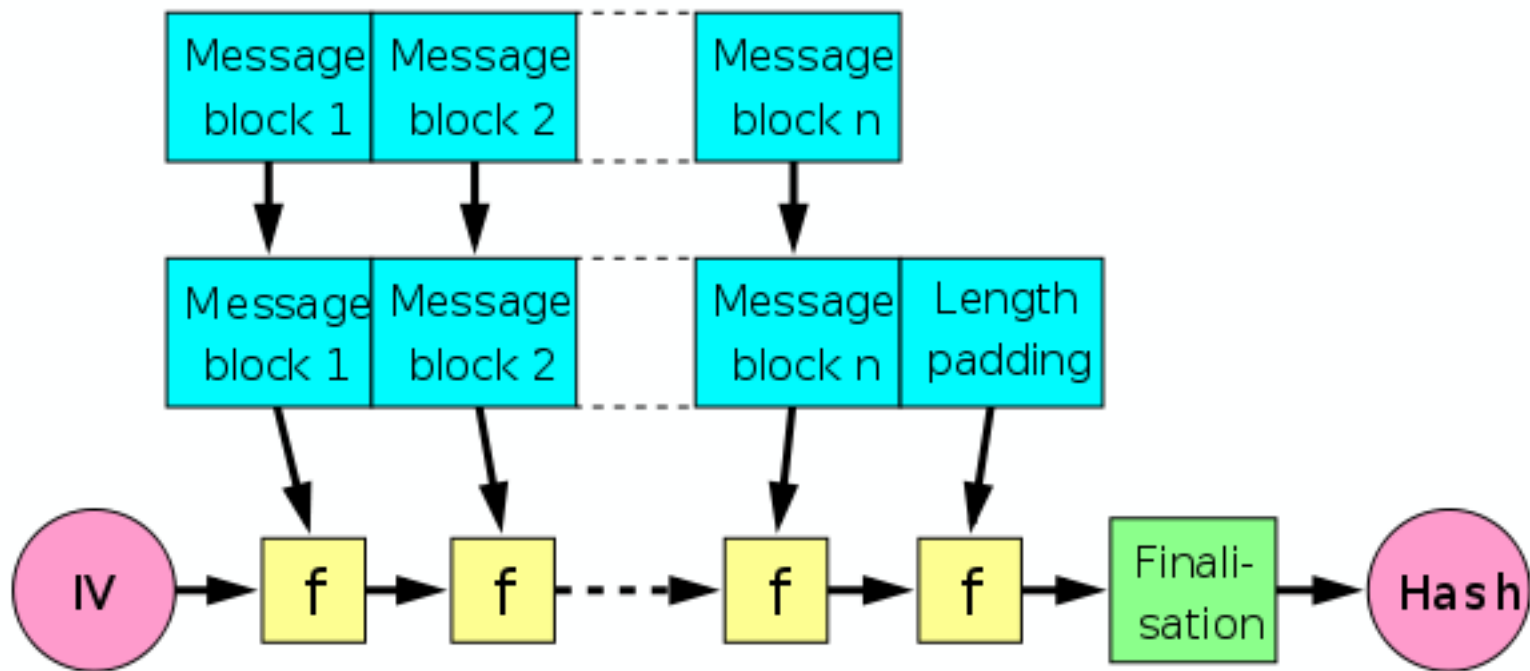


b 位分组， f 为压缩函数， L 轮链接迭代， n 位输出。



Merkle Damgård数据结构

- 简称MD数据结构





Message-Digest Algorithm 5

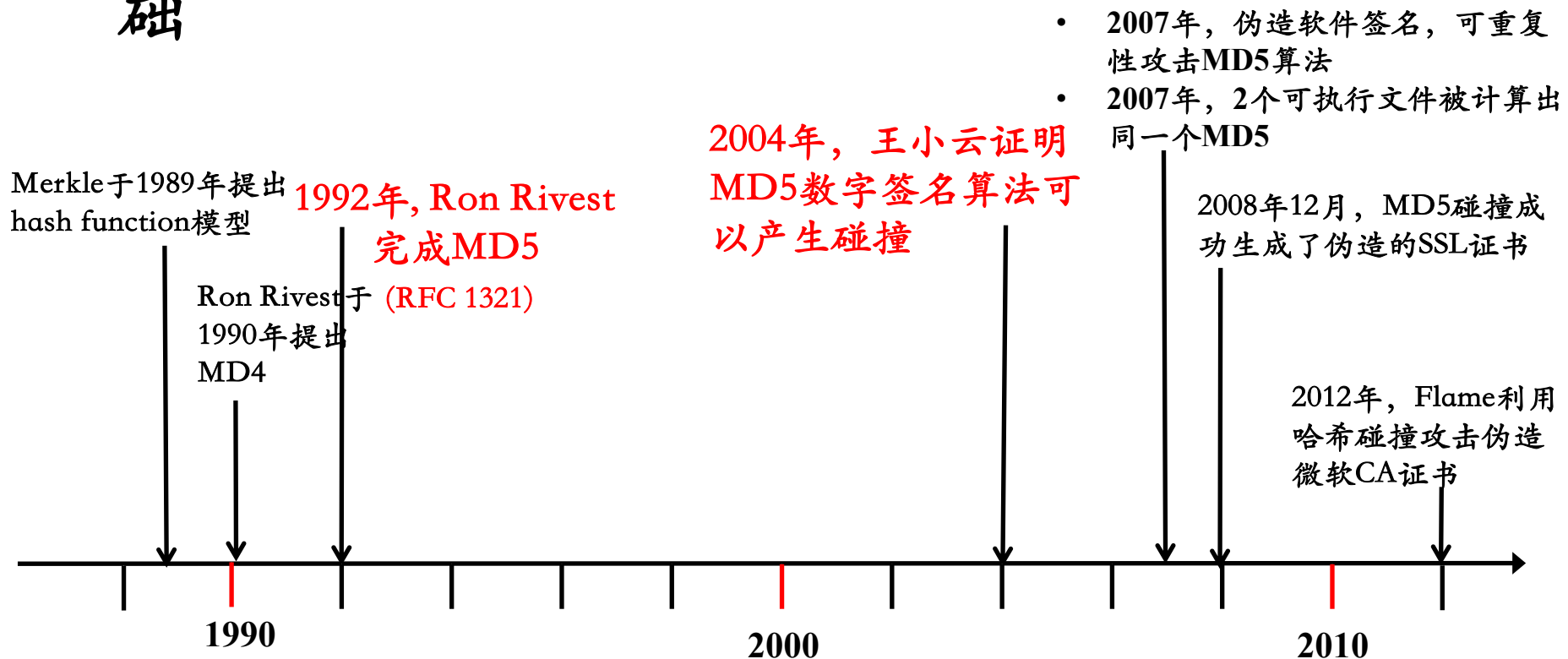
MD5

中國傳媒大學



MD5背景

- 在最近数年之前，MD5是最主要的hash算法
- 现行美国标准SHA-1以MD5的前身MD4为基础



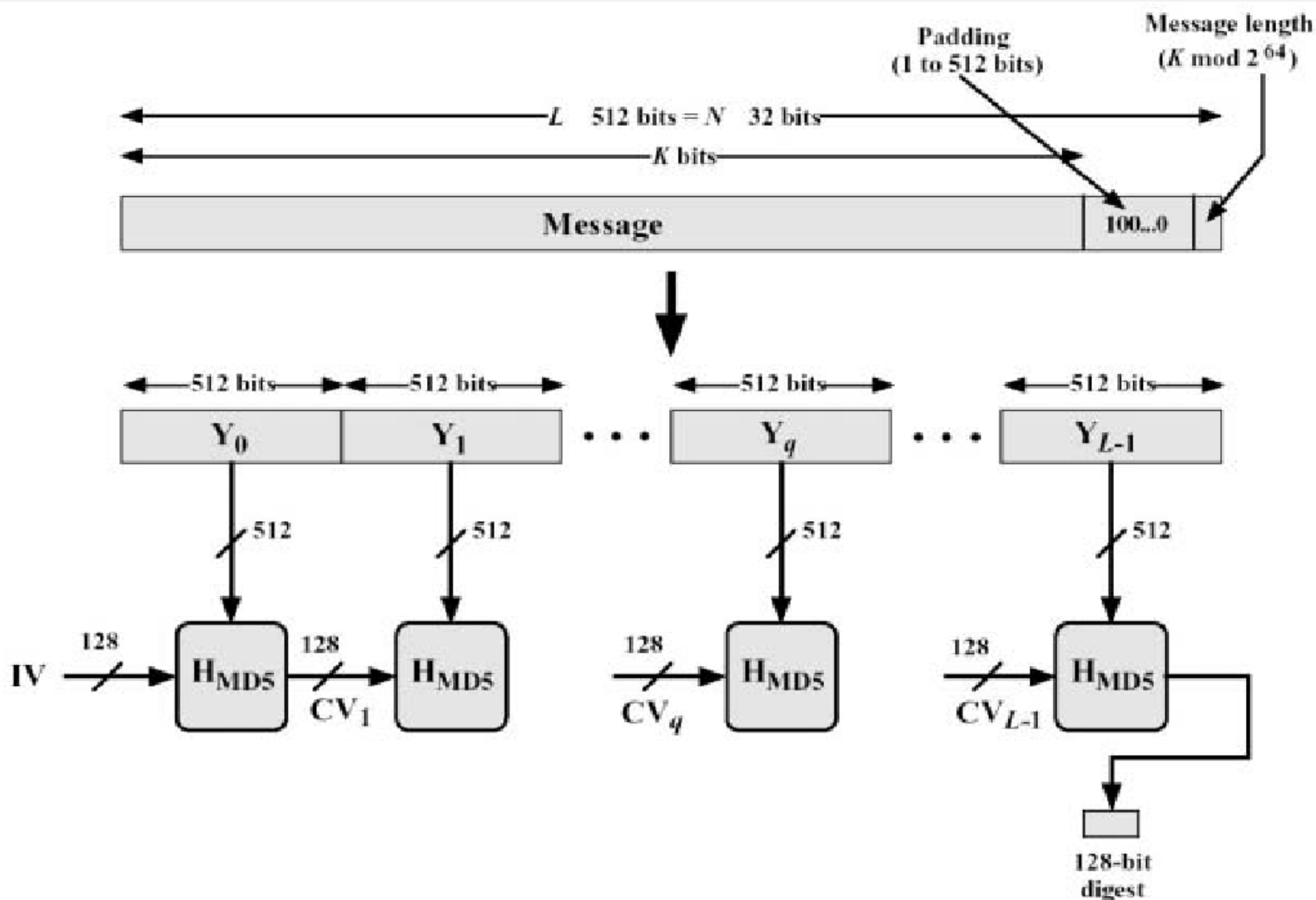


MD5描述

- 输入：任意长度的报文
- 输入分组长度：512 bit
- 输出：128 bit 报文



MD5原理图





MD5算法描述—step 1

- 附加长度值
- 对报文进行填充，使其比特数与448模512同余，即填充长度为512的整数倍减去64

$$|M_1| \equiv 448 \pmod{512}$$

$$\text{if } |M| \equiv 448 \pmod{512}, \text{ then } |M_1| = |M| + 512$$

- 填充方法：填充比特串的最高位为1，其余各位均为0



MD5算法描述—step 2

- 附加长度值

$$M_2 = M_1 \parallel Length \quad M = M \parallel Padding \parallel Length$$

$Length = |M| \bmod 2^{64}$, 低位字节优先, 表示为64bit长

若初始长度大于 2^{64} , 仅使用该长度的低64bit

- $|M_2|$ 为512的倍数: Y_0, Y_1, \dots, Y_{L-1}



MD5算法描述—step 3

- 初始化MD (Merkle Damgård) 缓存
- MD为128bit, 用于存放散列函数的中间及最终结果
- MD可表示为4个32bit的寄存器(A,B,C,D), 初始化如下:

A = 01 23 45 67 (0x67452301)

B = 89 AB CD EF (0xEFCDAB89)

