问题

为什么TCG组织认为TPM没有必要支持对称加密算法？

现在黑客开始趋向于攻击普通用户

- **黑产一：暗扣话费**
- 此类黑产以稀缺的服务提供商为上游，开发人员根据不同的服务提供商资源开发相应的软件，并将这些软件植入到伪装成色情、游戏、交友的应用中，实现暗扣话费。
- 据腾讯数据，每天互联网上约新增2750个此类新病毒变种，每天影响数百万用户，按人均消耗几十元话费估算，日掠夺话费金额数千万元。

现在黑客开始趋向于攻击普通用户

- **黑产二：广告流量变现**
- 某些内置于各类应用中的恶意广告联盟，通过恶意推送广告进行流量变现。这些广告内容大多没有底线，时常推送打色情擦边球的应用、博彩甚至手机病毒等。
- 腾讯数据显示，平均每天新增广告病毒变种257个，影响大约676万个用户。越是经济发达的地区，情况越严重。

现在黑客开始趋向于攻击普通用户

- **黑产三：手机应用分发**
- 软件推广难，部分厂商便找到相对便宜的渠道：通过手机应用分发黑产，用类似病毒的手法在用户手机上安装软件。
- 例如，有用户经常会发现手机里莫名其妙多出一些应用，这就是黑产人员通过手机恶意软件后台下载推广的应用，是手机黑产的变现途径。
- 据腾讯监测，手机恶意推广的病毒变种每天新增超过2200个，每天受影响的用户超过1000万个，主要影响中低端手机用户。
- 中，自动执行刷量操作。

现在黑客开始趋向于攻击普通用户

- **黑产四：木马刷量**
- 木马刷量黑产主要通过作弊手段骗取开发者推广费。
- 它有三种模式：通过模拟器模拟出大量手机设备伪装真实用户刷量；用“手机做任务轻松赚钱”等噱头吸引用户入驻平台后，欺骗用户使用某个App实现刷量；木马技术自动刷量。
- 目前主流的是利用木马刷量。木马开发者通过合作的方式，将木马植入到一些用户刚需应用中，再通过云端控制系统下发任务到用户设备中，自动执行刷量操作。

现在黑客开始趋向于攻击普通用户

- **黑产五：勒索病毒**
- 勒索病毒攻击者会通过弱口令漏洞入侵企业网站，再以此为跳板渗透到内网，然后利用局域网漏洞攻击工具，将勒索病毒分发到内网关键服务器，将企业核心业务及备份服务器数据加密。
- 病毒一旦得手，企业日常业务立刻陷于瘫痪。如果连备份系统也被破坏了，那基本只剩缴纳赎金这一条路。
- 勒索病毒产业链分工明确，有人负责制作勒索病毒生成器，有网站资源的人负责分发，各方参与利益分成。

现在黑客开始趋向于攻击普通用户

- **黑产六：控制肉鸡挖矿**
- 2017年底，腾讯监测发现，一款名为“ tlMiner”的挖矿木马在当年12月20日的传播量达到峰值，当天有近20万台机器受到该挖矿木马影响。
- 腾讯相关团队最终挖掘到一个公司化运营的大型挖矿木马黑色产业链。为非法牟利，这家公司搭建木马平台，发展下级代理商近3500个，通过网吧渠道、“吃鸡”外挂、盗版视频软件等投放木马，非法控制用户电脑终端389万台，进行数字加密货币挖矿、强制广告等非法业务，非法获利1500余万元。

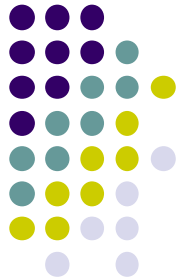
现在黑客开始趋向于攻击普通用户

- **黑产七：DDoS攻击**
- DDoS攻击，即分布式拒绝服务攻击。这是利用网络上已被攻陷的电脑（“肉鸡”）向目标电脑发动密集的“接受服务”请求，借以把目标电脑的网络资源及系统资源耗尽的一种攻击方式。
- 黑产人员通常将“肉鸡”联合起来，进行带有利益性的网站刷流量、邮件群发、瘫痪竞争对手等活动。因为攻击效果立竿见影、利益巨大，利用DDoS进行勒索、攻击竞争对手的情况越来越多，产业链分工越来越细。

目前的状态

- 目前的个人计算机（和相关的设备，如PDAs、智能手机，等）具有许多根本的安全和信任问题，包括：
  - 用户无法知道他的计算机在做什么
  - 用户无法知道其他人在做什么
  - 计算机消耗了越来越多的资源来扫描病毒或者攻击
  - 用户很容易被欺骗以访问恶意虚假网站
  - 笔记本信息的丢失
  - 服务器信息的丢失

问题的核心



- 这些问题的核心是基于这样一个事实：标准个人计算机结构不是专门为安全设计的

可信计算的根本目标

- 可信计算的根本目标就是解决上述问题
- 核心问题: 我们如何知道系统真正地在做我们认为它在做的事情？
- 最终, 这个问题可以通过修改硬件，以提供良好的信任基来实现，因为软件总是可以被改变的。

可信计算解决的攻击

- 可信计算的目的是解决所有的软件攻击，但只解决部分硬件攻击

商业限制

- 任何解决方案必须满足许多商业限制，包括：
- 开销（**Cost**）
- 兼容性（**Backwards compatibility**）
- 政府进出口法规**Government import/export regulations**
- 性能维护（**Maintain performance**）

短期目标: 保护存储（**Protected storage**）

- Customers can encrypt data on their hard disks in a way that is much more secure than software solutions.
- A standardized crypto-API gives access to hardware, which provides:
  - A (true) random number generator;
  - The potential for (unlimited amounts of) encrypted storage, where *no master key is stored on disk*, and *no password derivatives are stored in plaintext*.
- Digital signature keys can be protected and used in a secure hardware environment.

中期目标: 平台完整性（**Platform integrity**）

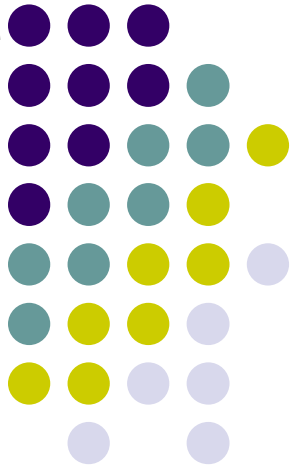
- Automatic prevention of access to information, if undesirable programs are executing.
- Enhanced data confidentiality through confirmation of ***platform integrity*** prior to ***decryption of sensitive data*** (system will only decrypt or permit access to secrets if the software environment is what was intended).

长期目标: 安全电子商务（Secure e-commerce）

- Customers and their partners, suppliers and customers can connect IT systems and expose only the data that is intended to be exposed.
- Platform identities and integrity metrics can be proven reliably to previously unknown parties.
- Secure online discovery of platforms and services:
  - Giving confidence in information about the ***software environment*** and the ***identity of a remote party***;
  - Enabling higher levels of trust when interacting with this party.

密码学及PKI简述

代炜琦
wqdai@hust.edu.cn

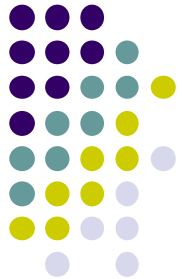


Outline

- 密码学基础:
  - 对称加密/非对称加密
  - PKI, X.509证书

对称加密

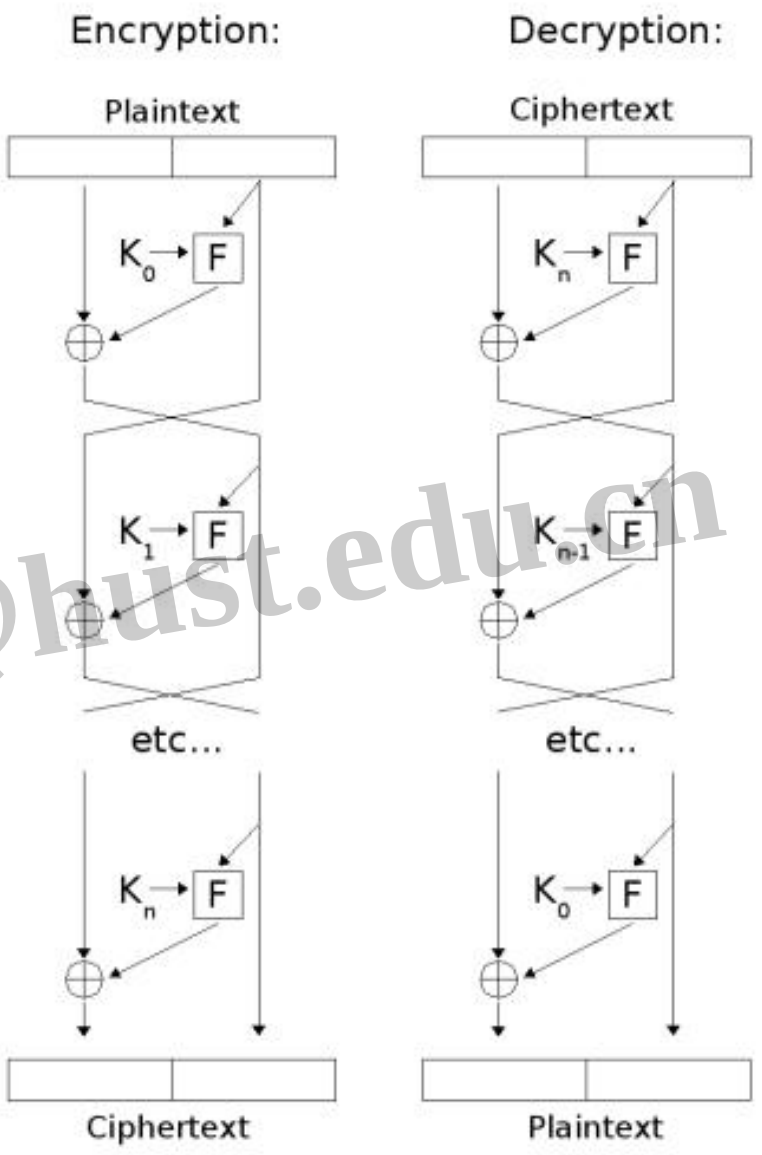
- Protects information using the same key for encryption and decryption
- Common symmetric algorithms: DES, AES, RC4, Blowfish



- Block cipher
- Feistel or modified Feistel: Blowfish, Camellia, CAST-128, DES, FEAL, ICE, KASUMI, LOKI97, Lucifer, MARS, MAGENTA, MISTY1, RC5, TEA, Triple DES (3DES), Twofish, XTEA, GOST_28147-89
- Generalised Feistel: CAST-256, MacGuffin, RC2, RC6, Skipjack, SMS4

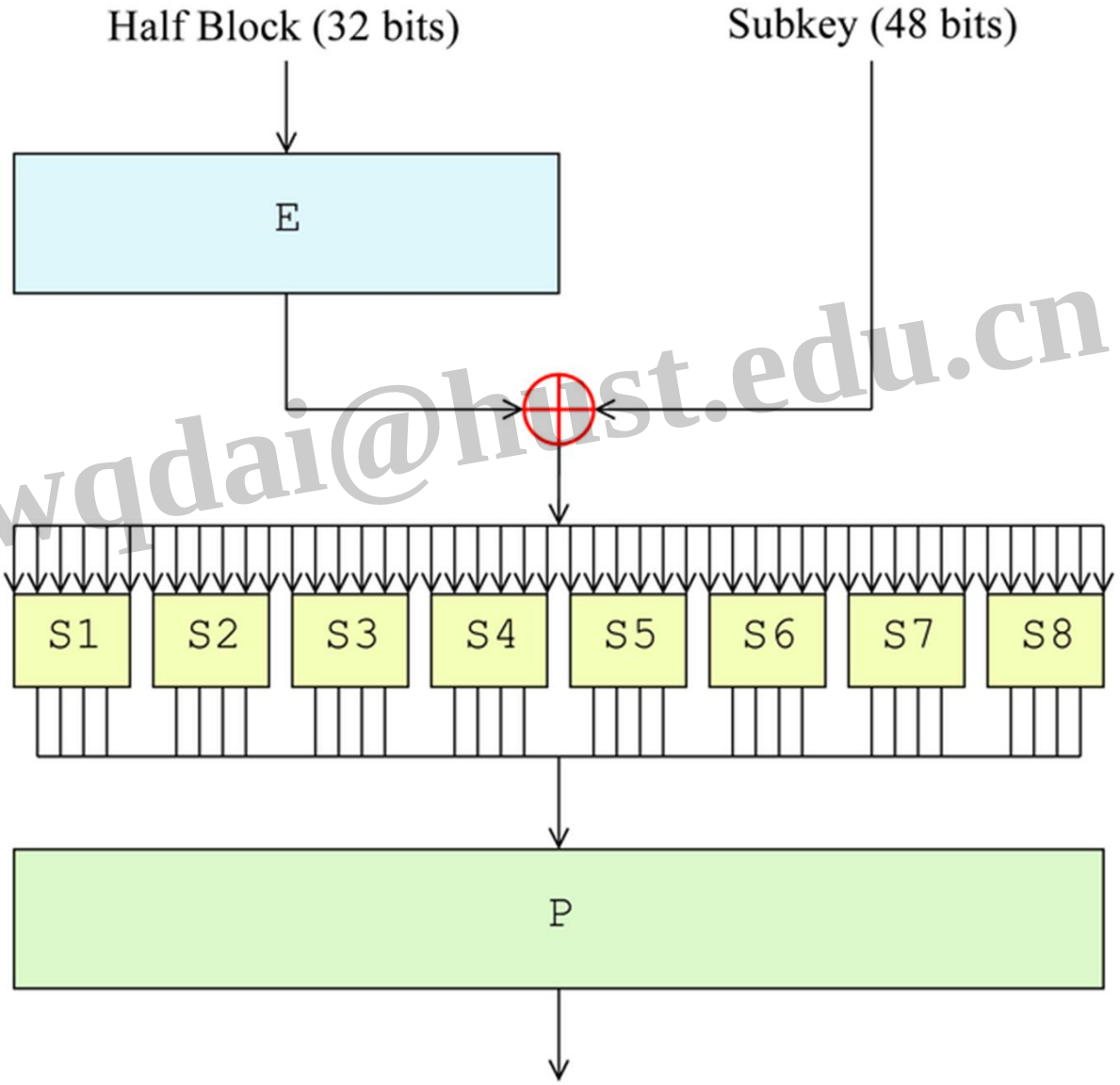
DES (Data Encryption Standard)

Feistel

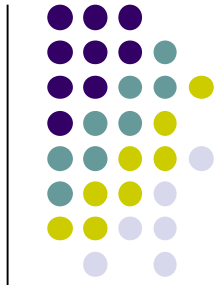


Feistel Cipher

DES: Function F

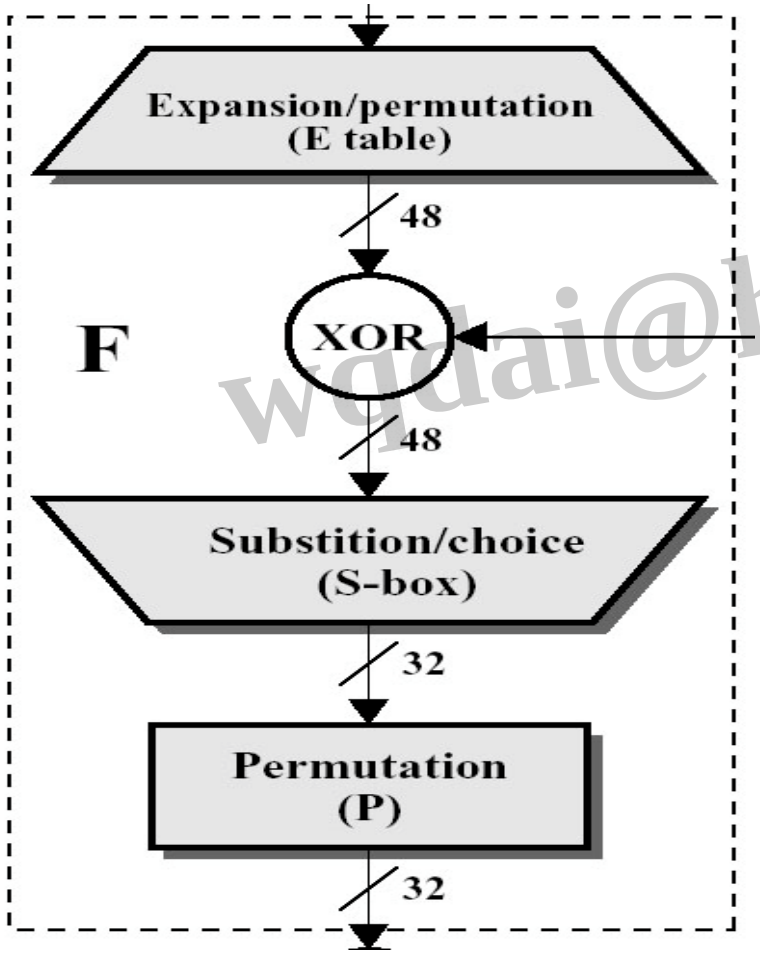


DES: Function F



S_5		Middle 4 bits of input															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Outer bits	**00**	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