C = FE DC BA 98 (0x98BADCFE)

D = 76 54 32 10 (0x10325476)



MD5算法描述—step 4

- 压缩：4个循环的压缩算法

$$CV_0 = IV$$

$$CV_i = H_{MD5}(CV_{i-1}, Y_i)$$



MD5算法描述—step 5

- 输出

$$\mathbf{MD} = \mathbf{CV}_L$$



其它常用Hash算法

- SHA-1
- RIPEMD-160
- HMAC



Hash小结

- Hash函数把变长信息映射到定长信息
- Hash函数不具备可逆性
- Hash函数速度较快
- Hash函数与对称密钥加密算法有某种相似性
- 对Hash函数的密码分析比对称密钥密码更困难
- Hash函数可用于消息摘要
- Hash函数可用于数字签名



本章内容提要

- 公钥密码
- 散列函数
- 密码学相关应用



密码学相关应用

- 数字签名
- 公钥基础设施 (PKI)



数字签名



数字签名的性质 (1/2)

- 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
 - 能与所签文件“绑定”
 - 签名者不能否认自己的签名
 - 容易被自动验证
 - 签名不能被伪造



数字签名的性质 (2/2)

- 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



数字签名的设计要求

- 签名必须是依赖于被签名信息的一个位串模板
- 签名必须使用某些对发送者是唯一的信息，以防止双方的伪造与否认
- 必须相对容易生成该数字签名
- 必须相对容易识别和验证该数字签名
- 伪造该数字签名在计算复杂性意义上具有不可行性，既包括对一个已有的数字签名构造新的消息，也包括对一个给定消息伪造一个数字签名
- 在存储器中保存一个数字签名副本是现实可行的



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)
- 安全性
 - 无条件安全的数字签名
 - 计算上安全的数字签名
- 可签名次数
 - 一次性的数字签名
 - 多次性的数字签名



公钥基础设施



为什么需要公钥基础设施?

- 可信赖的身份是一切交互活动的基础
 - 结交朋友
 - 贸易往来
- 可信电子身份是电子网络交互的基础
 - 电子商务
 - 电子邮件
- 可信赖身份的获得
 - 朋友的介绍
 - 权威机构颁发的身份证明
 - 公安局颁发的身份证
 - 工商局颁发的营业执照



为什么仅仅有公钥/私钥还不够？

- 仅仅公钥/私钥并不能保证可信身份
 - 数字签名：公布虚假公钥，用虚假私钥签名
 - 加密数据：用非法公钥替换合法公钥，窃听数据
- 需要一种安全有效的发布公钥的机制
 - 提供可信身份
 - 保证公钥和可信身份的对应关系
 - 保证公钥没有被篡改



公开密钥基础设施PKI

- 1976年Diffie和Hellman在《密码新方向》中提出了著名的D-H密钥交换协议，标志着公钥密码体制的出现。Diffie和Hellman第一次提出了不基于秘密信道的密钥分发，这就是D-H协议的重大意义所在。
- PKI (Public Key Infrastructure) 是一个用公钥概念与技术来实施和提供安全服务的具有普适性的安全基础设施。PKI公钥基础设施的主要任务是在开放环境中为开放性业务提供数字签名服务。



PKI之描述

- PKI是一个用非对称密码算法原理和技术来实现并提供安全服务的具有通用性的安全基础设施，是一种遵循标准的利用公钥加密技术为网上电子商务、电子政务的开展，提供一整套安全的基础平台。
- PKI管理平台能够为网络中所有需要采用加密和数字签名等密码服务的用户提供所需的密钥和证书管理，用户可以利用PKI平台提供的安全服务进行安全通信。



PKI之动机

- 公钥技术
 - 如何提供数字签名功能
 - 如何实现不可否认服务
 - 公钥和身份如何建立联系
 - 为什么要相信这是某个人的公钥
 - 公钥如何管理
- 方案：引入证书(certificate)
 - 通过证书把公钥和身份关联起来



PKI提供的基本服务

- 认证

- 采用数字签名技术，签名作用于相应的数据之上
 - 被认证的数据 —— 数据源认证服务
 - 用户发送的远程请求 —— 身份认证服务

- 完整性

- PKI采用了两种技术

- 数字签名：既可以是实体认证，也可以是数据完整性
- MAC(消息认证码)：如DES-CBC-MAC或者HMAC-MD5



PKI提供的基本服务

- 保密性

- 用公钥分发随机密钥，然后用随机密钥对数据加密

- 不可否认

- 发送方的不可否认 —— 数字签名

- 接受方的不可否认 —— 收条 + 数字签名

- 历史行为的不可否认



PKI的应用考虑 (1/2)

- 在提供前面四项服务的同时，还必须考虑

—性能

- 尽量少用公钥加解密操作，在实用中，往往结合对称密码技术，避免对大量数据作加解密操作
- 除非需要数据来源认证才使用签名技术，否则就使用MAC或者HMAC实现数据完整性检验

—个体命名

- 如何命名一个安全个体，取决于CA的命名登记管理工作



PKI的应用考虑 (2/2)

—在线和离线模型

- 签名的验证可以在离线情况下完成
- 用公钥实现保密性也可以在离线情况下完成
- 离线模式的问题：无法获得最新的证书注销信息

—证书中所支持算法的通用性

- 在提供实际的服务之前，必须协商到一致的算法



PKI要处理的问题

- 初始身份的确认
- 好密钥的安全生成
- 证书的颁发、更新和终止
- 证书有效性的检查
- 证书和相关信息的分发
- 密钥的安全存档和恢复
- 签名和时间戳的产生
- 信任关系的建立和管理



PKI基本组成

- PKI由以下几个基本部分组成：

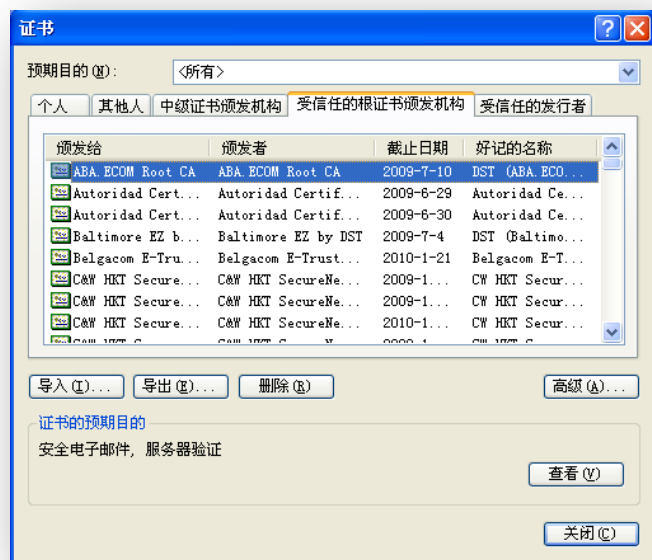
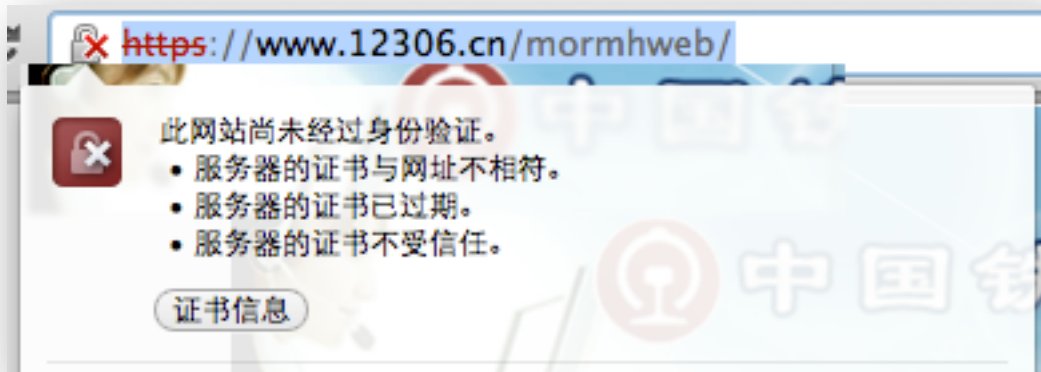
- 公钥证书
- 证书作废列表 (CRL)
- 策略管理机构 (PMA)
- 认证机构 (CA)
- 注册机构 (RA)
- 证书管理机构 (CMA)
- 证书存档 (Repository)
- 署名用户 (Subscriber)
- 依赖方 (Relying party)
- 最终用户 (End User)

教材内容

- CA
- 证书和证书库
- 密钥备份及恢复系统
- 密钥和证书的更新系统
- 证书历史档案
- 应用接口系统
- 交叉认证



身边的数字证书





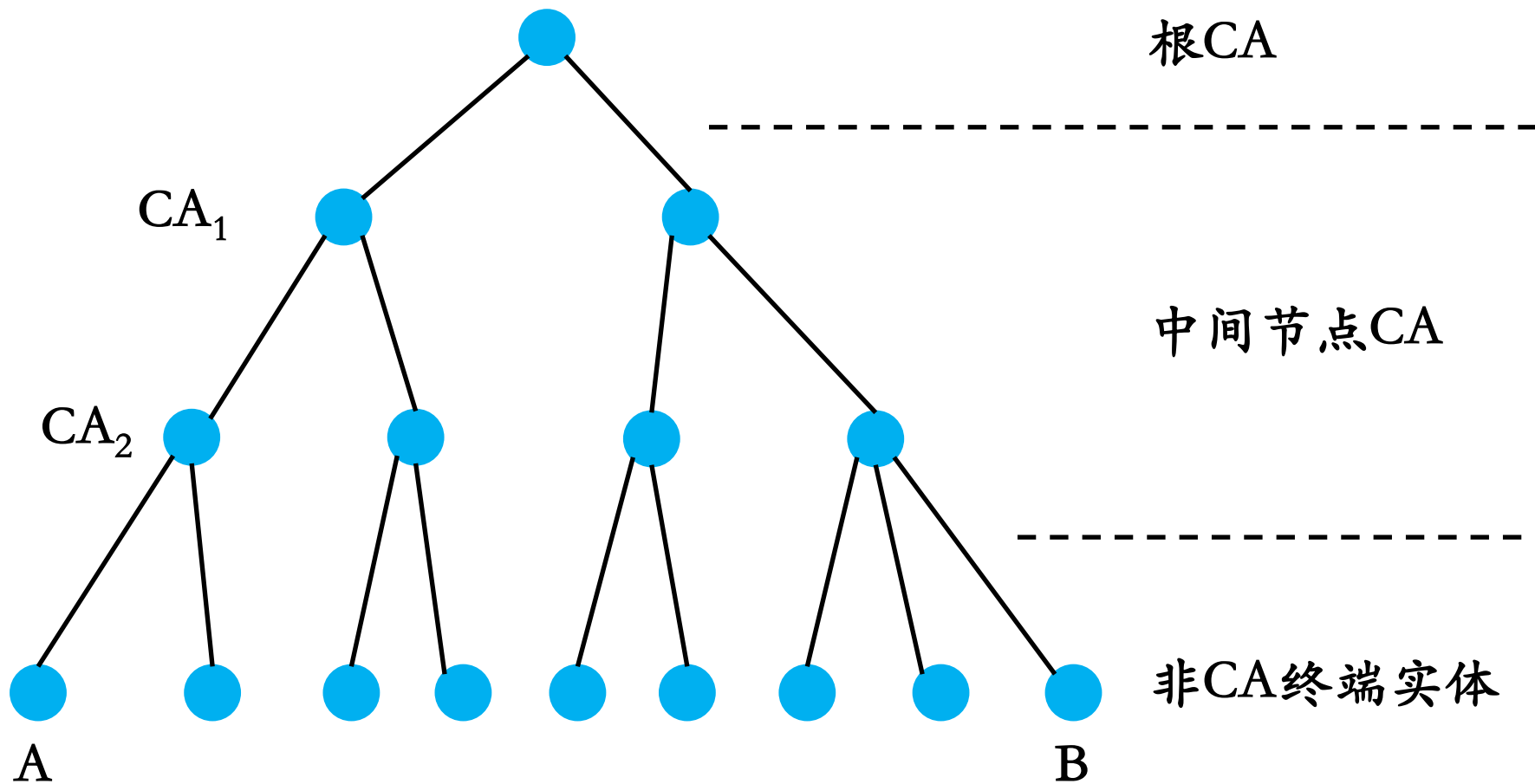
PKI 交叉认证模型

- 树状认证模型
- 网状认证模型
- 桥式模型
- 信任列表模型
- 相互承认模型



PKI信任模型(1/2)