DES: Function F



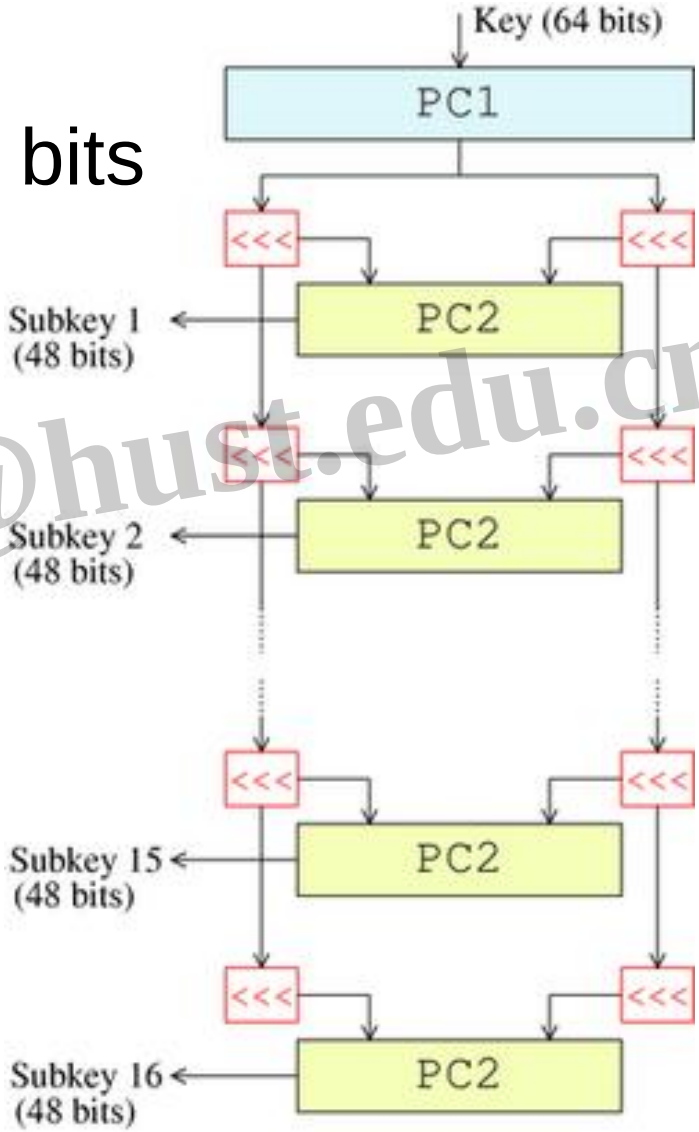
Expansion: 32 → 48

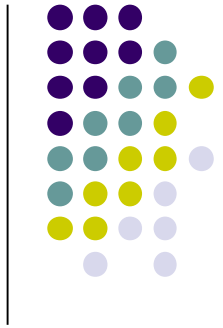
S-box: 6 → 4

Permutation:

The key-schedule of DES

- Parity checking: 8 bits
- Key: 56 bits



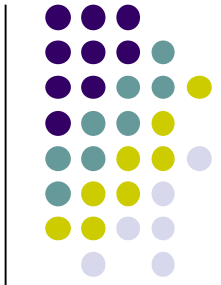


- DES: easy to attack
  - Brute force attack
    - 1998, US$250,000 56 hours
    - 2006, US$10,000 9 days;
    - 2007, 6.4 days;
    - 2008, less than 1 day
  - Other attacks

Triple-DES

- ciphertext = $E_{K3}(D_{K2}(E_{K1}(\text{plaintext})))$
- plaintext = $D_{K1}(E_{K2}(D_{K3}(\text{ciphertext})))$

Triple-DES

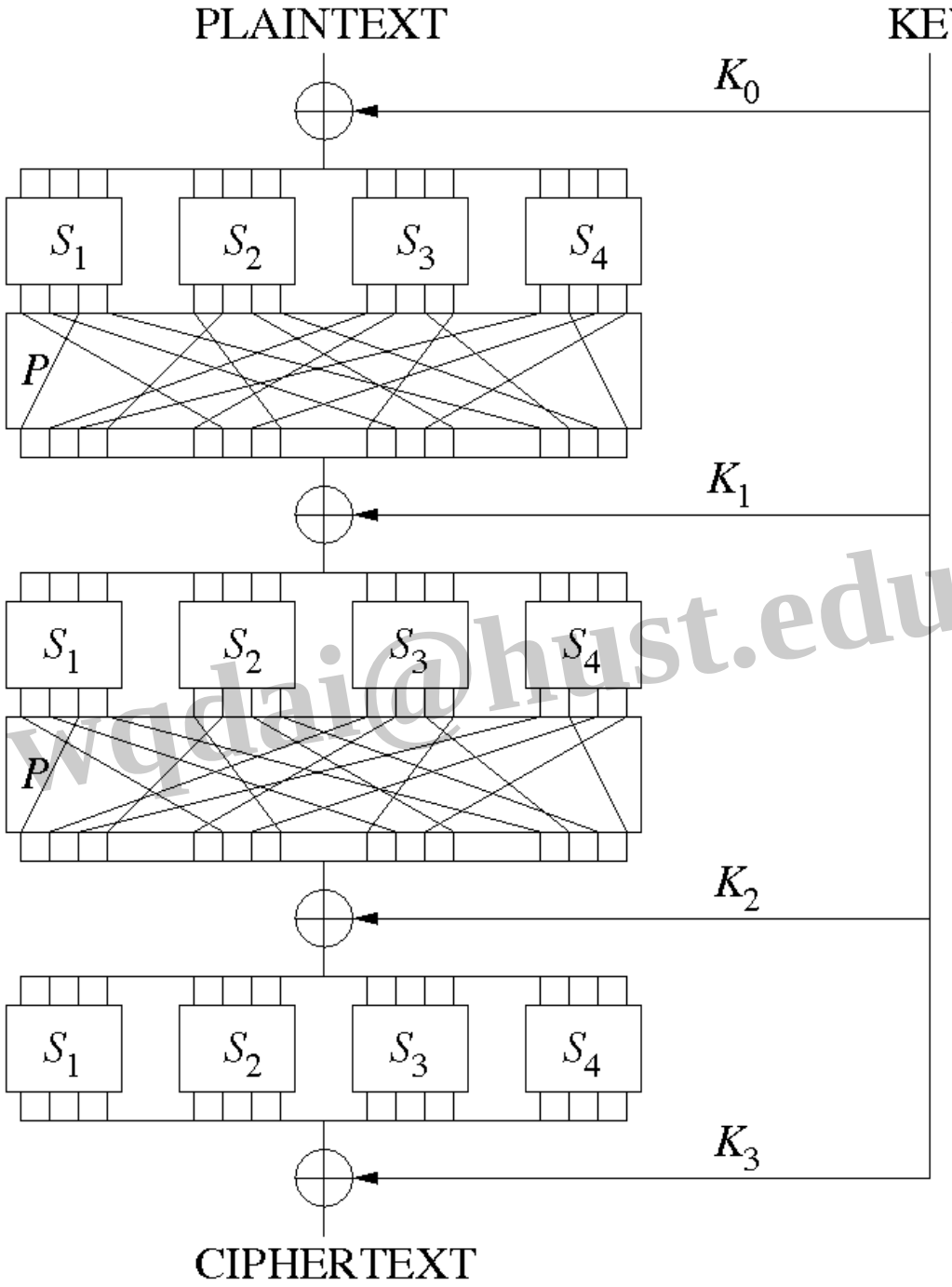


- The standards define three keying options:
  - Keying option 1: All three keys are independent. 3*56=168 key bits.
  - Keying option 2: K1 and K2 are independent, and K3 = K1.
    - 2*56=112 key bits.
    - IS 3DES the same as 2DES?
      - ciphertext = $E_{K2}(E_{K1}(\text{plaintext}))$ ---double DES
      - ciphertext = $E_{K1}(D_{K2}(E_{K1}(\text{plaintext})))$ ---triple DES
      - meet-*in-the-middle attacks (similar problem: birthday attacks)*. Effective length: 112-bits ➔56-bits
  - Keying option 3: All three keys are identical, i.e. K1 = K2 = K3.

AES (Advanced encryption standard)

- AES-128, AES-192 and AES-256
- Originally published as Rijndael
- Steps:
  - SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
  - ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
  - MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column
  - AddRoundKey—each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule.

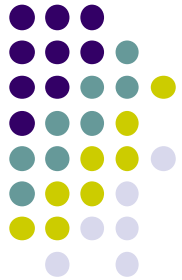
AES



A sketch of a Substitution-Permutation Network with 3 rounds, encrypting a plaintext block of 16 bits into a ciphertext block of 16 bits.

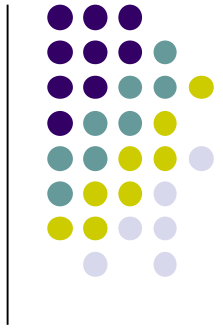
The S-boxes are the *Si*'s, the P-boxes are the same *P*, and the round keys are the *Ki*'s.

AES



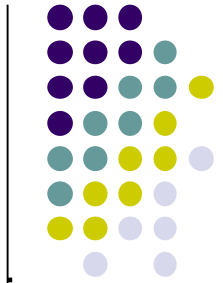
- Until May 2009, the only successful published attacks against the full AES were side-channel attacks on specific implementations

AES



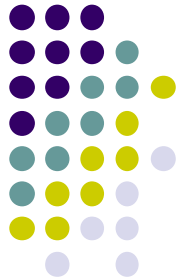
- How to encrypt a file
- Encrypt file.txt to file.enc using 256-bit AES in CBC (cipher-block chaining) mode, the output is base64 encoded
  - File.txt: Hello, we have a AES testing here.
  - *openssl enc -aes-256-cbc -a -salt -in file.txt -out file.enc*
  - enter aes-256-cbc encryption password:
  - Verifying - enter aes-256-cbc encryption password:
  - File.enc:
    U2FsdGVkX1/IptVFH9WL4txkcknTY8kwzPrBZTK0OnsZaosC5Dip4ykuuWewSdDZ
    Tv3ib1tDgYYT7Ju75NinLw==

Symmetric Encryption



- Symmetric-key cryptosystems use the same key for encryption and decryption of a message, though a message or group of messages may have a different key than others.
- A significant *disadvantage* of symmetric ciphers is the key management necessary to use them securely.
  - Each distinct pair of communicating parties must, ideally, share a different key, and perhaps each ciphertext exchanged as well.
  - The number of keys required increases as the square of the number of network members, which very quickly requires complex key management schemes to keep them all straight and secret.
  - The difficulty of securely establishing a secret key between two communicating parties, when a secure channel doesn't already exist between them, also presents a chicken-and-egg problem which is a considerable practical obstacle for cryptography users in the real world.

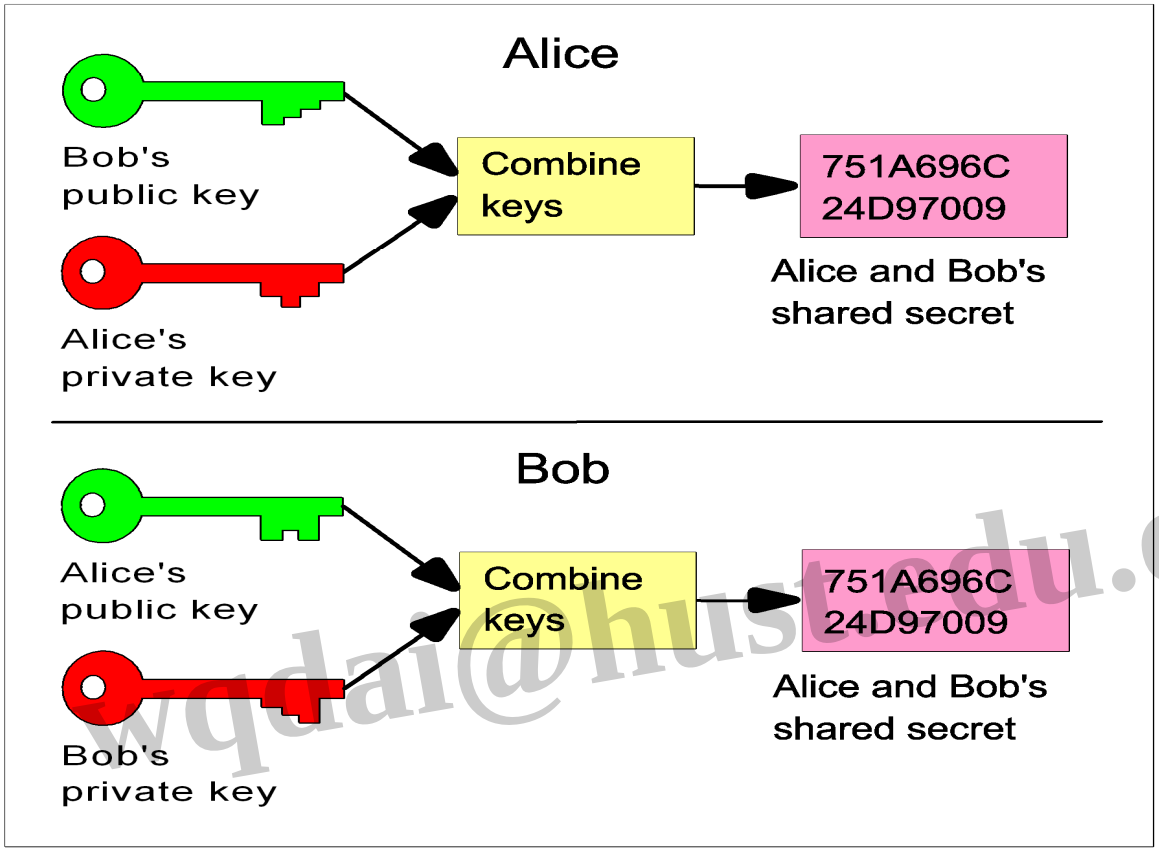
Asymmetric Encryption



- Uses a different key for encryption and decryption
  - Compared with symmetric encryption, orders of magnitude slower
  - The cipher text is larger than the plain text
- Also know as Public Key Cryptography
- Combination of symmetric encryption and asymmetric encryption

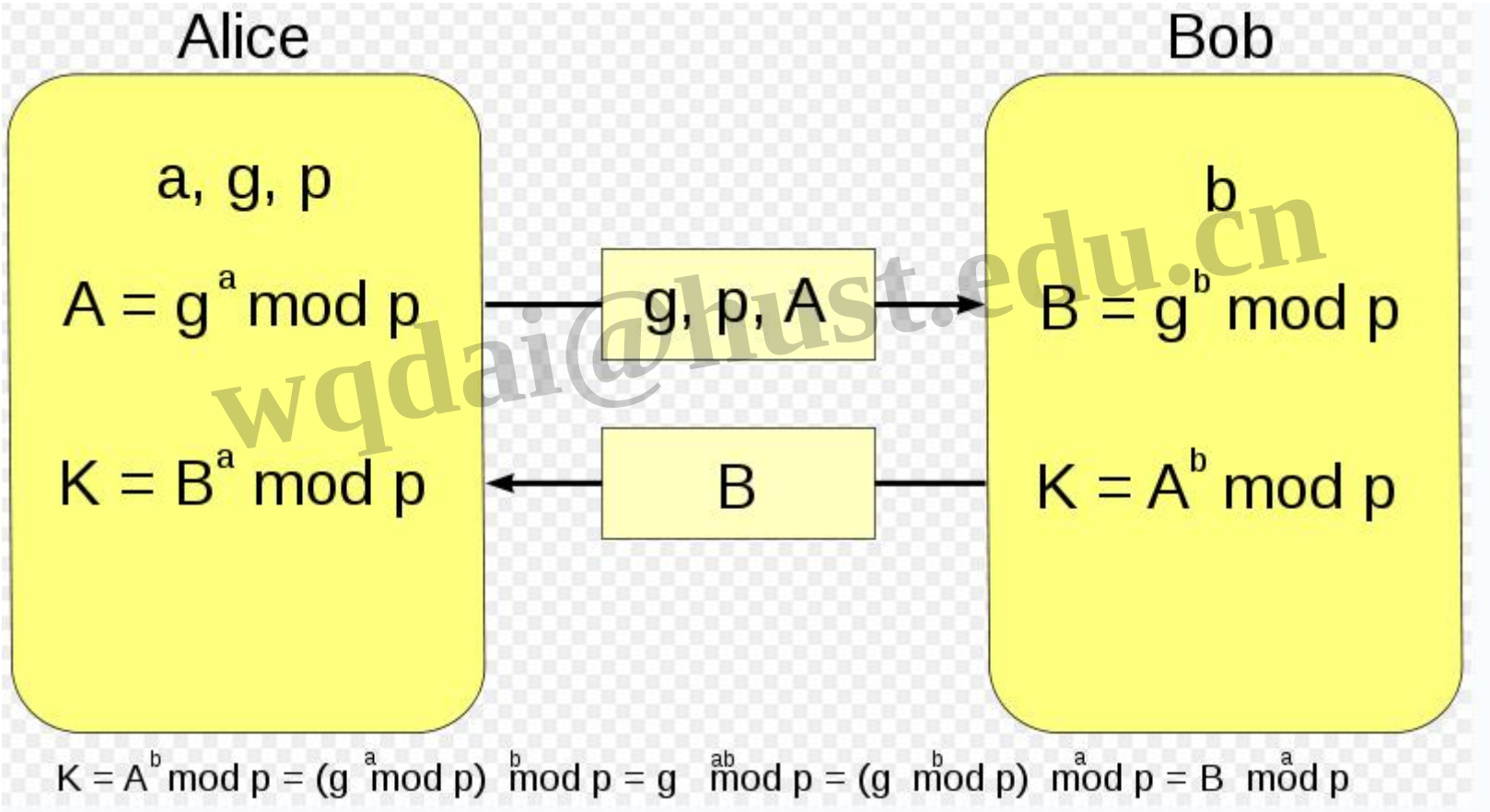
Asymmetric Encryption

- Asymmetric encryption is not normally be used for directly encrypting data due to the computation consumption.
- Encryption: for creating a symmetric key which then is used for encryption
- Signature: for signing the digest of data

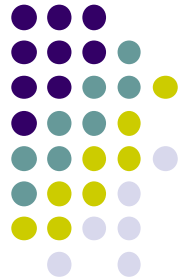


In the Diffie-Hellman key exchange scheme, each party generates a public/private key pair and distributes the public key. After obtaining an authentic copy of each other's public keys, Alice and Bob can compute a shared secret offline. The shared secret can be used as the key for a symmetric cipher.

Diffie-Hellman key exchange protocol

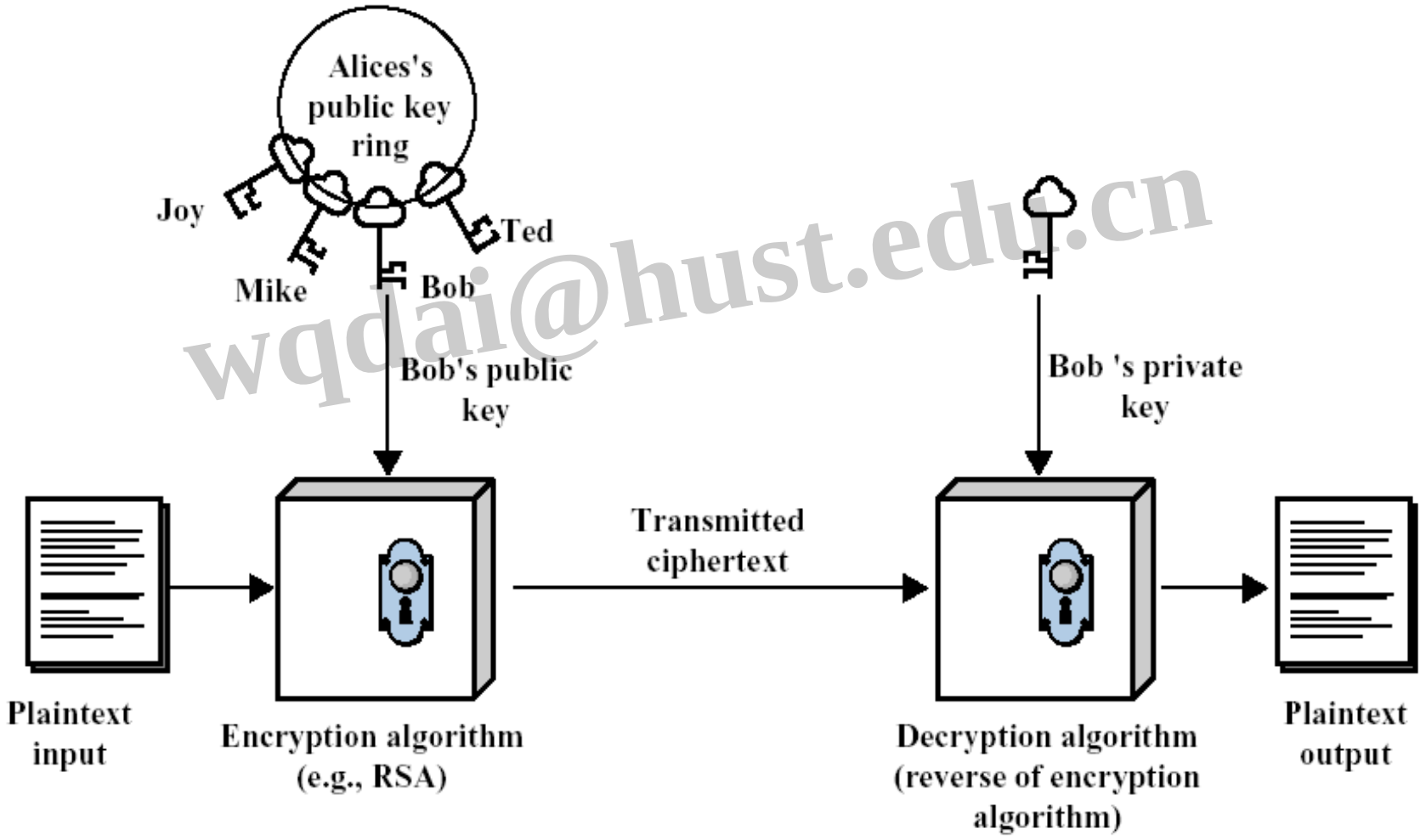


Asymmetric Encryption

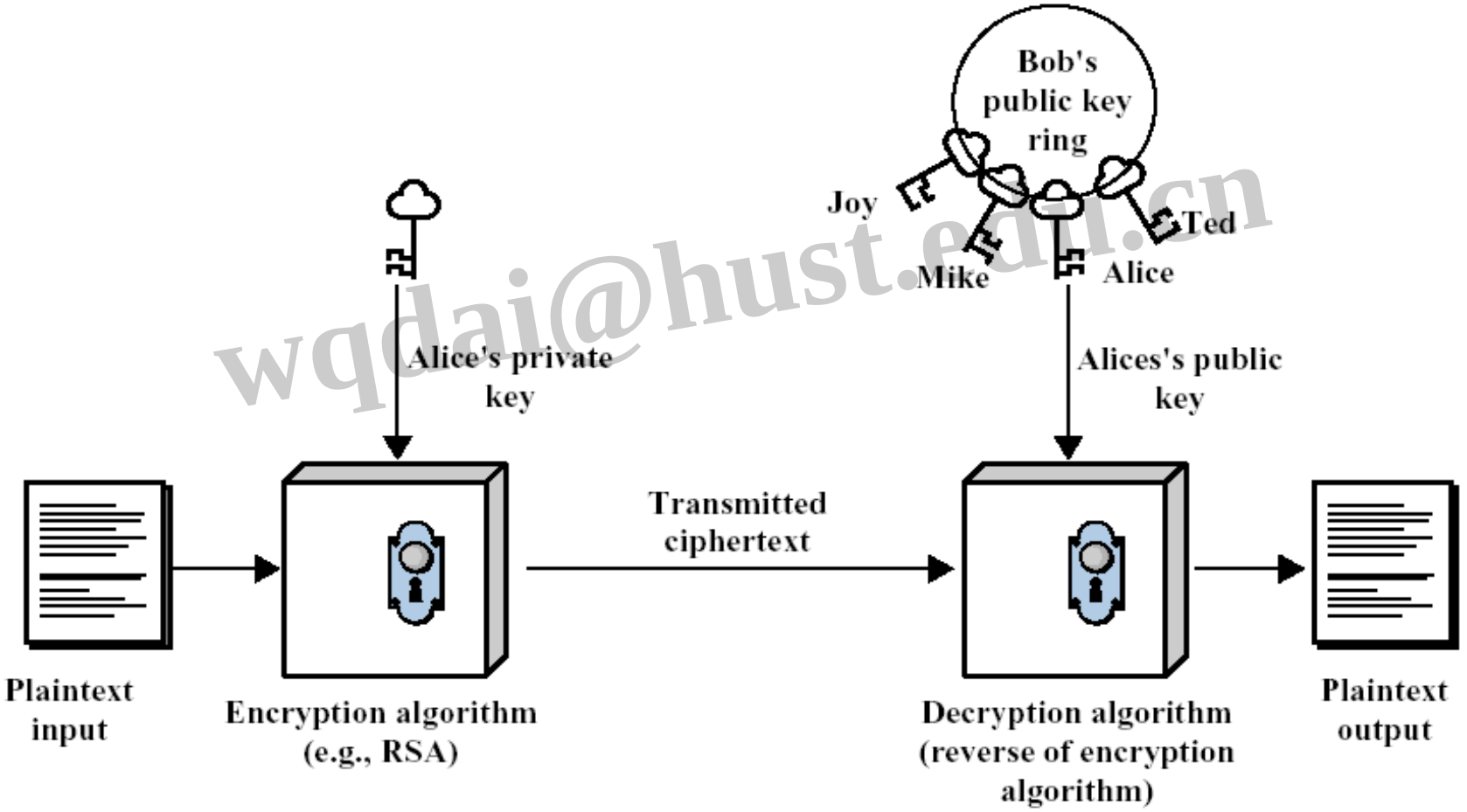


- The Integer Factorization Problem, RSA
- The Discrete Logarithm Problem, ElGamal
- The Elliptic Curve Discrete Logarithm Problem
- Knapsack problem
- 整数分解问题，RSA
- 离散对数问题，ElGamal
- 椭圆曲线离散对数问题
- 背包问题

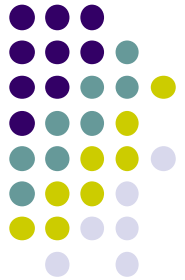
Asymmetric Encryption



Asymmetric Encryption

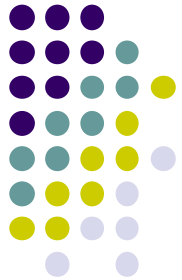


Some issues of RSA

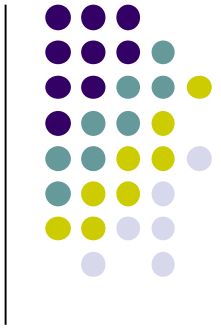


- Speed
- Key distribution
  - Digital signature
  - Components of PKI
- Etc.

Cryptographic hash

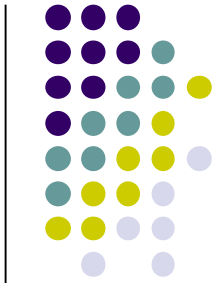


- Takes some input and creates a fixed-size output
- Some important properties
  - Given the hash output, you have no way to determine the input to the hash
  - Changing one bit of the input causes at least 50 percent of the output bits to change
  - Input to the hash is order-dependent. Hashing A then B does not produce the same value as hashing B then A
- Provides the “identity” of an entity (some piece of software)
- MD5,SHA-1, etc.



- SMS4—a Chinese symmetric encrypt/decrypt algorithm
  - 128 bits, CBC (Cipher Block Chaining) mode
- SM2—a Chinese asymmetric encrypt/decrypt algorithm (ECC- Elliptic Curve Cryptography)
- SM3—a Chinese hash algorithm

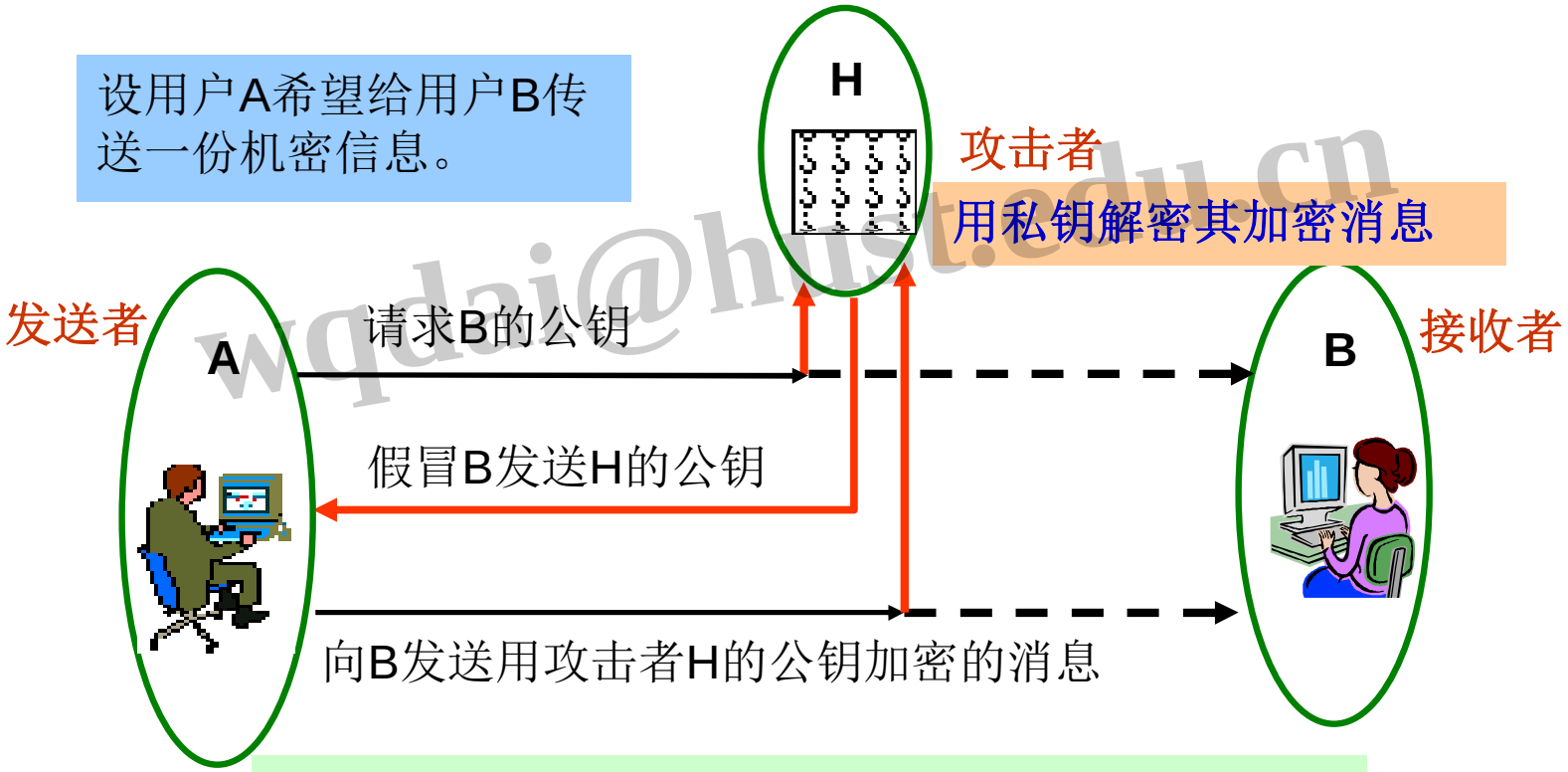
公开密钥基础设施PKI



- PKI的动机
- PKI提供的基本服务
- PKI安全服务特点
- PKI的体系结构和组成
- PKI的X.509证书
- PKI的实际应用中特别关注的问题
- PKI密钥的使用周期
- PKI的核心—CA系统
- PKI的应用
- TLS/SSL

公钥基础设施PKI——问题的提出

怎样才能知道任意一个公开密钥是属于谁的？如果收到一个自称是某人的公开密钥，能相信它吗？

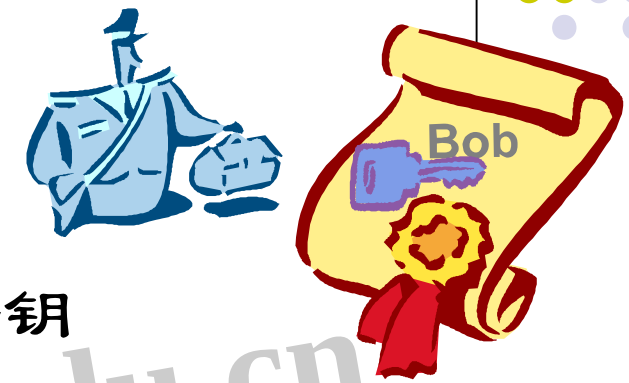


问题所在：对公钥缺乏鉴别机制！

PKI之动机

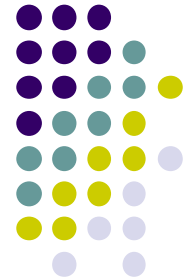
- 公钥技术
  - 签名：利用公私钥对天然的不对称性，使用数字签名可以进行认证、防抵赖和消息完整性鉴别等服务，但是，依赖方如何相信该签名者的身份和签名所使用的私钥的切实对应关系？
  - 加密
  - 公私钥对和身份如何建立联系
    - 为什么要相信这确实是某个人的公钥
  - 公钥如何管理
- 方案：引入证书(certificate)
  - 通过证书把公钥和身份关联起来

为什么需要证书？



- 安全服务依赖于：
  - 自己拥有秘密密钥
  - 其伙伴是否拥有你的公开密钥
- 通信伙伴采用你的公钥来核实你是否拥有秘密密钥
- 问题：通信伙伴如何能肯定所使用的公钥属于你？
- 解决：让第三方即 CA 对公钥进行公证，公证后的公钥就是证书。

公开密钥基础设施PKI

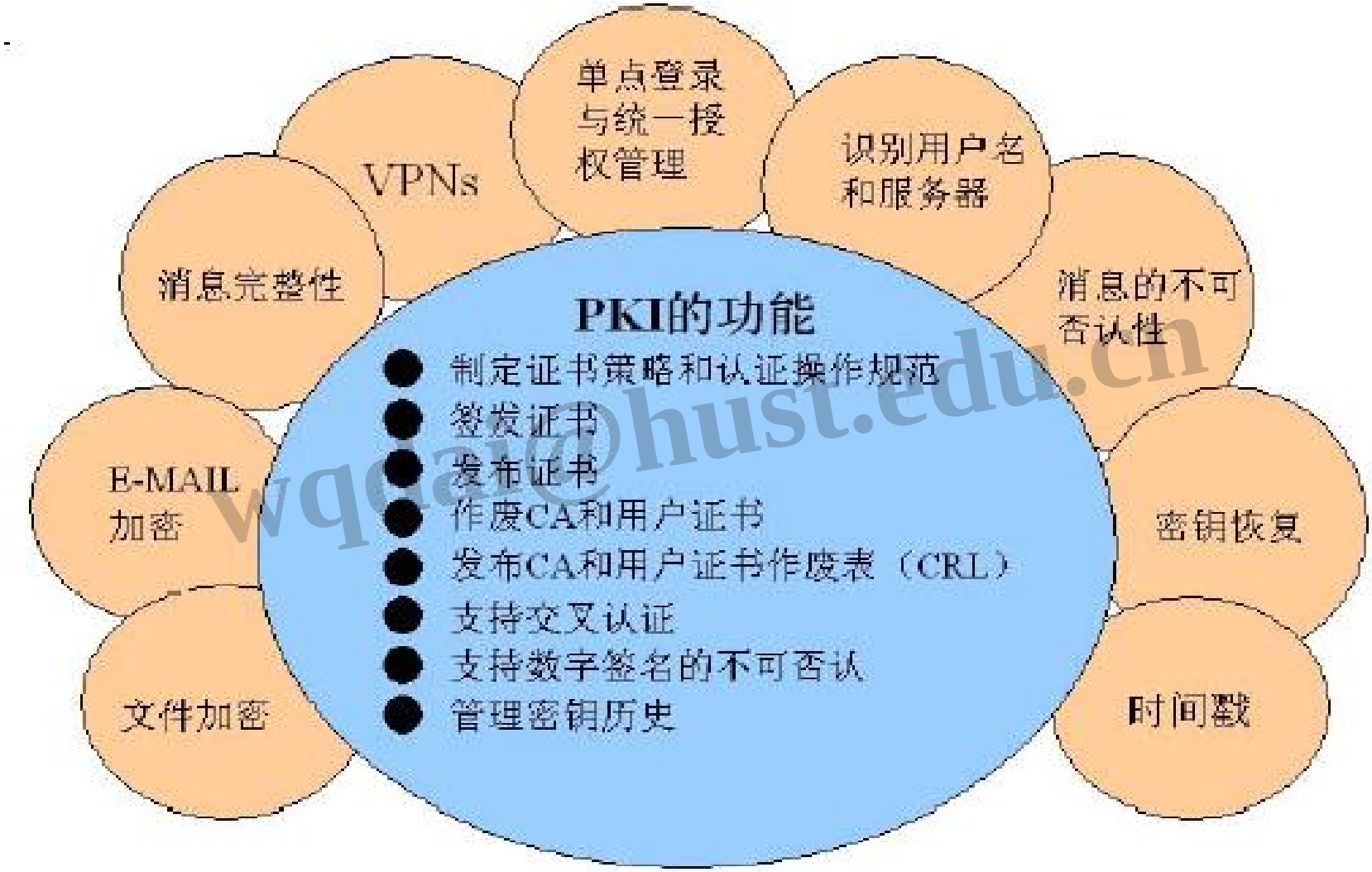


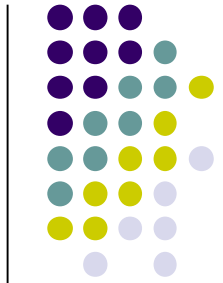
- PKI（Public Key Infrastructure）采用证书进行公钥管理，通过第三方的可信任机构（认证中心，即CA），把用户的公钥和用户的其他标识信息捆绑在一起，其中包括用户名和电子邮件地址等信息，以在Internet网上验证用户的身份。PKI把公钥密码和对称密码结合起来，在Internet网上实现密钥的自动管理，保证网上数据的安全传输。