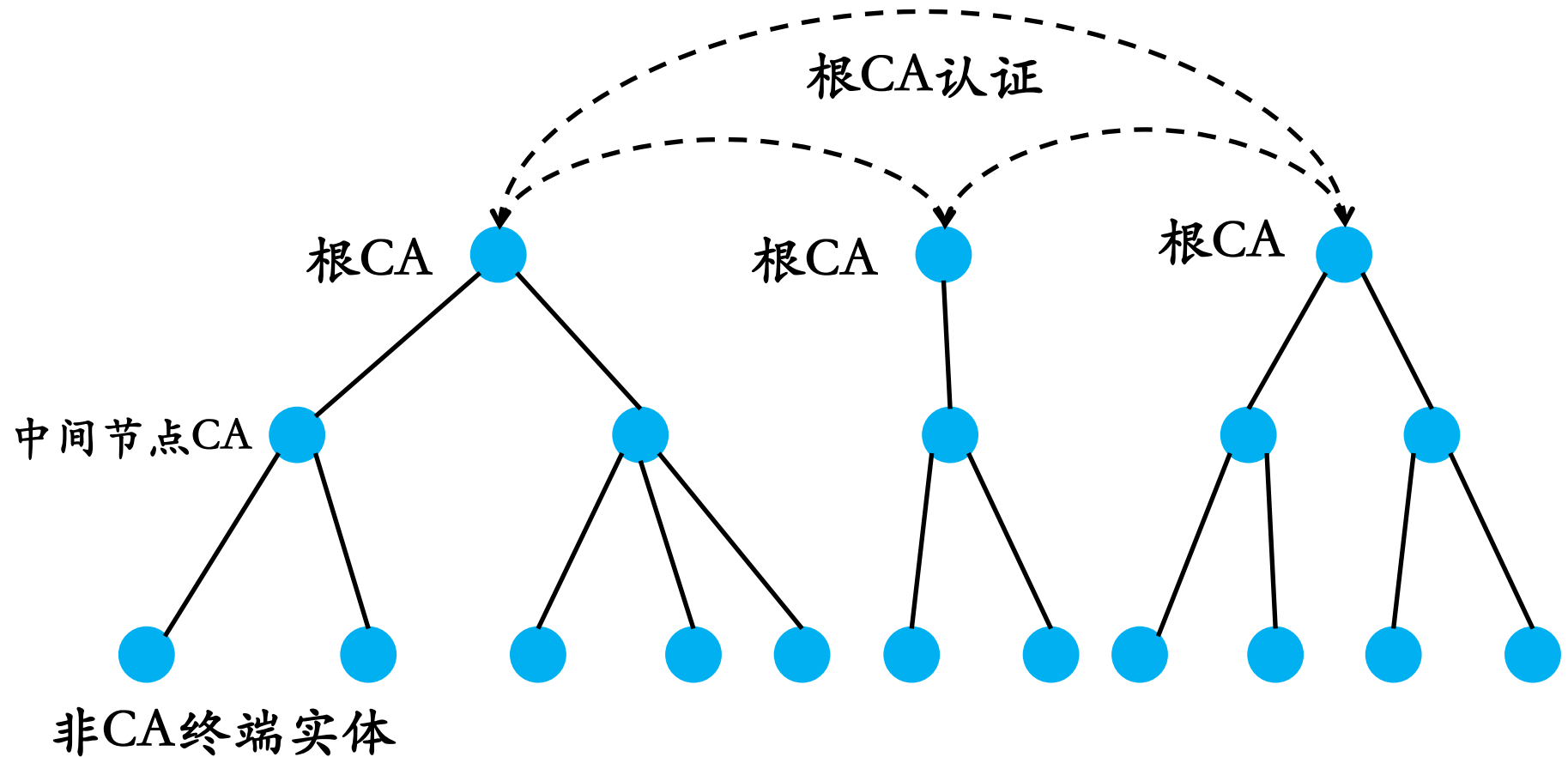
- 严格层次结构模型（树状认证模型）





PKI信任模型(2/2)

- 分布式结构模型





CA如果被黑?

- 荷兰CA供应商DigNotar因黑客入侵而导致破产
 - 疑似伊朗黑客在2011年7月中旬入侵了DigiNotar服务器，发行了531个伪造证书，包括了Google、微软、雅虎、Twitter、Facebook、中情局、军情六处和摩萨德等
 - 2011年8月29日 Microsoft 安全通报 (2607712) 虚假数字证书可导致欺骗



证书如果被伪造?

- 2002年9月，Windows系统漏洞导致无法检测出伪造的漏洞
 - 当有人使用S/MIME接受伪造的署名邮件，或利用SSL访问伪造的Web站点时，Windows上的应用将不会发出警告。
- 2012年6月，Microsoft 安全通报 (2718704) 未经授权的数字证书可导致欺骗
 - Flame病毒利用该漏洞伪装成Windows补丁更新程序绕过了所有市面上安全软件的查杀
- 2012年9月，黑客已经成功嵌入Adobe公司的数字签名伪装成Adobe开发的软件
 - 主要受影响的是Windows平台的Adobe软件和三个在Mac和Windows平台上使用Adobe Air的应用，目前其他Adobe软件并未受到影响
 - Adobe官方的一个软件构建服务器中未发现被植入恶意代码