公开密钥基础设施PKI

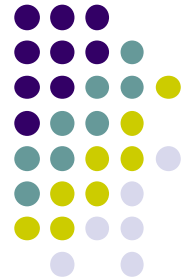
- 公钥基础设施（Public Key Infrastructure，PKI）是建立在公钥密码体制上的信息安全基础设施，为网络应用提供身份认证、加密、数字签名、时间戳等安全服务。
- 因此，PKI是一个使用公钥密码技术来实施和提供安全服务的、具有普适性的安全基础设施的总称，而并不特指某一具体密码设备及其管理设备。
- 实际上，PKI是生成、管理、存储、分发和撤销基于公开密码的公钥证书所需要的硬件、软件、人员、策略和规程的总和。
- PKI已广泛用于保障电子商务和电子政务的安全。

PKI是信息安全的基础设施之一



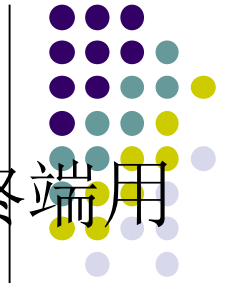


PKI提供的基本服务



- 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
    - 远程设备生成的challenge信息 —— 身份认证
- 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
- 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

PKI安全服务特点－终端用户透明性



一般基础设施具有一个很重要的特点，就是对终端用户的操作来说几乎完全是透明的。

基础设施应该提供的是一个“黑盒子”级的服务。因此，所有的安全操作应当隐没在用户的后面，无需额外的干预，无需用户注意密钥和算法，不会因为用户的错误操作对安全造成危害。

终端用户的透明性操作，还应包括以下观点：安全不应该成为用户完成任务的障碍；安全操作无需用户具有特别的知识，无需用户进行特殊的处理，不会严重增加用户的负担；除了初始的登录操作之外，基础设施应当用一种对用户安全透明的方式完成所有与安全有关的工作。

PKI安全服务特点－全面的安全性

作为一个公钥基础设施最大的益处是在整个应用环境中，使用单一可信的安全技术——公钥技术。它能保证数目不受限制的应用程序、设备和服务器无缝地协调工作，安全地传输、存储和检索数据，安全地进行事务处理，安全地访问服务器等。电子邮件应用、Web访问、防火墙、远程访问设备、应用服务器、文件服务器、数据库或更多的其他设备，能够用一种统一的方式理解和使用安全服务基础设施。在这种环境中，不仅极大地简化了终端用户使用各种设备和应用程序的方式，而且简化了设备和应用系统的管理工作，保证执行相同等级的安全策略

PKI安全服务特点—开放和互操作性

- **互操作性**

在一个企业内部，实施多个点对点的解决方案，无法实现互操作性。因为这些方案是独立开发的，具有互不兼容的操作流程和工具平台。相反，安全基础设施具有很好的互操作性，因为每个应用程序和设备以相同的方式访问和使用基础设施。

- **开放性**

PKI系统的设计具有开放型的特点。任何先进技术的早期设计，都希望在将来能和其他企业间实现互操作。一个基于开放的、国际标准公认的基础设施技术比一个专有的点对点的技术方案更可信和方便。

- **一致的解决方案**

安全基础设施为所有的应用程序和设备提供了可靠的、一致的解决方案。与一系列互不兼容的解决方案相比，这种一致性的解决方案在一个企业内更易于安装、管理和维护。一致性解决方案使管理开销小和简单，这是基础设施的重要优势。

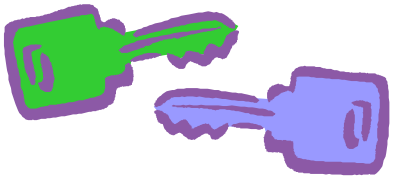
密钥对的用法

- 用于加密的密钥对

用公钥加密

用私钥解密

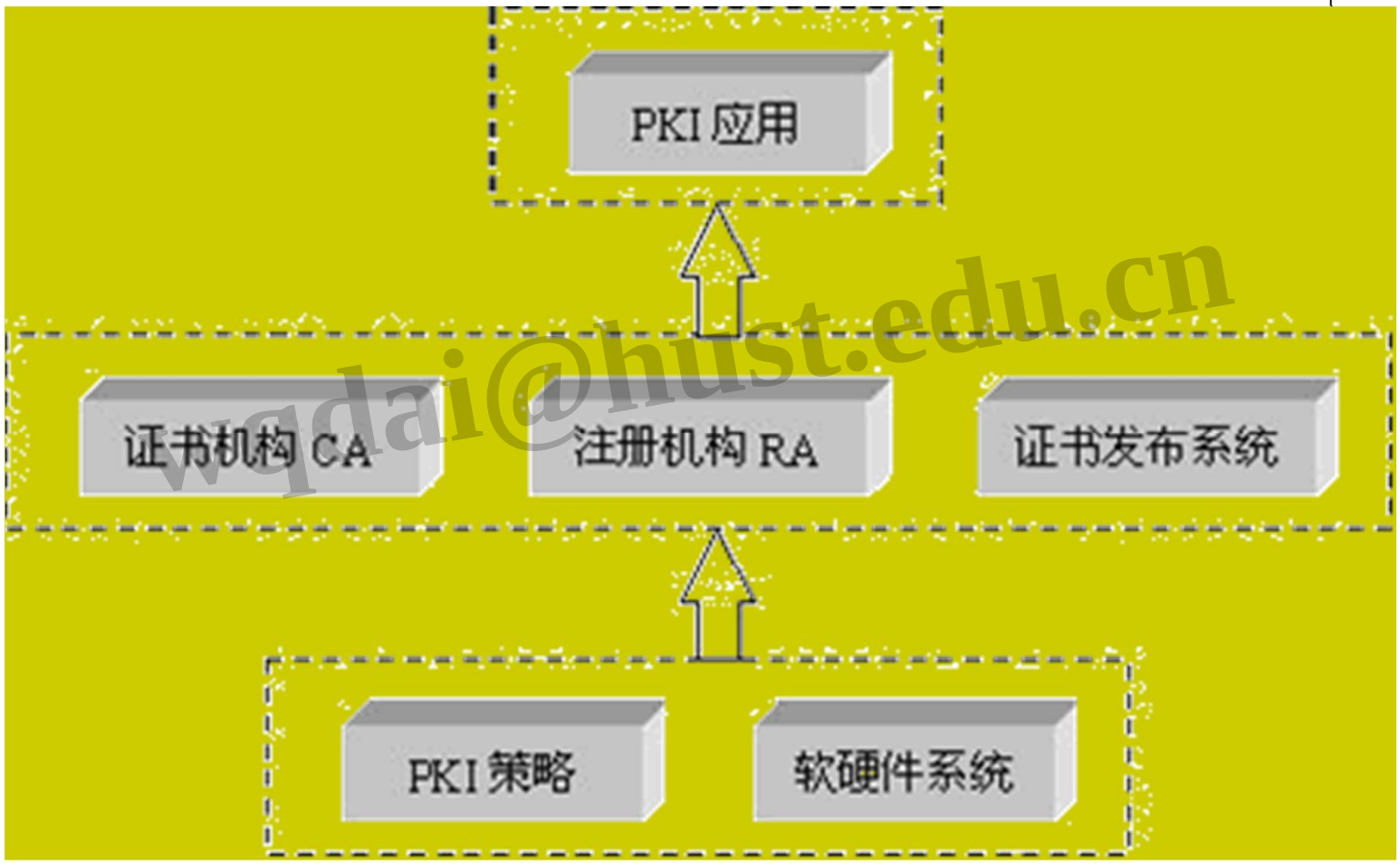
- 用于签名的密钥对



用私钥签名

用公钥验证

PKI的基本框图

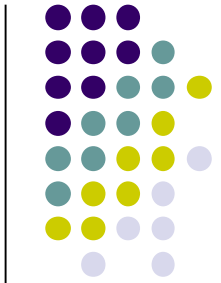


PKI基本组成

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）

PKI基本组成



公钥证书

由可信实体签名的电子记录，记录将公钥和密钥（公私钥对）所有者的身份捆绑在一起。公钥证书是PKI的基本部件。

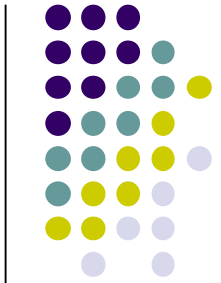
证书作废列表（CRL）

作废证书列单，通常由同一个发证实体签名。当公钥的所有者丢失私钥，或者改换姓名时，需要将原有证书作废。

策略管理机构（PMA）

监督证书策略的产生和更新，管理PKI证书策略。

PKI基本组成



注册机构（RA）

互联网定义： 一个可选PKI实体（与CA分开），不对数字证书或证书作废列单（CRL）签名，而负责记录和验证部分或所有有关信息（特别是主体的身份），这些信息用于CA发行证书和CLR以及证书管理中。RA在当地可设置分支机构LRA。

PKIX用语： 一个可选PKI实体与CA分开，RA的功能随情况而不同，但是可以包括身份认证和用户名分配，密钥生成和密钥对归档，密码模件分发及作废报告管理。

国防部定义： 对CA负责当地用户身份（**标识**）识别的人。

PKI基本组成

认证机构（CA）

互联网定义： 一个可信实体，发放和作废公钥证书，并对各作废证书列表签名。

国防部定义： 一个授权产生，签名，发放公钥证书的实体。CA全面负责证书发行和管理（即，注册进程控制，身份标识和认证进程，证书制造进程，证书公布和作废及密钥的更换）。CA还全面负责CA服务和CA运行。

联邦政府定义： 被一个或多个用户所信任发放和管理X.509公钥证书和作废证书的机构。

PKI基本组成

证书管理机构（CMA）

将CA和RA合起来称CMA(certificate management authority)。

证书存档(Repository)

一个电子站点，存放证书和作废证书列表(CRL)，CA在用证书和作废证书。

署名用户（Subscriber）

署名用户是作为主体署名证书并依据策略使用证书和相应密钥的实体。

PKI基本组成

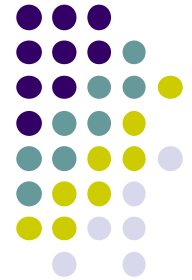
依赖方(Relying party)

一个接收包括证书和签名信息的人或机构，利用证书提供的公钥验证其有效性，与持证人建立保密通信，接收方处于依赖的地位。

最终用户 (End User)

署名用户和依赖方的统称，也称末端实体(End-entity)，可以是人，也可以是机器，如路由器，或计算机中运行的进程，如防火墙。

PKI中的证书



- 证书(certificate)，有时候简称为cert
- PKI适用于异构环境中，所以证书的格式在所使用的范围内必须统一
- 证书是一个机构颁发给一个安全个体的证明，所以证书的权威性取决于该机构的权威性
- 一个证书中，最重要的信息是个体名字、个体的公钥、机构的签名、算法和用途
- 签名证书和加密证书分开
- 最常用的证书格式为X.509 v3

PKI中的证书

- 数字证书是网络用户的身份证明，相当于现实生活中的个人身份证。
- 数字证书提供了一种系统化的、可扩展的、统一的、容易控制的公钥分发方法。是一个防篡改的数据集合，它可以证实一个公开密钥与某一最终用户之间的捆绑。为了提供这种捆绑关系，需要一组可信第三方实体来担保用户的身份。
- 第三方实体称为证书颁发机构CA，它向用户颁发数字证书，证书中含有用户名、公开密钥以及其他身份信息，并由证书颁发机构对之进行了数字签名，即证书的拥有者是被证书机构所信任的。

PKI中的证书

- 若信任证书机构 ⇒ 信任证书拥有者

  所以，通过证书，可以使证书的拥有者向系统中的其它实体证明自己的身份。

- 在大型网络中，这样的证书颁发机构(CA)有多个。如果这些CA之间存在信赖关系，则用户就可通过一个签名链去设法认证其中任一CA所签发的证书。

数字证书基本内容：

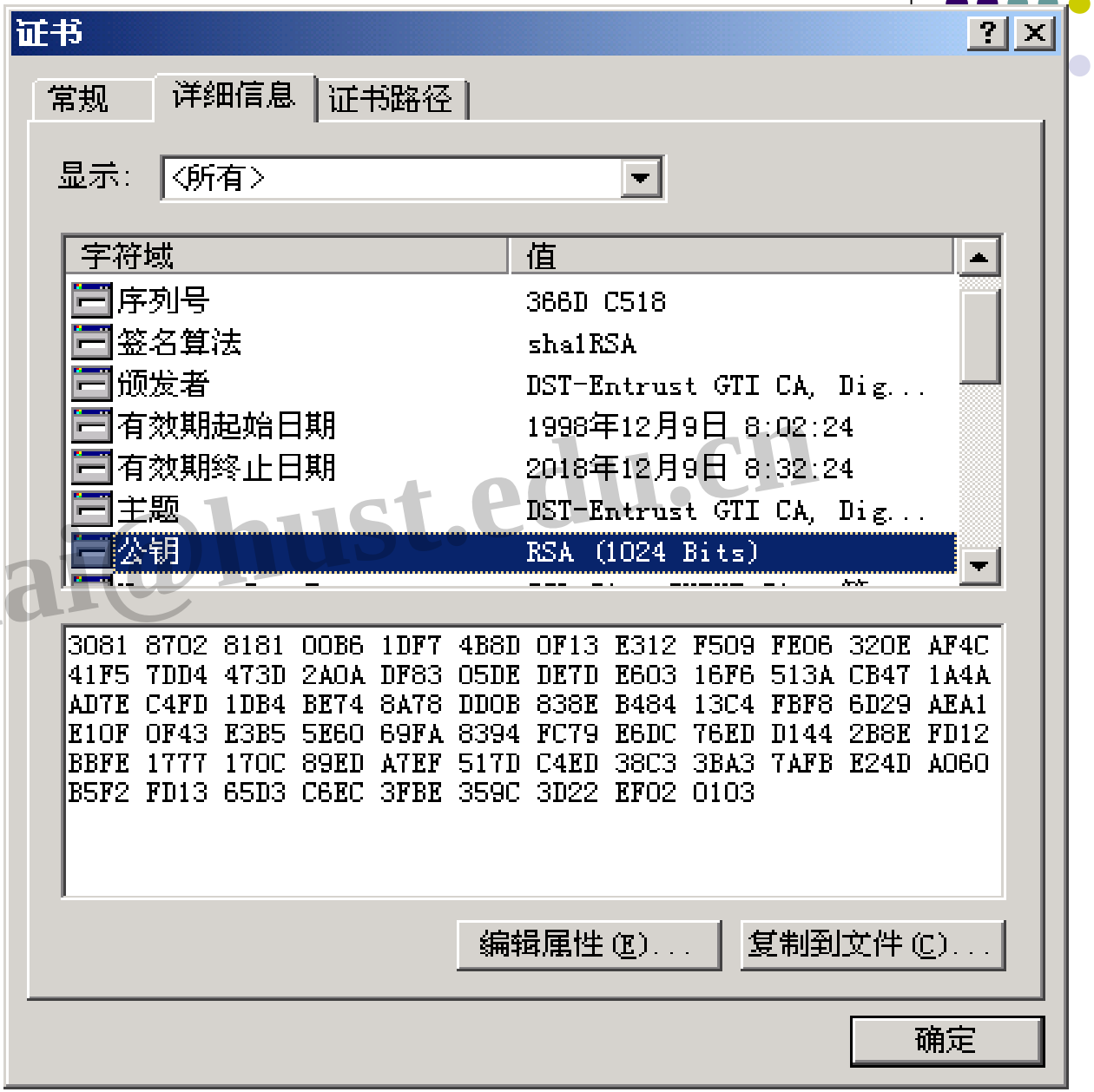
用户标识＋用户公钥及其它信息＋CA签名

PKI中的证书

基于PKI的数字证书将公钥与其用户的身份捆绑在一起，证书必须要有一定的标准格式。目前广泛采用的证书格式是国际电信联盟（ITU）提出的X.509v3格式

内容	说明
版本V	X．509版本号
证书序列号	用于标识证书
算法标识符	签名证书的算法标识符
参数	算法规定的参数
颁发者	证书颁发者的名称及标识符(X.500)
起始时间	证书的有效期
终止时间	证书的有效期
持证者	证书持有者的姓名及标识符
算法	证书的公钥算法
参数	证书的公钥参数
持证书人公钥	证书的公钥
扩展部分	CA对该证书的附加信息，如密钥的用途
数字签名	证书所有数据经H运行后CA用私钥签名

数字证书实例

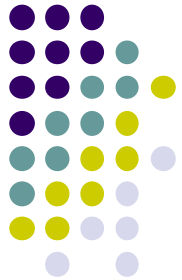


X.509 Certificate Format

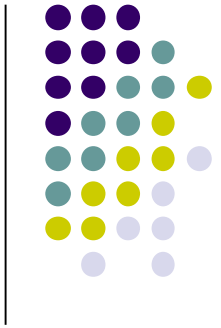
- Content of certificate
  - Subject's DN
  - CA's DN
  - Expiration time
  - Extension
  - Signature
  - Etc.

```
openssl x509 -in usercert.pem -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2937 (0xb79)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: O=EDU, O=HUST, CN=Certification Authority
        Validity
            Not Before: Apr 16 09:57:51 2012 GMT
            Not After : Apr 16 09:57:51 2013 GMT
        Subject: O=EDU, OU=HUST, CN=TestUser
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:b0:54:9b:0c:f0:f5:24:78:15:52:45:75:9b:9a:
                    5c:12:bc:8c:70:38:cb:c3:9f:0d:02:6c:ce:24:77:
                    d1:01:64:18:e3:69:6f:d3:4f:00:5f:2a:24:ee:2f:
                    23:8d:49:0d:c1:57:d0:0f:9d:d5:97:3e:8a:8d:cb:
                    71:7b:71:96:ec:03:d2:1e:21:a9:4e:8b:42:ad:56:
                    d3:55:92:dc:9b:b2:91:06:25:94:72:14:df:a4:a6:
                    b6:cb:24:aa:04:56:23:81:f2:ad:43:1b:4a:ef:d3:
                    dc:d6:03:b9:fa:a9:11:49:70:aa:b3:78:fc:ec:2b:
                    2f:19:a3:f6:cb:dd:b2:3c:e5
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Cert Type:
                SSL Client, S/MIME
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                7E:49:96:EB:2A:28:6B:3F:08:0F:1E:9C:77:60:4A:ED:15:2B:0E:37
            X509v3 Authority Key Identifier:
                keyid:18:05:C0:FC:0B:D1:B7:3A:F4:65:92:09:FB:59:A1:5F:C7:88:C4:F0
                DirName:/O=Grid/O=NorduGrid/CN=NorduGrid Certification Authority
                serial:00

            X509v3 Subject Alternative Name:
                email:testuser@hust.edu.cn
    Signature Algorithm: sha1WithRSAEncryption
        18:22:87:57:70:21:2d:85:0d:d3:25:cc:ef:e9:b9:c6:23:bb:
        ce:ec:f1:5d:71:71:5c:71:8e:1e:e3:61:e4:9b:3a:09:9c:de:
        b5:9f:a2:12:a4:92:fa:26:b6:69:c9:db:60:1e:d2:67:cd:fe:
        c3:5a:65:5c:31:3b:cd:97:35:e8:f9:96:86:fc:c6:9f:a2:ae:
        86:0c:f9:1a:ed:e9:99:e1:58:33:5a:32:e2:0a:4d:b1:40:fc:
        78:1f:9d:37:19:8e:b3:e6:e6:db:4d:62:8a:db:a8:30:ab:ee:
        3f:da:6f:4c:64:69:3c:10:61:58:e7:25:cd:25:ca:36:96:1a:
        c7:4b
```



- Version: 3 (0x2)
- Serial Number: 2937 (0xb79)
- Signature Algorithm: sha1WithRSAEncryption
- Issuer: O=EDU, O=HUST, CN=Certification Authority
- Validity
  - Not Before: Apr 16 09:57:51 2010 GMT
  - Not After : Apr 16 09:57:51 2011 GMT
- Subject: O=EDU, OU=HUST, CN=TestUser
- Subject Public Key Info



- X509v3 extensions:
  - X509v3 Basic Constraints:
  - CA:FALSE
  - Netscape Cert Type:
  - SSL Client, S/MIME
  - X509v3 Key Usage:
  - Digital Signature, Non Repudiation, Key Encipherment
  - Netscape Comment:
  - OpenSSL Generated Certificate
  - X509v3 Subject Key Identifier:
  - 7E:49:96:EB:2A:28:6B:3F:08:0F:1E:9C:77:60:4A:ED:15:2B:0E:37
  - X509v3 Authority Key Identifier:
  - keyid:18:05:C0:FC:0B:D1:B7:3A:F4:65:92:09:FB:59:A1:5F:C7:88:C4:F0
  - DirName:/O=EDU/O=HUST/CN=Certification Authority
  - serial:00

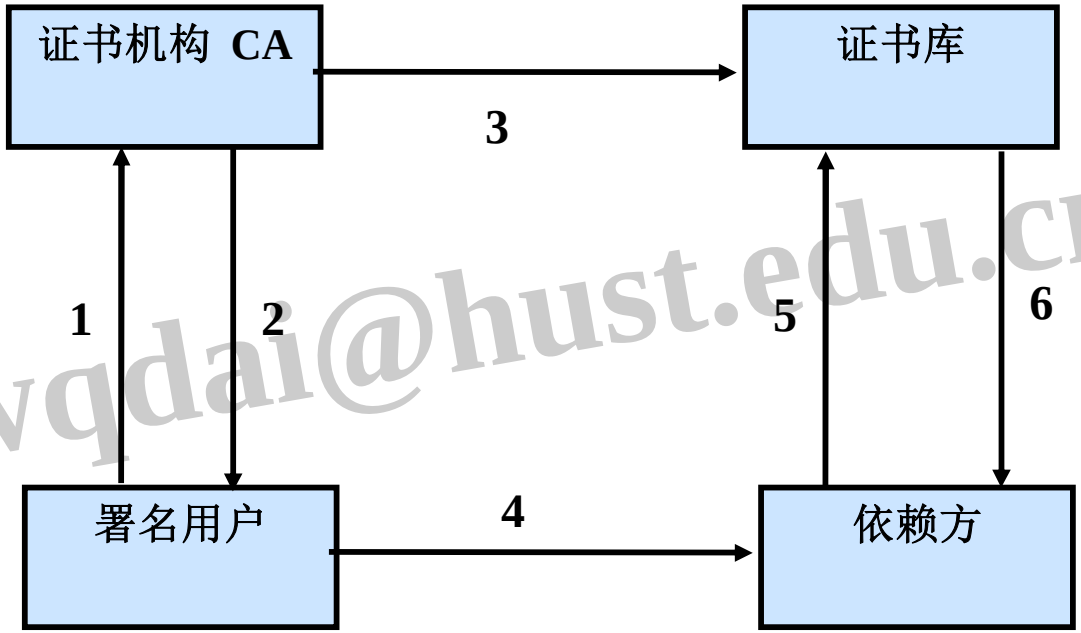
  - X509v3 Subject Alternative Name:
  - email:testuser@hust.edu.cn

PKI的运行

X509标准PKIX

1）署名用户向证明机构（CA）提出数字证书申请；
2）CA验明署名用户身份，并签发数字证书；
3）CA将证书公布到证书库中；
4）署名用户对电子信件数字签名作为发送认证，确保信件完整性，不可否认性，并发送给依赖方。
5）依赖方接收信件，用署名用户的公钥验证数字签名，并到证书库查明署名用户证书的状态和有效性；
6）证书库返回证书检查结果；

PKI的运行



PKI的实际应用考虑—i

- 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
- 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
- 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的实际应用考虑－ii

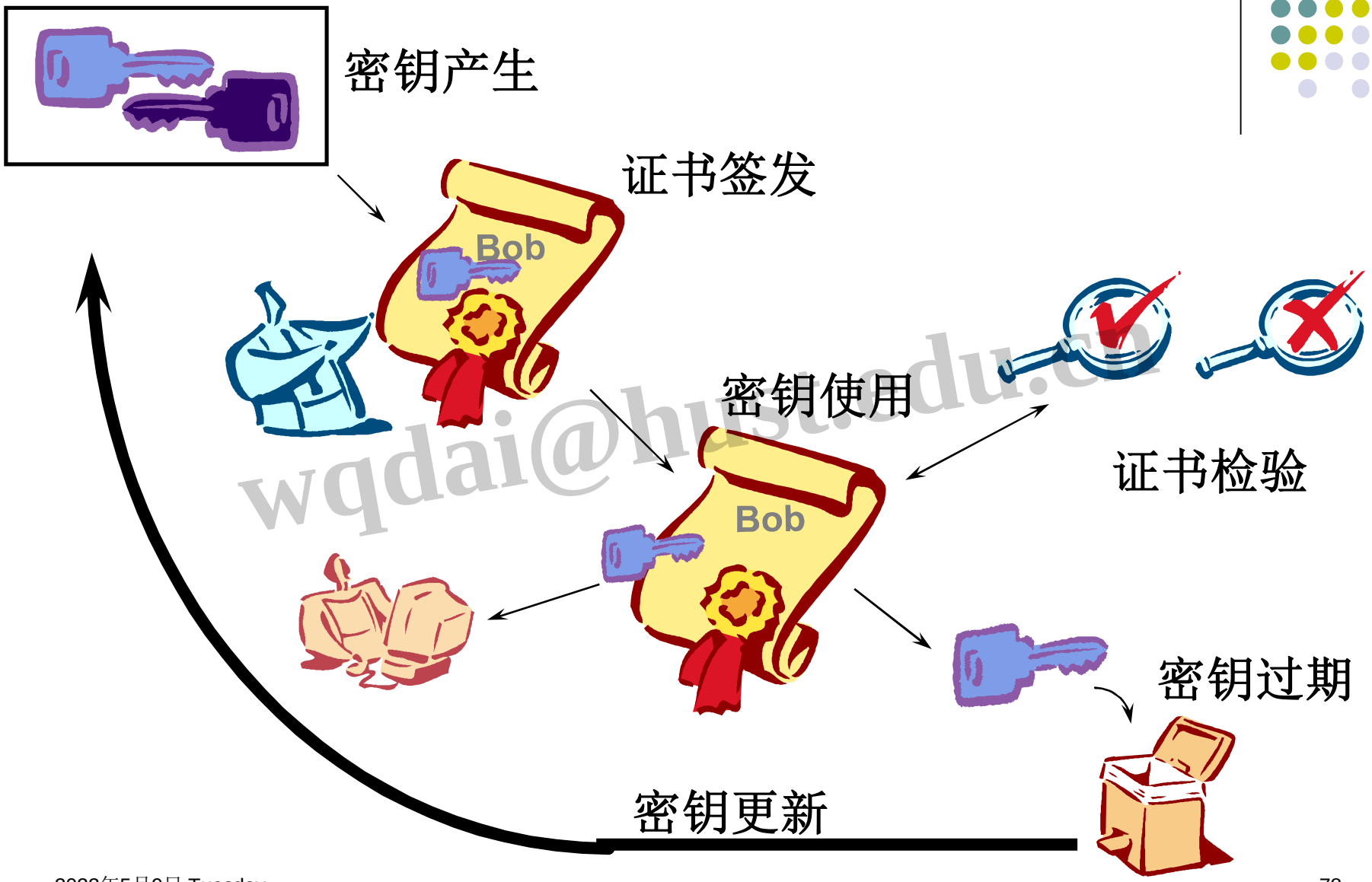
- 可靠稳定的安全运营设施
  - 完善的人员安全管理
  - **物理安全：**特殊的消防、监控及入侵检测设施，工作人员进入安全服务器区域都采用电子（智能卡）及生物（指纹）验证，并制定了双进双出等安全策略
  - **逻辑安全逻辑：**包括对整个内部网的网络逻辑安全、主机安全、应用安全和数据库安全，对整个PKI/CA系统逻辑进行深入分析和部署
  - **通信安全：**提供完善的访问控制、授权管理，保证了所有网络通信中的信息安全
  - **密钥管理安全：**硬件加密设备将支持标准的加解密算法、保证加解密速度和密钥存储容量，并且具有安全的操作管理系统。此外，加密设备符合有关硬件加密设备的安全性要求，如开箱自毁、防暴力攻击、防辐射、严格的访问控制等
  - **审计安全：**PKI/CA系统在运行过程中涉及大量功能模块之间的相互调用，以及各种管理员的操作，因此使用功能模块调用日志、运行审计、数据中心管理员审计、RA管理员审计对这些活动以日志的形式进行记载，以方便系统错误分析、风险分析等

PKI中密钥和证书的管理

密钥/证书生命周期管理的各个阶段：

- 初始化阶段
- 颁发阶段
- 取消阶段
  - 证书过期
  - 证书撤销

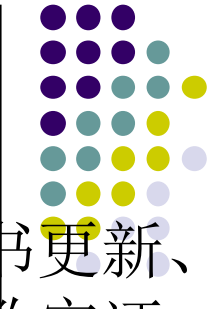
密钥生命周期



CA系统的主要内容

- CA实现的功能
- CA的结构和组件
- CA中密钥的管理
- CA的信任关系

CA(Certificate Authority)的功能



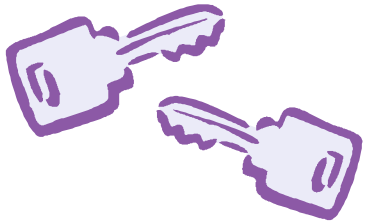
- 概括地说，认证中心（CA）的功能有：证书发放、证书更新、证书撤销和证书验证。CA的核心功能就是发放和管理数字证书，具体描述如下：

1. 接收验证最终用户数字证书的申请。
2. 确定是否接受最终用户数字证书的申请-证书的审批。
3. 向申请者颁发、拒绝颁发数字证书-证书的发放。
4. 接收、处理最终用户的数字证书更新请求-证书的更新。
5. 接收最终用户数字证书的查询、撤销。
6. 产生和发布证书废止列表（CRL）。
7. 数字证书的归档。
8. 密钥归档。
9. 历史数据归档。

CA系统的主要内容

- CA实现的功能
- CA的结构和组件
- CA中密钥的管理
- CA的信任关系

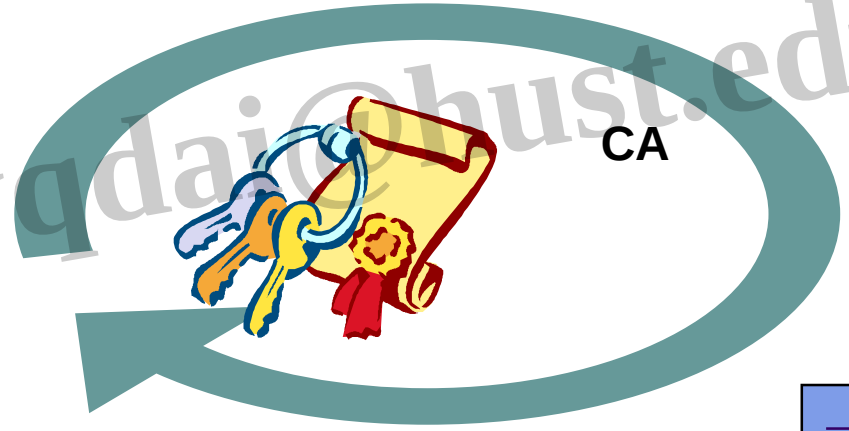
CA的基本框架—物理结构



密钥

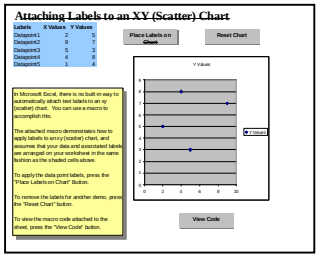
健壮的数据库系统

CA硬件

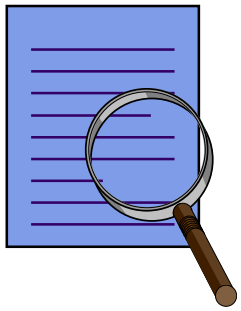


CA

无缝的目录接口



管理和运行平台



安全的审计

CA系统的主要内容

- CA实现的功能
- CA的结构和组件
- CA中密钥的管理
- CA的信任关系

CA的密钥管理功能

- 公私密钥对的产生
- 公钥的获取
- 密钥备份和恢复
- 证书撤消
- 自动更新密钥
- 密钥历史档案

公私密钥对的产生

- 1、用户生成

  在这种方式中，用户自己生成密钥对，然后将公钥以安全的方式传送给CA。

- 2、CA生成

这种方式是CA替用户生成密钥对，然后将其以安全的方式传送给用户。该过程必须确保密钥对的机密性、完整性和可验证性。该方式下由于用户的私钥为CA所产生，故对CA的可信性有更高的要求。如果是签名密钥，CA必须在事后销毁用户的私钥。