安全需要信任，安全基于信任

中国传媒大学



应用演示

中国传媒大学



计算任意文件的散列值

- HashTab

- 直接集成在Windows资源管理中

- 计算文件的 MD5、SHA1 与 CRC-32 哈希值

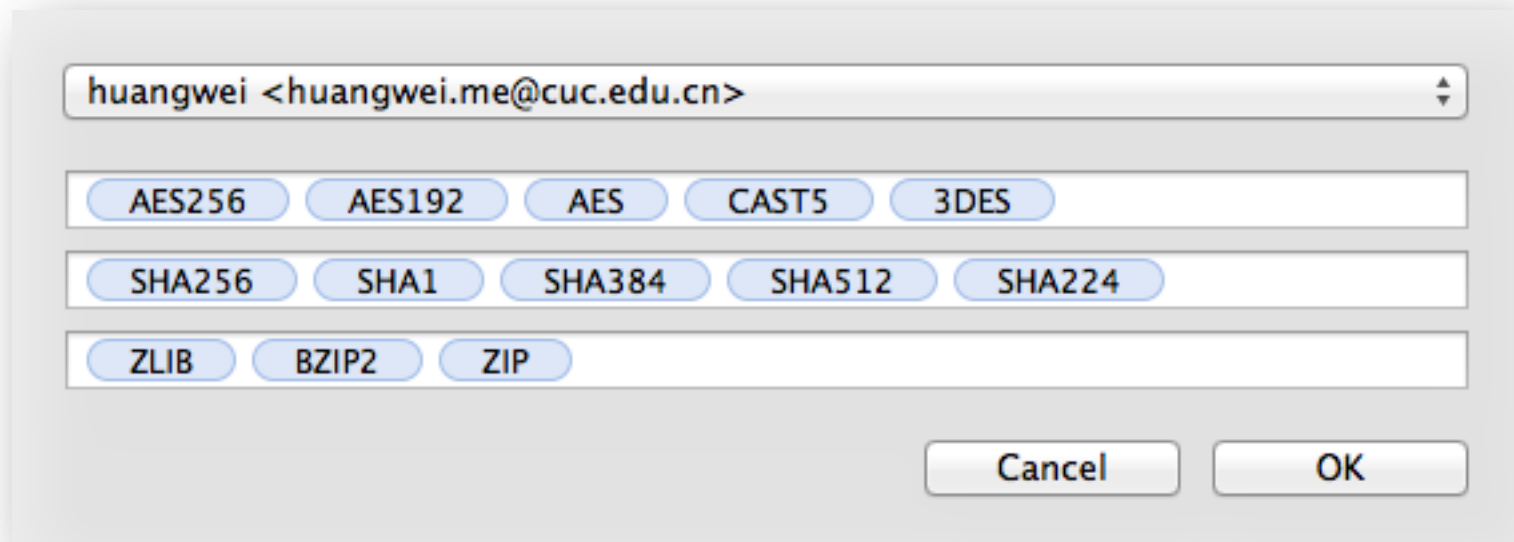
- 比较两个文件的哈希值是否相等





非对称加解密

- GnuPG
—GNU Privacy Guard或GPG
- 开源加密与签名解决方案





在Windows上使用GnuPG

- GnuPG (win32cli-1.4.9.exe)
 - <ftp://ftp.gnupg.org/gcrypt/binary/gnupg-w32cli-1.4.9.exe>
- gnupgshell-1.0.0.windows.zip
 - <http://www.tech-faq.com/gnupg-shell/gnupgshell-1.0.0.windows.zip>
- GPGee-1.4.0-BETA1
 - <http://www.net-security.org/gotogo.php?cat=2&id=642>



使用GPG Shell管理你的公私钥对

PGP Key Generation Assistant

Name and Email Assignment

must have a name associated with it. The name and Email
or correspondents know that the public key they are using

huangwei.cuc
(Name must be at least 5 characters long)
demo@huangwei.me
demo for teaching

Click advanced for more key settings.

Advanced Key Settings

Key type: RSA

☐ Generate separate signing subkey

Key size: 2048 1024-4096

Expiration: ☒ never ☐ 2012- 9-22

OK Cancel


Advanced...

Help < Back Next > Cancel



设置私钥访问口令

PGP Key Generation Assistant



Passphrase Assignment

Your private key will be protected by a passphrase. It is important to keep this passphrase secret and do not write it down.