公钥的获取

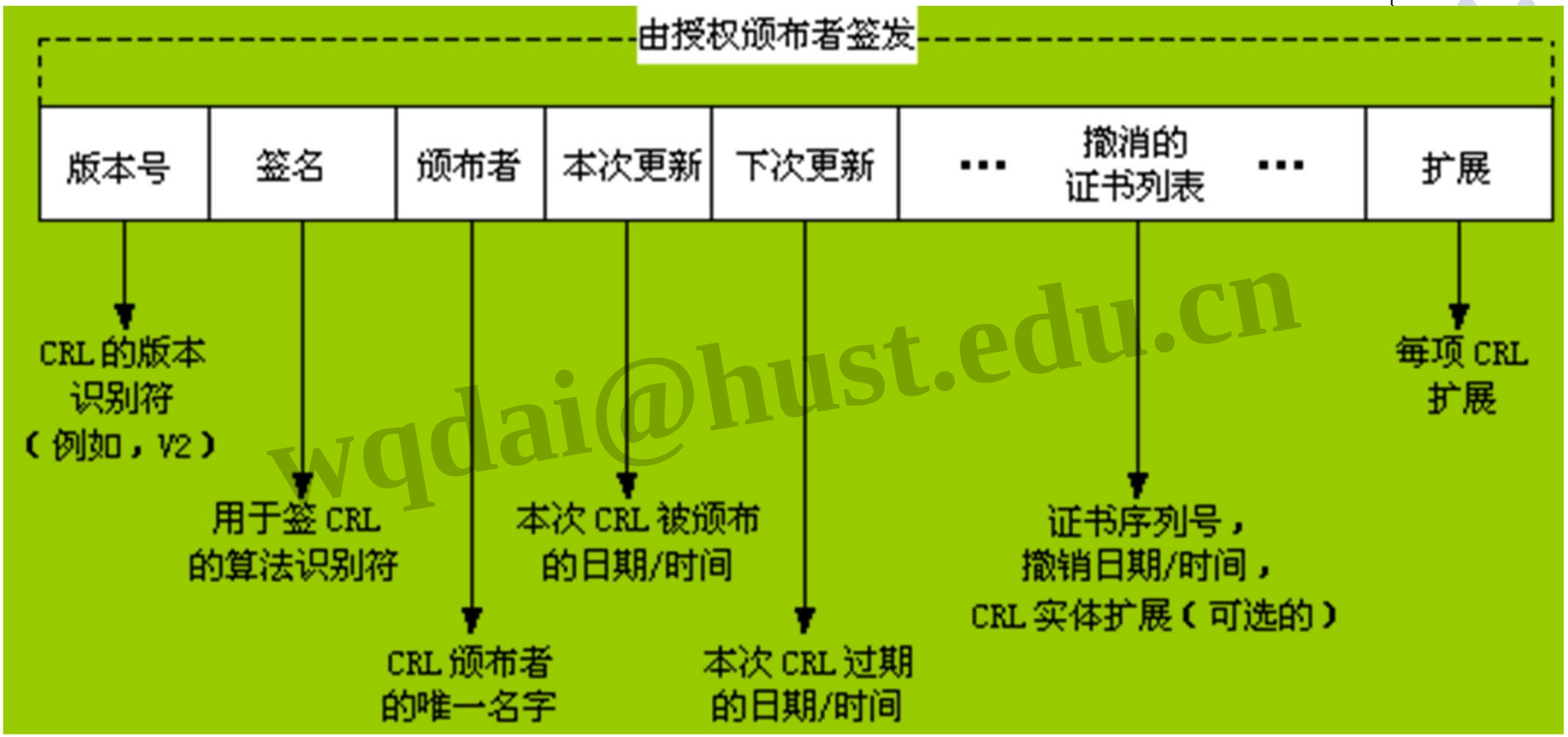
用户在网上可通过两种方式获取通信对方的公钥，以用来传送加密数据，实现安全通信。

- 由通信对方将自己的公钥随同发送的正文信息一起传送给用户。
- 所有的证书集中存放于一个证书库中，用户在网上可从该地点取得通信对方的证书

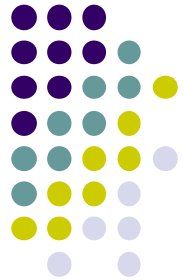
证书撤销—i

- 认证机构CA签发证书来为用户的身份和公钥进行捆绑，可是因种种原因，还必须存在一种机制来撤消这种捆绑关系，将现行的证书撤消。
- 这种撤消的原因通常有：用户身份姓名的改变、私钥被窃或泄露、用户与其所属企业关系变更等。这样就必须存在一种方法警告其他用户不要再使用这个公钥。在PKI中，这种警告机制，被称作证书撤销。而使用的手段就是前面所提到的证书撤销列表（CRL）。
- 证书撤消列表的结构是按X.509证书标准定义的，其结构如图所示

证书撤销—ii



证书撤销—iii



- CRL
- OCSP，Online Certificate Status Protocol,在线证书状态协议

CA系统的主要内容

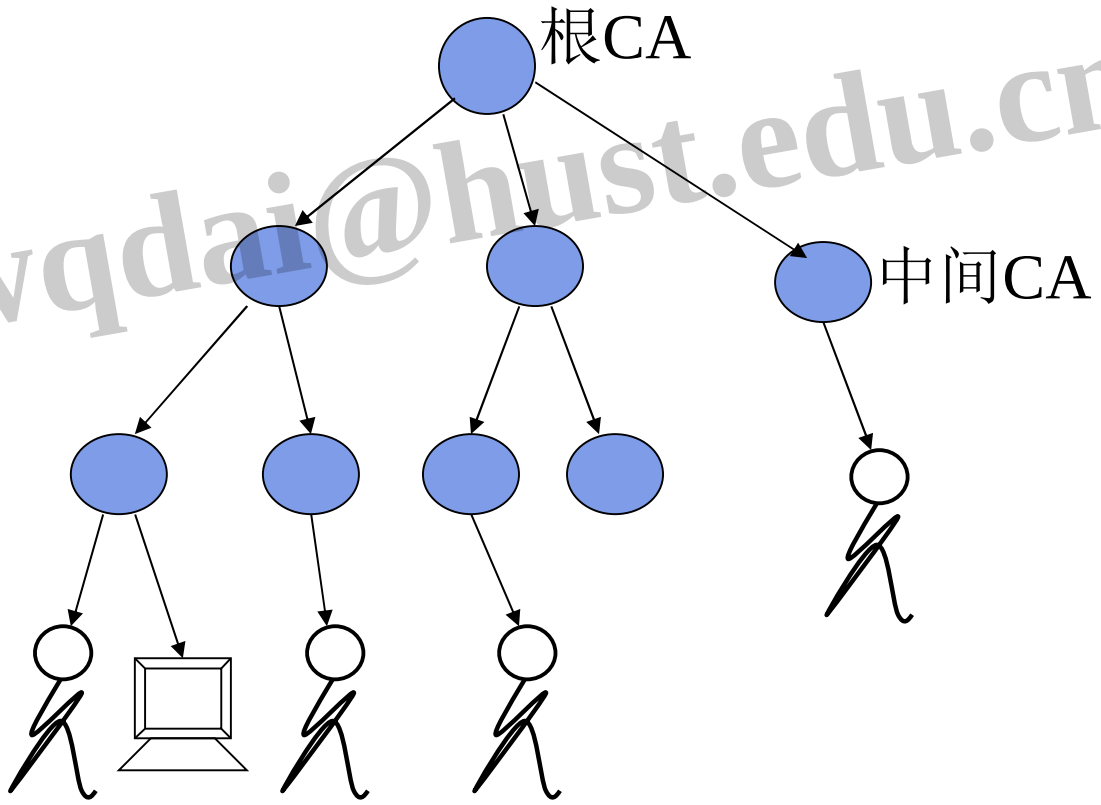
- CA实现的功能
- CA的结构和组件
- CA中密钥的管理
- CA的信任关系

CA信任关系

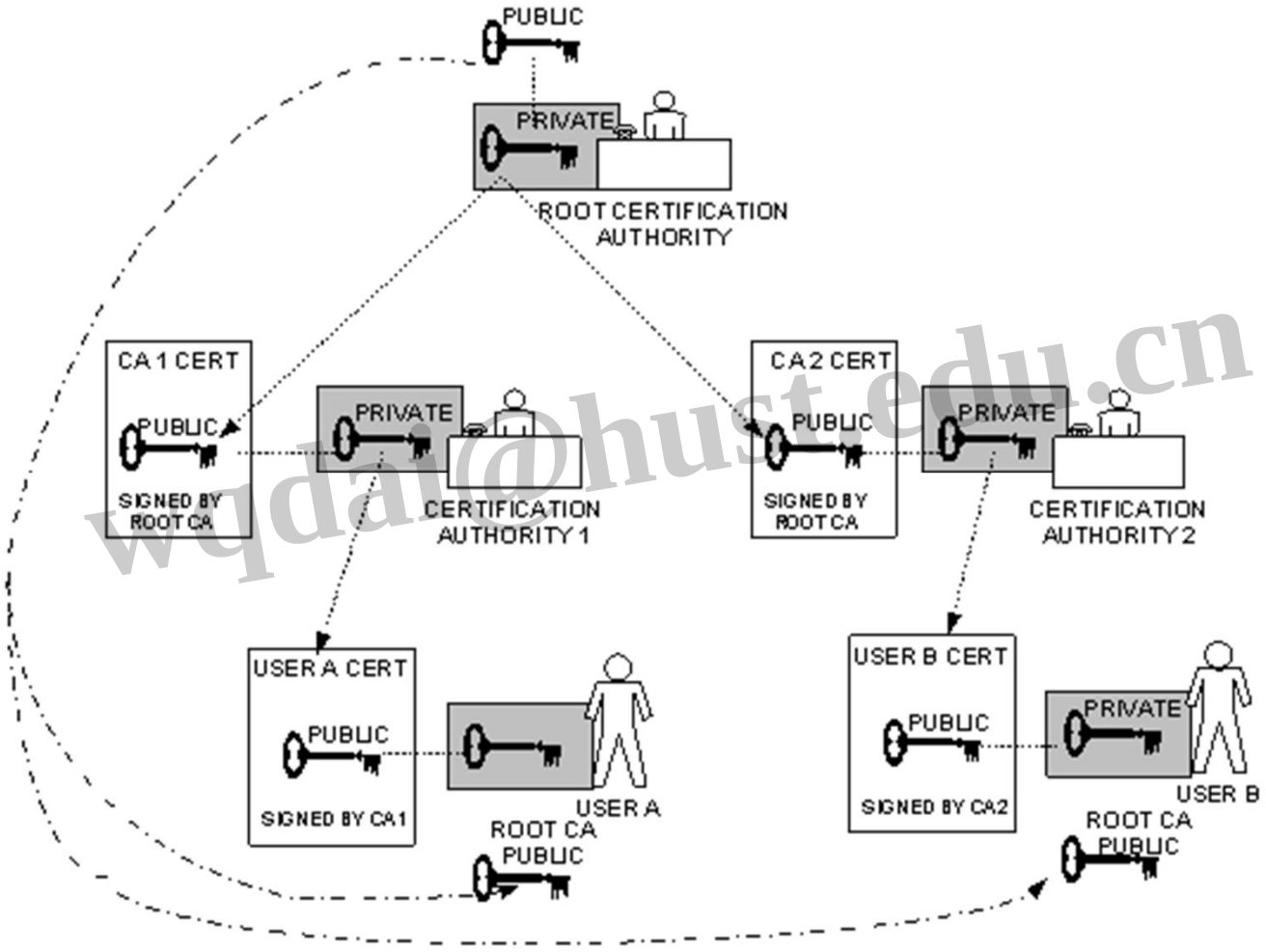
- 当一个安全个体看到另一个安全个体出示的证书时，他是否信任此证书？
  - 信任难以度量，总是与风险联系在一起
- 可信CA
  - 如果一个个体假设CA能够建立并维持一个准确的“个体-公钥属性”之间的绑定，则他可以信任该CA，该CA为可信CA
- 信任模型
  - 基于层次结构的信任模型
  - 交叉认证
  - 以用户为中心的信任模型
  - 浏览器信任列表认证模型

CA层次结构

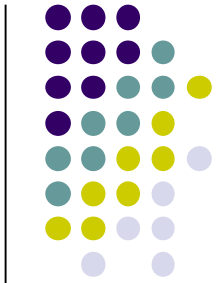
- 对于一个运行CA的大型权威机构而言，签发证书的工作不能仅仅由一个CA来完成
- 它可以建立一个CA层次结构



CA层次结构



CA层次结构的建立

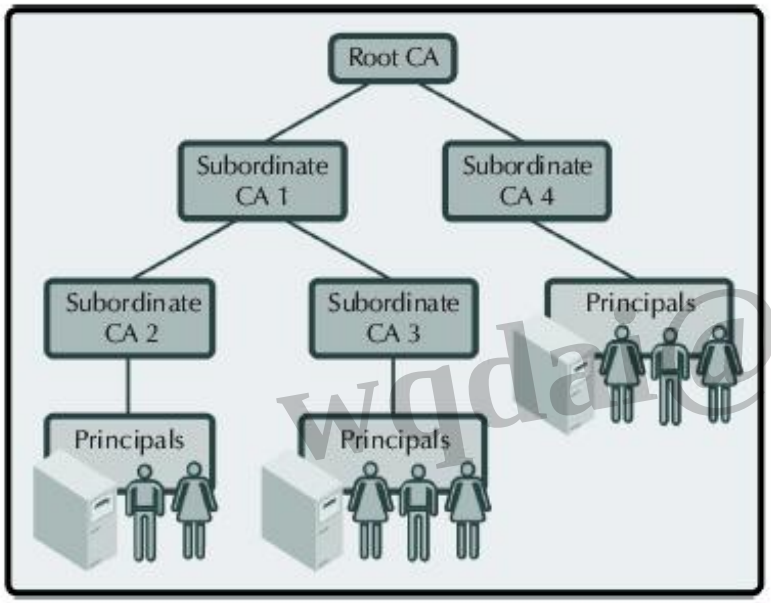


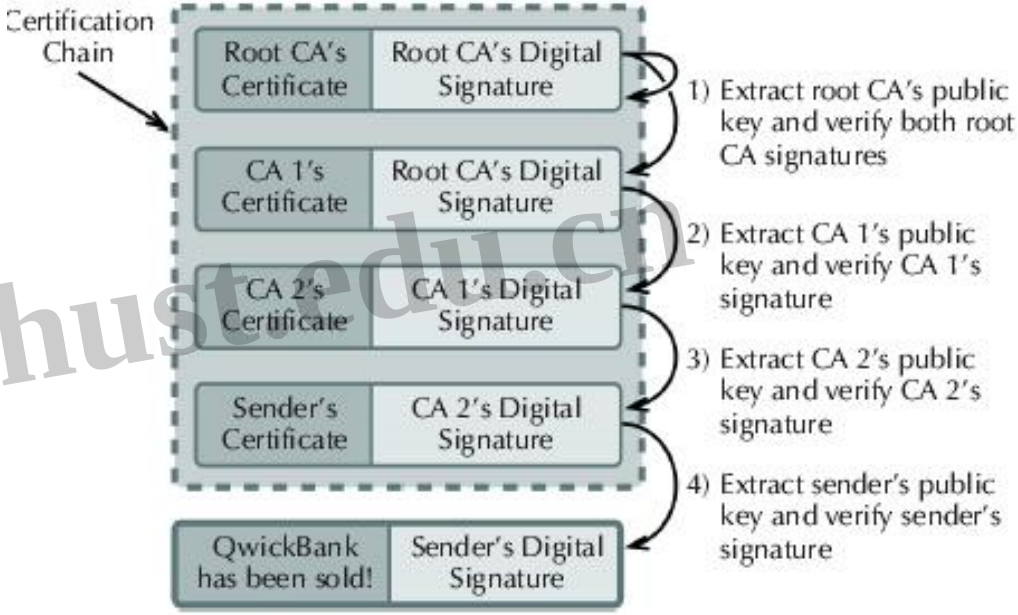
- 根CA具有一个自签名的证书
- 根CA依次对它下面的CA进行签名
- 层次结构中叶子节点上的CA用于对安全个体进行签名
- 对于个体而言，它需要信任根CA，中间的CA可以不必关心(透明的)；同时它的证书是由底层的CA签发的
- 在CA的机构中，要维护这棵树
  - 在每个节点CA上，需要保存两种cert
    (1) Forward Certificates: 其他CA发给它的certs
    (2) Reverse Certificates: 它发给其他CA的certs

层次结构CA中证书的验证

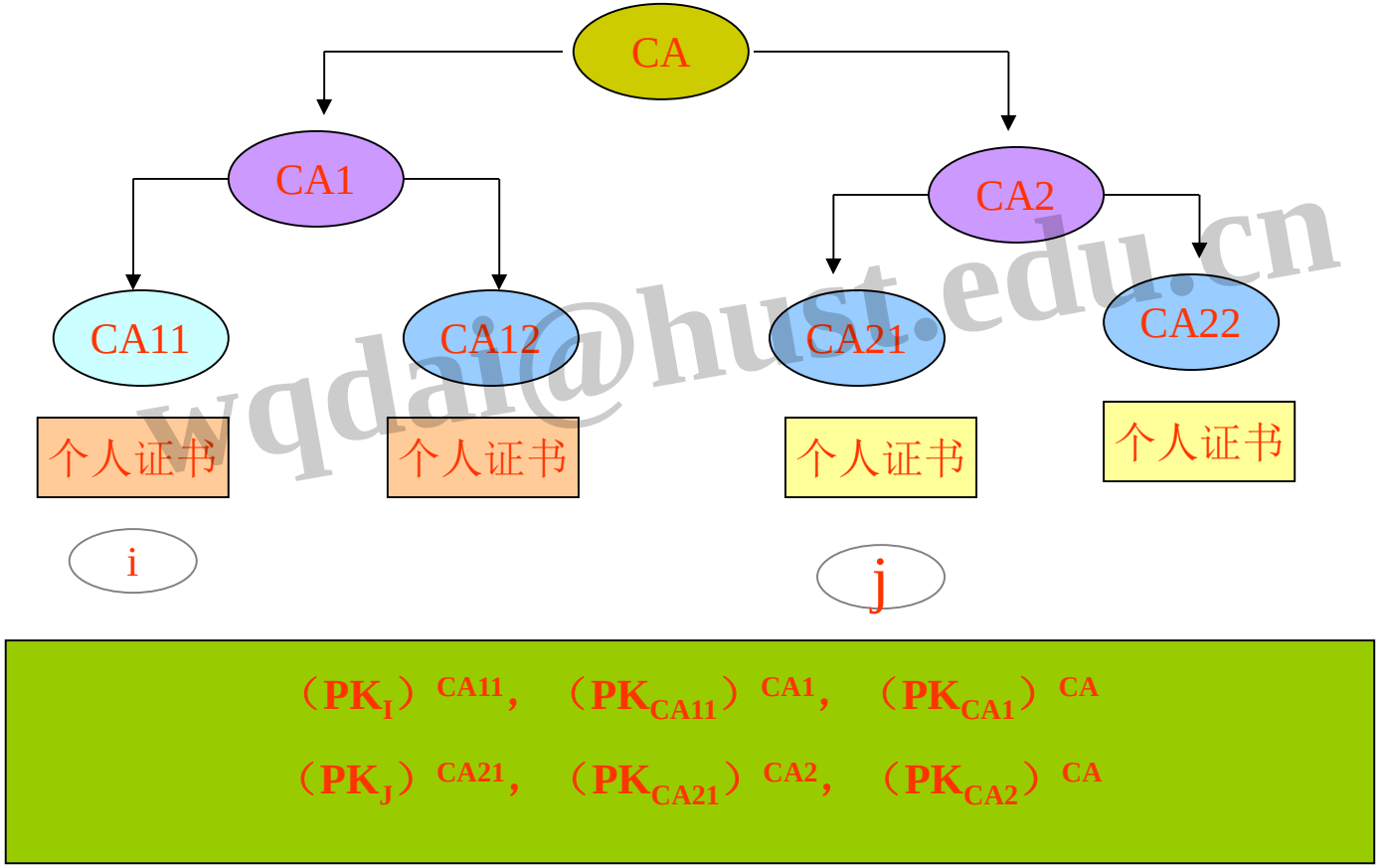
- 假设个体A看到B的一个证书
- B的证书中含有签发该证书的CA的信息
- 沿着层次树往上找，可以构成一条证书链，直到根证书
- 验证过程：
  - 沿相反的方向，从根证书开始，依次往下验证每一个证书中的签名。其中，根证书是自签名的，用它自己的公钥进行验证
  - 一直到验证B的证书中的签名
  - 如果所有的签名验证都通过，则A可以确定所有的证书都是正确的，如果他信任根CA，则他可以相信B的证书和公钥
- 问题：证书链如何获得？

证书链的验证示例





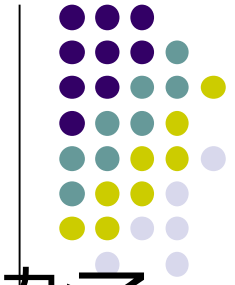
CA 证明链



CA认证模型

如果用户 i和j都属于CA11，那么i和j之间的密钥交换，只需要持有CA11开具的证明书就可以，即：对用户 i和j的公钥PKi 和PKj分别盖章，如$(Pki)^{ca11}$，$(Pkj)^{ca11}$,那么用户i和j就能证明密钥是对方的密钥。

CA认证模型



如果用户j的证书是CA21开具的，那么情况就复杂了，各自具有：

i方：$(Pki)^{ca11}$，$(CA11)^{CA1}$，$(CA1)^{CA}$
j方：$(Pkj)^{ca21}$，$(CA21)^{CA2}$，$(CA2)^{CA}$

这就形成了层层证明的证明链（certification chain）。这里，符号$(CA1)^{CA}$是CA对CA1的公钥盖章，只是证明本公钥是CA1的。

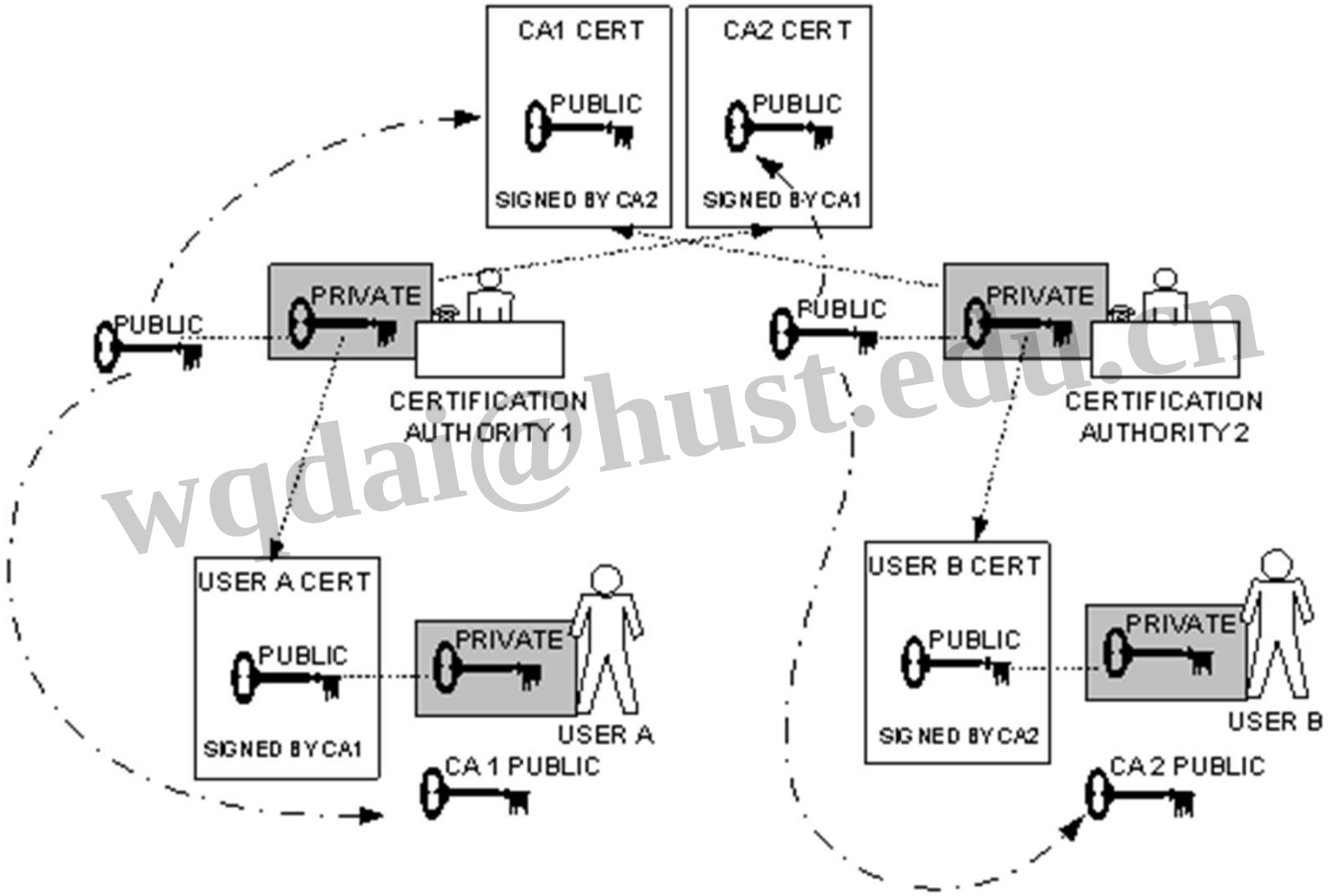
交叉认证

- 两个不同的CA层次结构之间可以建立信任关系
  - 单向交叉认证
    - 一个CA可以承认另一个CA在一定名字空间范围内的所有被授权签发的证书
  - 双向交叉认证
- 交叉认证可以分为
  - 域内交叉认证(同一个层次结构内部)
  - 域间交叉认证(不同的层次结构之间)
- 交叉认证的约束
  - 名字约束
  - 路径长度约束
  - 策略约束

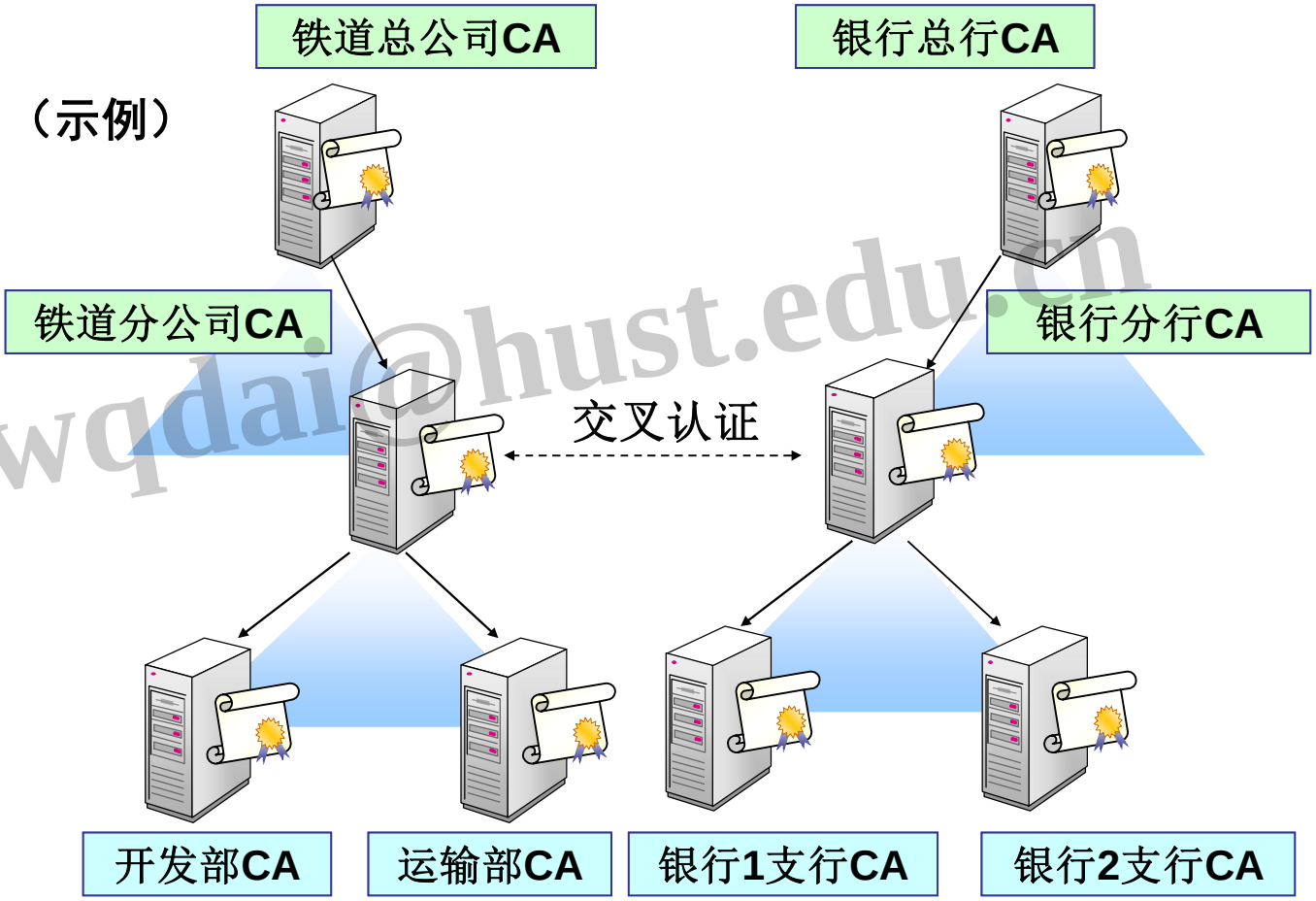
交叉认证

- 交叉认证是把以前无关的CA连接到一起的认证机制。当两者隶属于不同的CA时，可以通过信任传递的机制来完成两者信任关系的建立。
- CA签发交叉认证证书是为了形成非层次的信任路径。一个双边信任关系需要两个证书，它们覆盖每一方向中的信任关系。这些证书必须由CA之间的交叉认证协议来支持。当某证书被证明是假的或者令人误解的时候，该协议将决定合作伙伴的责任。