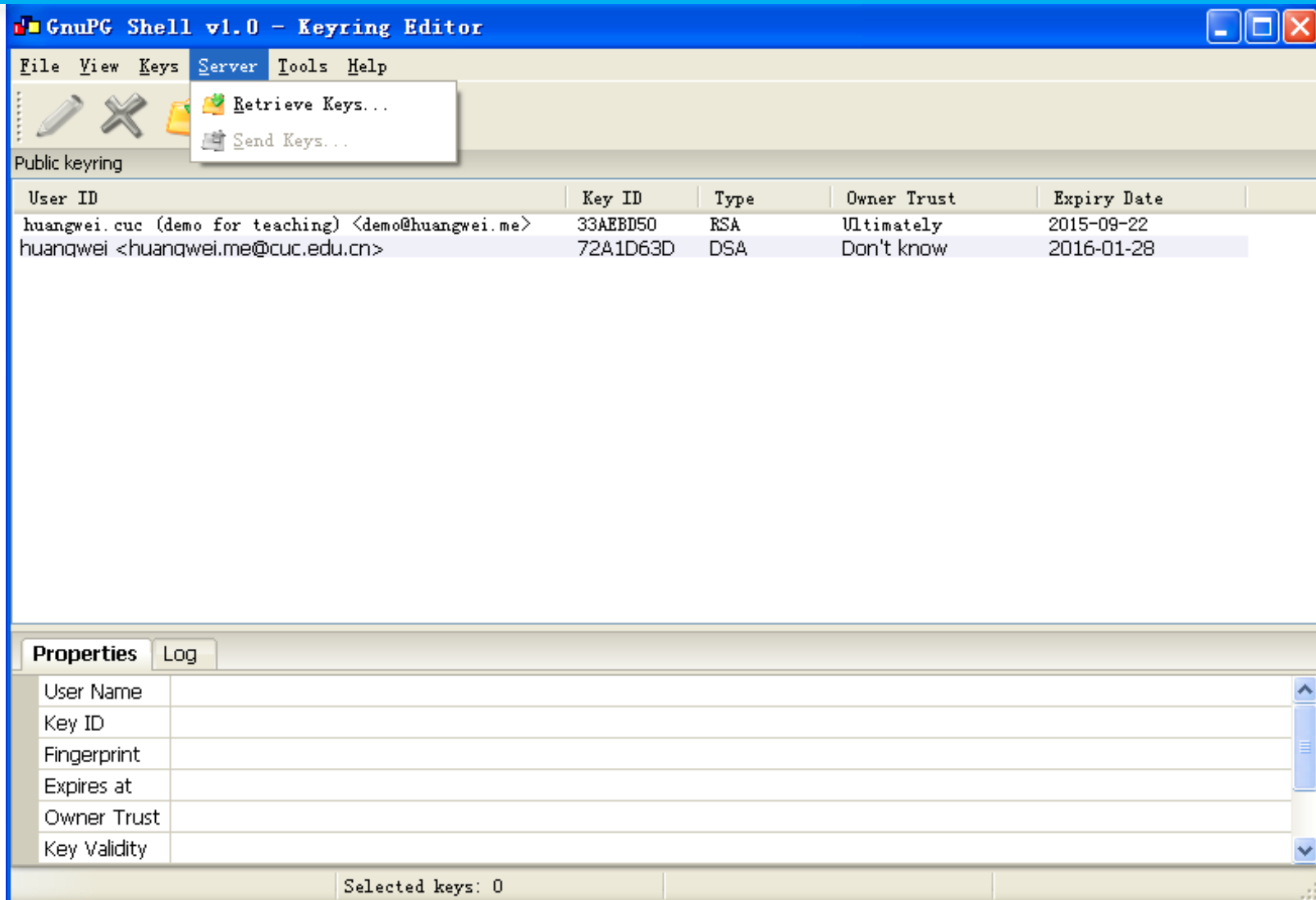
Your passphrase should be at least 8 characters long and should contain non-alphabetic characters.

Passphrase:

Confirmation:



从公钥服务器导入他人的公钥





在公钥服务器上检索公钥

- 也可通过
name进行检索

Import Key

Choose server:

hkp://keys.gnupg.net

Key to import

Key ID: 72A1D63D

Name:

Search Clear

Search results:

Key ID	Name
72A1D63D	huangwei <huangwei.me@cuc.edu.cn>

Import Cancel



编辑已有密钥

GnuPG Shell v1.0 - Keyring Editor

File View Keys Server Tools Help

Public keyring

User ID	Key ID	Type	Owner Trust	Expiry Date
huangwei.cuc (demo for teaching) <demo@huangwei.me>	33AEBD50	RSA	Ultimately	2015-09-22
huangwei <huangwei.me@cuc.edu.cn>	72A1D63D	DSA	Don't know	2016-01-28

Edit...
Edit Owner Trust...
Delete
Export

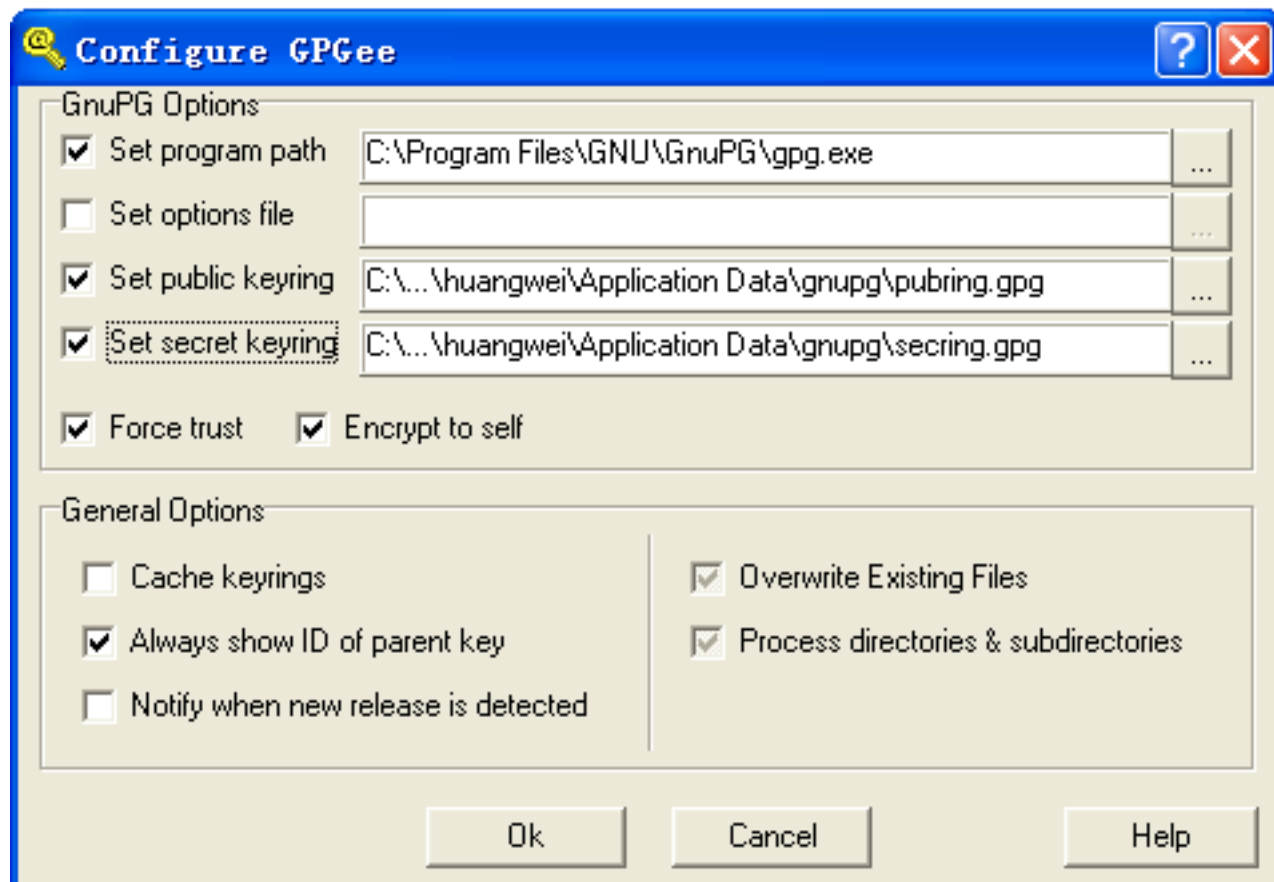
Properties Log

User Name	huangwei <huangwei.me@cuc.edu.cn>
Key ID	72A1D63D
Fingerprint	9CEF681E55F0B821F847C48FF6B48A0C060EA483
Expires at	2016-01-28
Owner Trust	Don't know
Key Validity	Fully Valid

Total keys: 2 Selected keys: 1

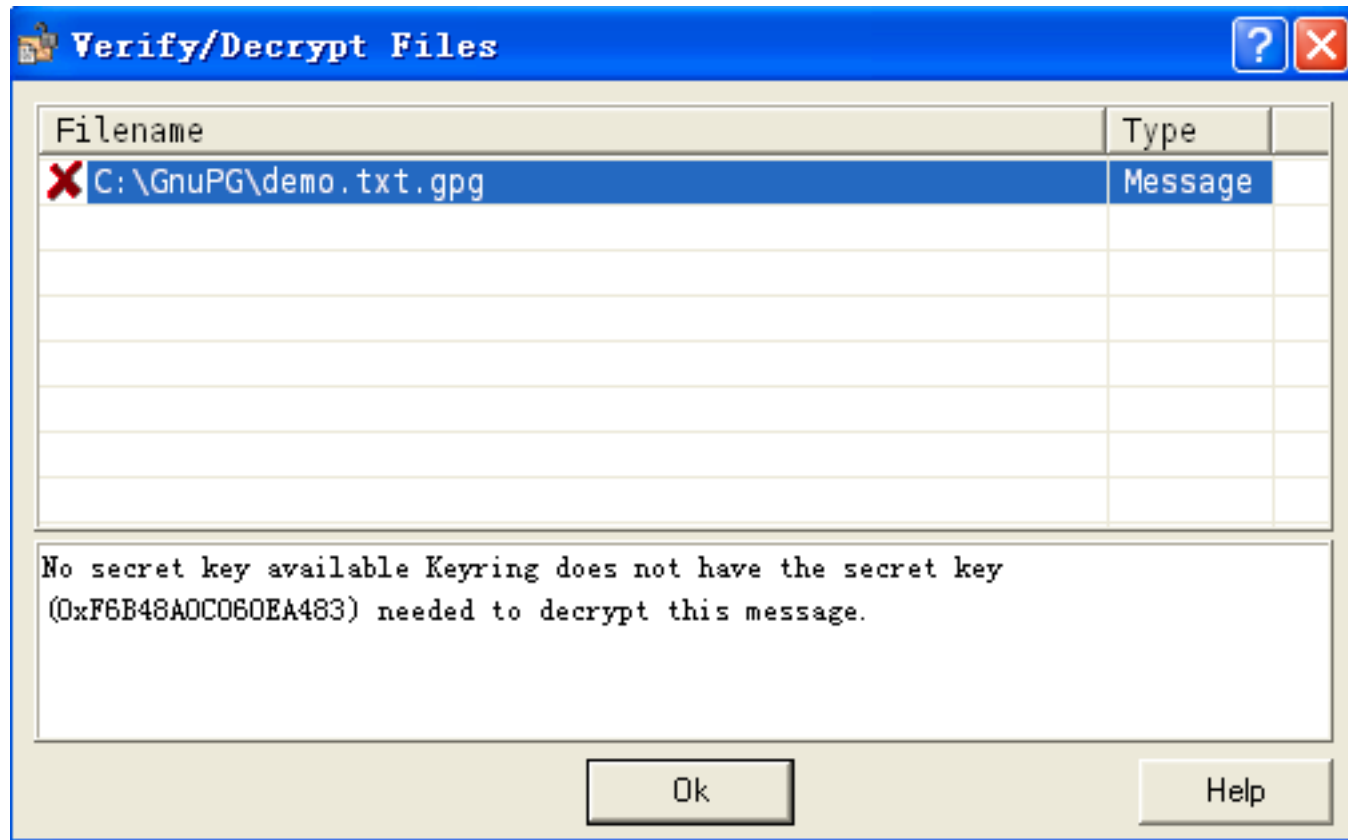


设置GPGe相关参数





没有私钥无法解密





备份你的私钥

- GPG Shell不支持直接导出私钥，只能导出公钥
- 但GPG命令行程序可以！

— `gpg --export-secret-keys -a "User Name" > private.key`

```
C:\Program Files\GNU\GnuPG>gpg --list-secret-keys
C:/Documents and Settings/huangwei/Application Data/gnupg\secring.gpg
-----
sec  2048R/33AEBD50 2012-09-22 [expires: 2015-09-21]
uid          huangwei.cuc <demo for teaching> <demo@huangwei.me>

C:\Program Files\GNU\GnuPG>gpg --export-secret-keys -a "huangwei.cuc"
-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: GnuPG v1.4.9 (MingW32)

lQ0+BFBdewIBCAC+cUaMz7eQpCG6J44NSXAUFhi1ORXWqKQbUu3kUGhMIQyFUjD0
```



使用GPGee加密/解密/签名/验证签名

Show Time

中国传媒大学



课后思考题

- 试分析保密通信系统模型中所有可能的风险点？



课后作业

• GnuPG任务

—导入老师的公钥

- `huangwei <huangwei.me@cuc.edu.cn>`
- 公钥Short ID: 72A1D63D
- 公钥服务器地址: `hkp://keys.gnupg.net`

—用我的公钥加密以下数据（文本文件）

- 你的姓名/专业/学号/邮箱地址

以上请填写你的真实信息，以便登记作业

—将加密后文件发送给我（二选一）

- 邮箱: huangwei.me@cuc.edu.cn
- 新浪微博私信: @中传黄玮

中国传媒大学