交叉认证



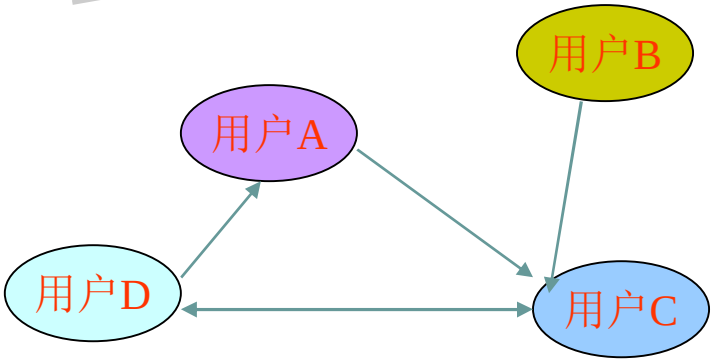
交叉认证



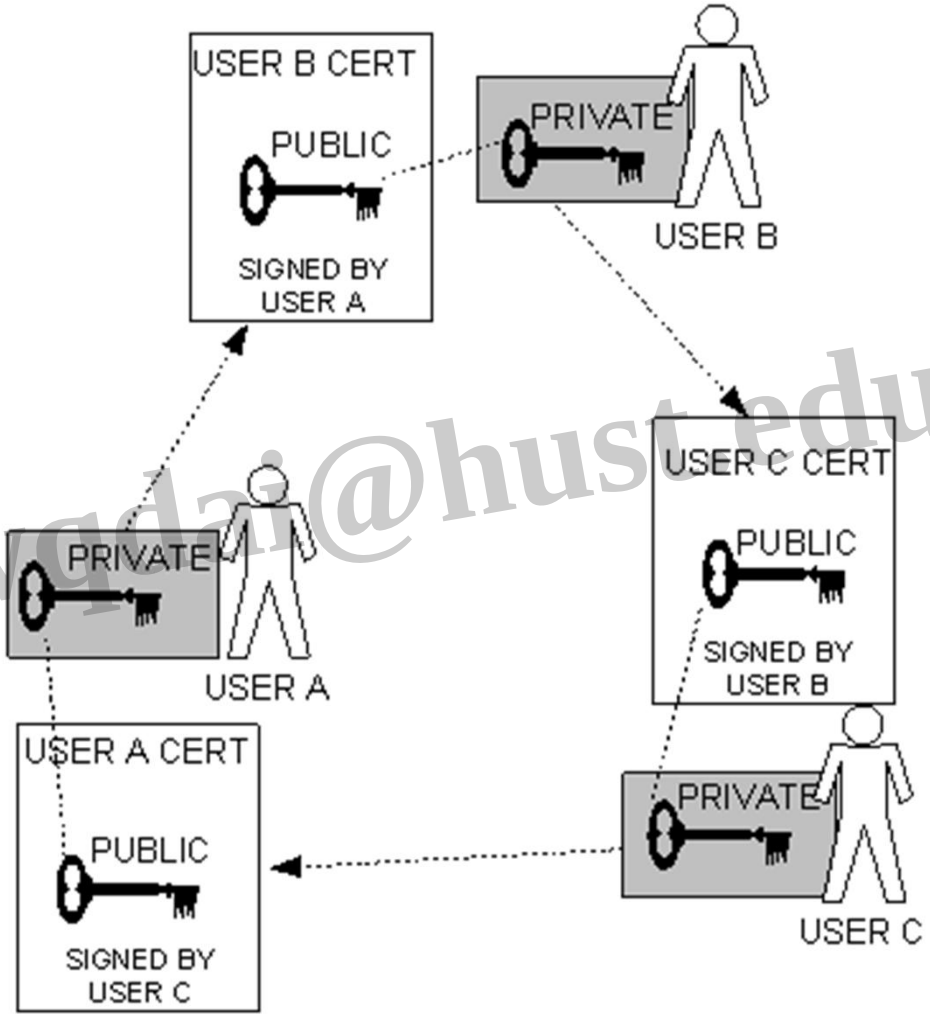
以用户为中心的认证

- 不存在集中式CA,用户之间相互颁发证书
- 每个用户为自己颁发的证书提供担保
- 自组织认证模型

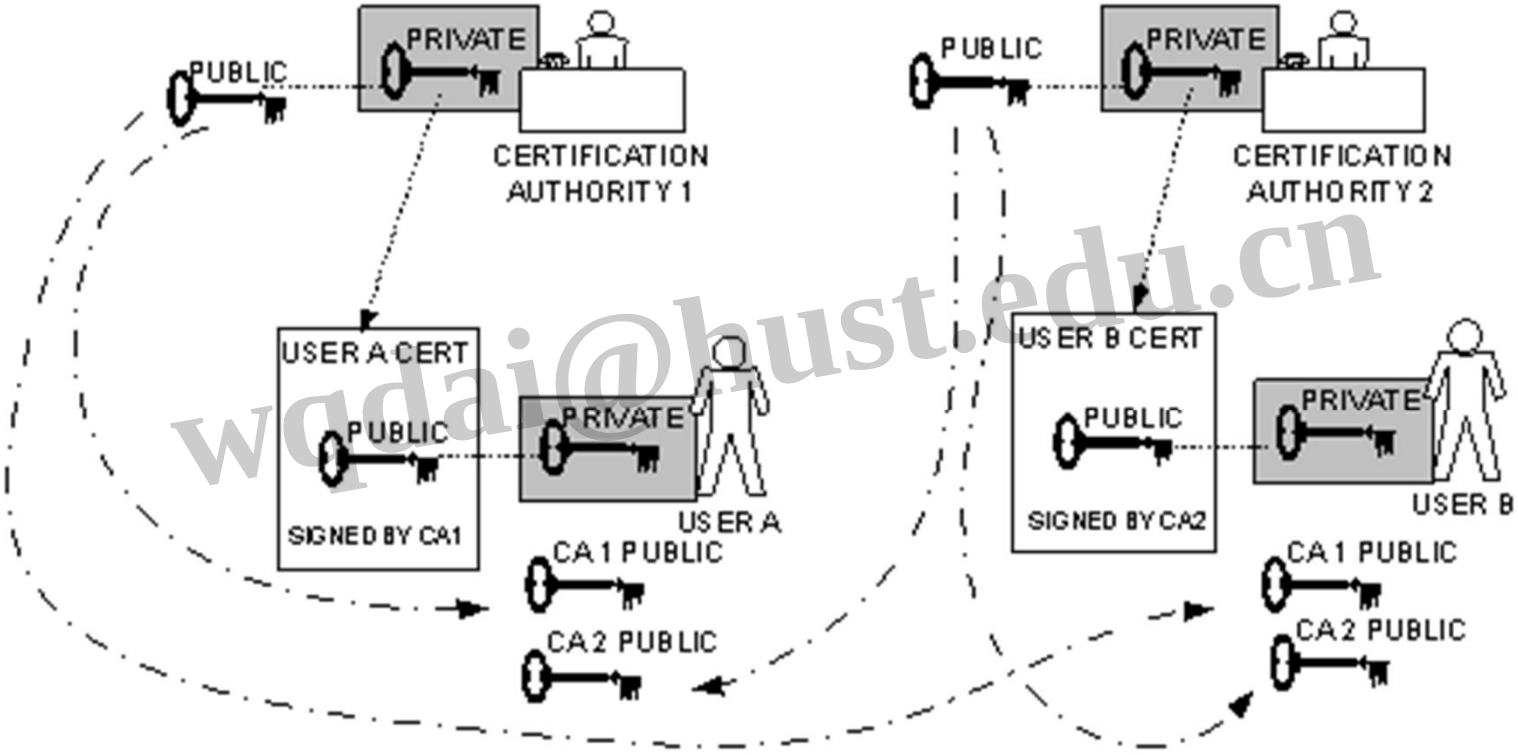
  eg:PGP
- 信任关系按照自然人的信任关系传播



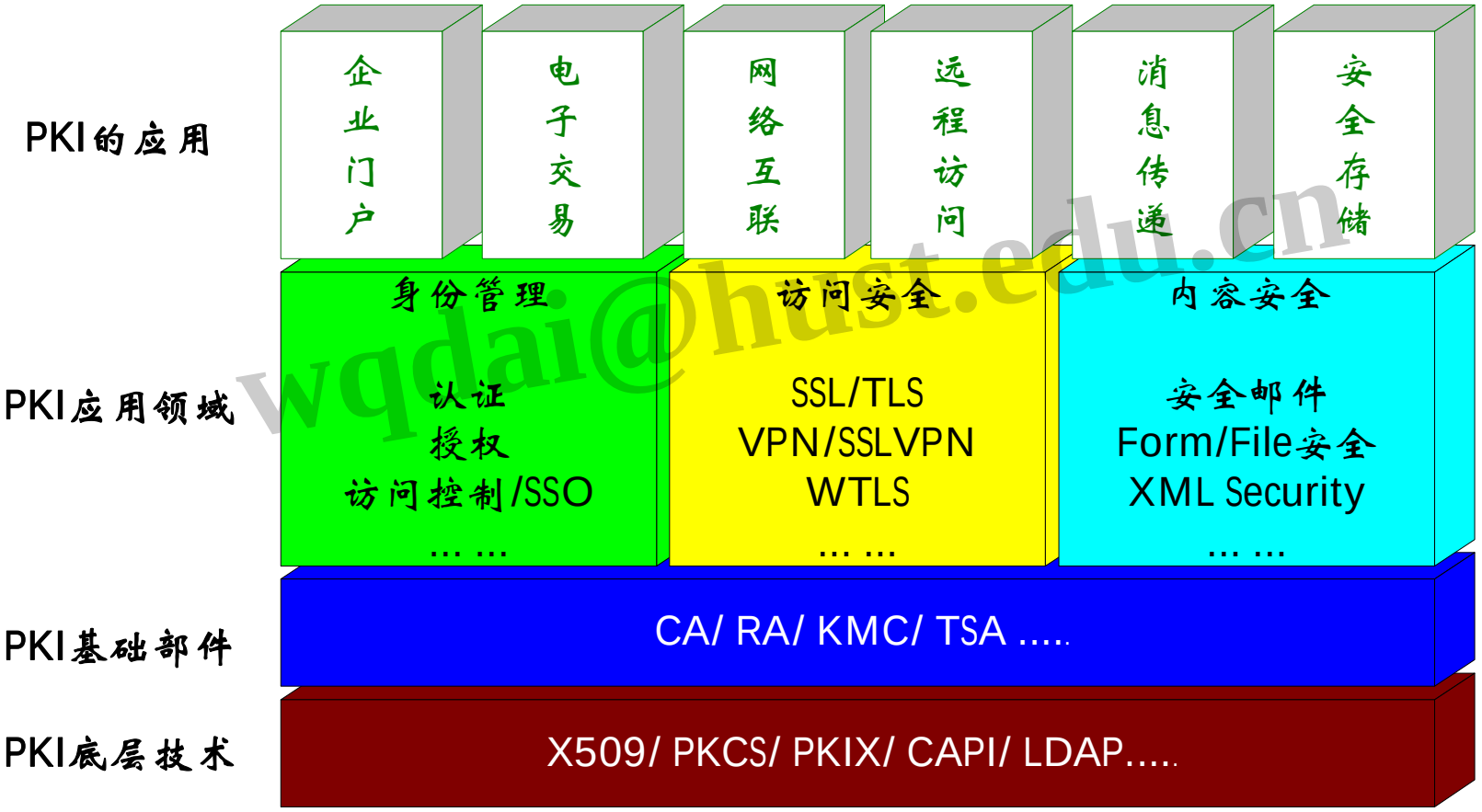
以用户为中心的认证



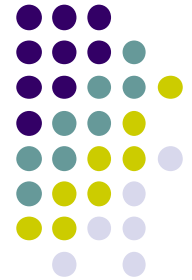
浏览器信任列表认证



PKI的应用领域与技术



PKI的标准化



- 随着PKI的逐渐普及，为了更好地为社会提供服务，不同厂商的PKI产品需要互连互通。
- 如电力用户要用数字证书到银行去交电费，银行的PKI就要对电力用户的证书进行认证（确认身份）。通常电力PKI和银行PKI是不同厂商的产品，这就需要两家PKI产品能互操作，这需要支持相同的标准，如证书格式及接口规范等。
- 同时，PKI产品自身的安全性也非常重要，这就需要专门的机构和标准规范对产品的安全功能和性能进行测评认定。
- 因此，标准化就成了PKI发展的必然趋势。

PKI的标准化

- 两代PKI标准
  - 第一代PKI标准，主要包括：
    - 美国RSA公司的公钥加密标准（Public Key Cryptography Standards，PKCS）系列
    - 国际电信联盟的ITU-T X.509
    - IETF组织的公钥基础设施X.509（Public Key Infrastructure X.509，PKIX）标准系列
    - 无线应用协议（Wireless Application Protocol ,WAP）论坛的无线公钥基础设施（Wireless Public Key Infrastructure，WPKI）标准等。

PKI的标准化

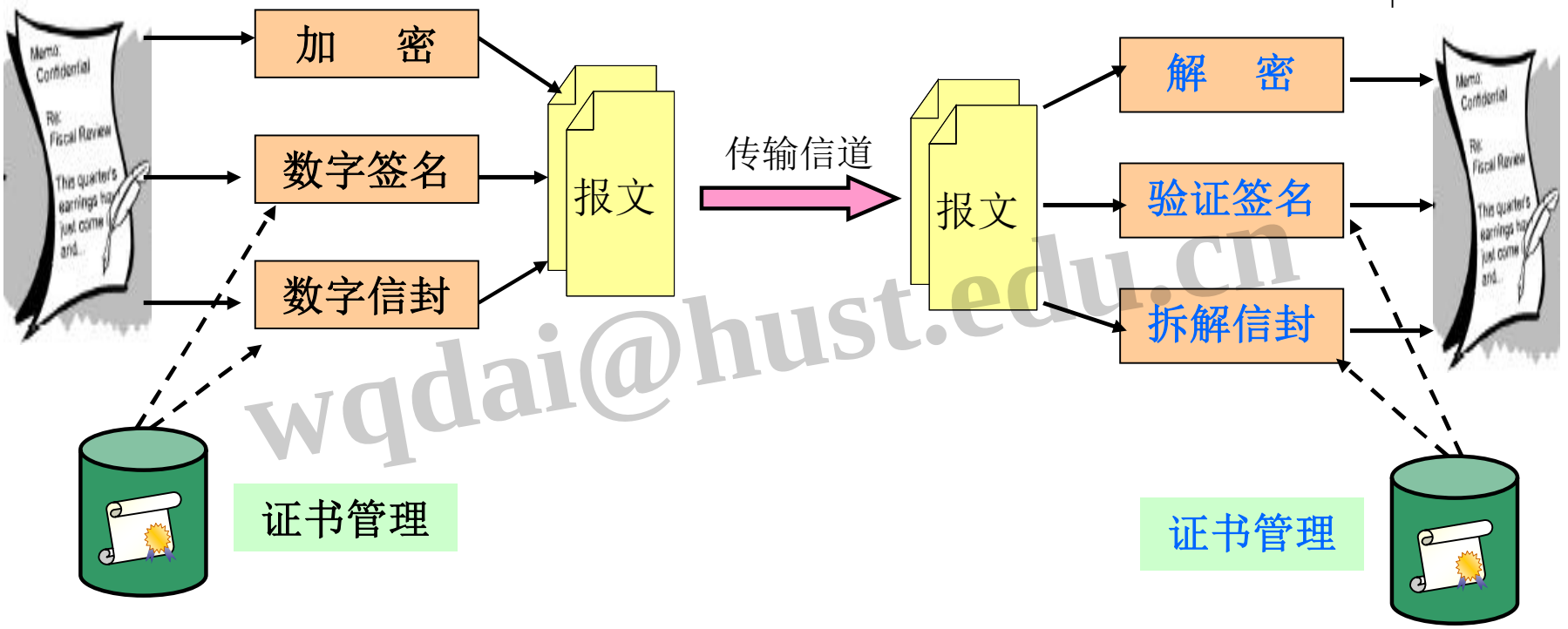
- 两代PKI标准
  - 第二代PKI标准，主要包括：
    - XML密钥管理规范，其全称为XML Key Management Specification，XKMS。
    - XKMS由两部分组成：XML密钥信息服务规范（XML Key Information Service Specification，X-KISS）和XML密钥注册服务规范（XML Key Registration Service Specification，X-KRSS）。它已经成为W3C的推荐标准。

公钥基础设施PKI——应用与典型案例

PKI本身是一种安全技术，它的目的是要在虚拟、开放的网络世界建立起一种信任机制，为网上业务系统的安全以及参与网上业务的各方的相互信任提供安全机制，为网上业务过程中身份认证和访问控制、信息保密性、信息完整性和业务操作的抗抵赖等安全需求提供可靠保证。

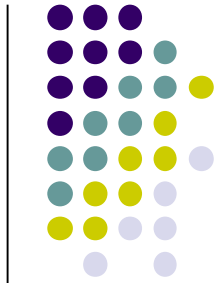
- PKI的主要目标是对各种安全应用服务提供支撑，并与具体的应用系统是分离的，PKI系统的设计、开发、生产及管理都可独立进行，不需考虑具体应用的特殊性；
- PKI通过延伸到用户本地的接口为各种应用提供安全服务，如身份鉴别、数字签名、机密性保护，访问控制等等；

一个基于PKI的文件安全传输示意图



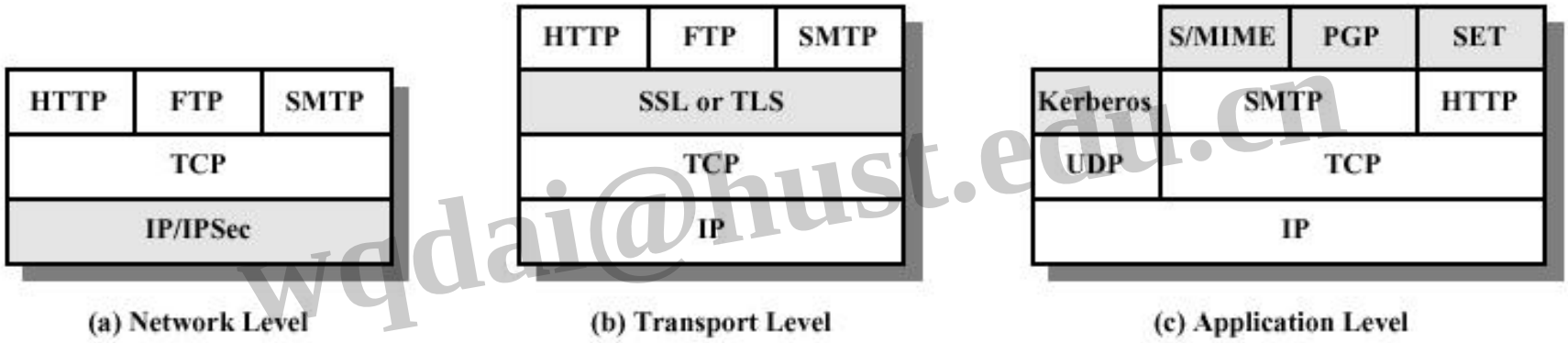
基于PKI的各种安全服务包括：

鉴别服务、机密性服务、抗抵赖以及访问控制等等。



TLS/SSL

TCP/IP协议栈中的安全



SSL (Secure Socket Layer)
安全套接层协议

- 特点：
  - 提供传输层的安全服务（按7层协议，也可以认为是会话层）
  - 最先(1995年)是由Netscape公司开发的，被广泛应用于Web等安全服务，后改编为Internet标准TLS (Transport Layer Security)（TLS1.0，RFC2246与SSL3.0类似）
  - 独立于应用层
  - 传输层采用TCP提供可靠业务
  - SSL 可分成两层：
    - SSL握手协议：用于在客户与服务器之间建立安全连接之前交换安全信息
    - SSL记录协议 （低）：确定数据安全传输模式

SSL解决的问题（功能）

- 客户对服务器的身份认证
  - SSL服务器允许客户端（如浏览器）使用标准的公钥加密技术和一些可靠的认证中心（CA）的证书，来确认服务器的合法性。
- 服务器对客户的身份认证
  - 也可通过公钥技术和证书进行认证，也可通过用户名，password来认证。
- 建立服务器与客户之间安全的数据通道
  - SSL要求客户与服务器之间的所有发送的数据都被发送端加密、接收端解密，同时还检查数据的完整性

SSL提供的安全服务

- 用户和服务器的合法性认证
  - using X.509v3 digital certificates
- 传输数据的机密性
  - using one of DES, Triple DES, IDEA, RC2, RC4, …
- 传输数据的完整性
  - using MAC with MD5 or SHA-1

SSL协议的发展

1 SSL(Secure Socket Layer)是Netscape公司设计的主要用于web的安全传输协议。

2 IETF(www.ietf.org)将SSL作了标准化，即RFC2246，并将其称为TLS（Transport Layer Security），TLS1.0与SSL3.0差别很小。

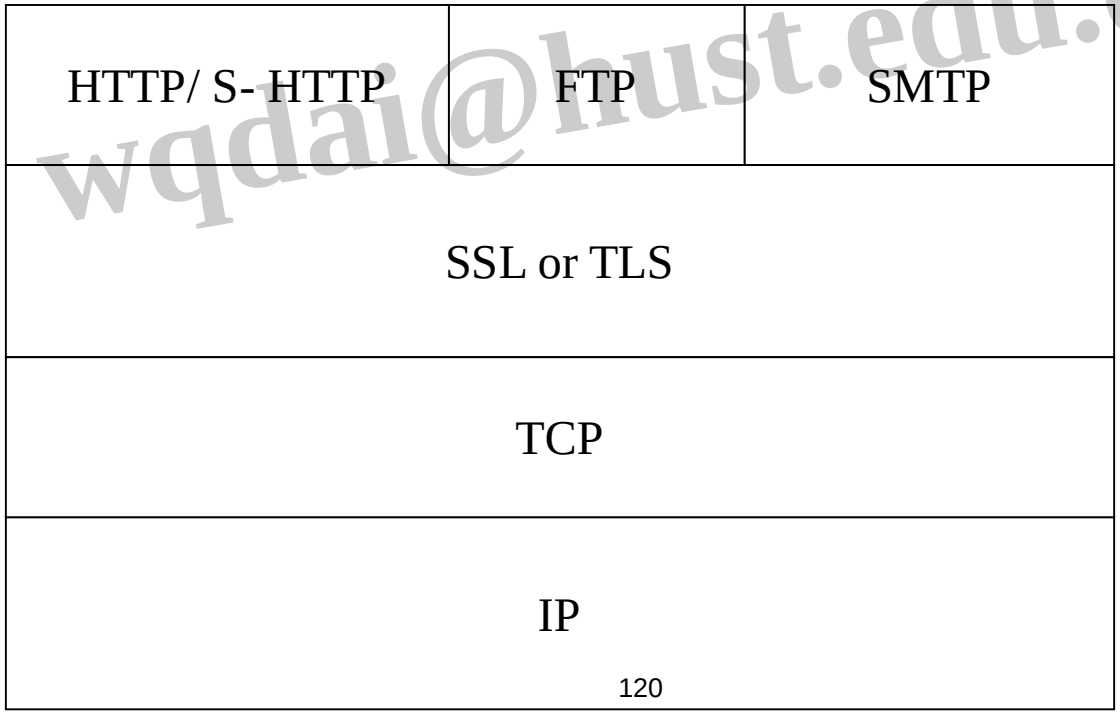
3 在WAP的环境下，由于手机处理和存储能力有限，wap论坛（www.wapforum.org）在TLS的基础上做了简化，提出了WTLS协议（Wireless TLS），以适应无线的特殊环境。

SSL协议的位置

- SSL协议要求建立在可靠的传输层协议(如：TCP)之上。SSL协议的优势在于它是与应用层协议独立无关的。
- 高层的应用层协议(例如：HTTP，FTP，TELNET)能透明的建立于SSL协议之上。SSL协议在应用层协议通信之前就已经完成加密算法、通信密钥的协商以及服务器认证工作。在此之后应用层协议所传送的数据都会被加密，从而保证通信的私密性。

SSL协议的位置

SSL协议建立在传送层和应用层之间，由记录协议和握手协议组成，其中记录协议在握手协议下端。SSL在TCP之上建立了一个加密通道。



主要功能

1、SSL服务器认证：允许用户确认服务器身份。支持SSL协议的客户机软件能使用公钥密码标准技术检查服务器证书、公用ID是否有效和是否由在客户信任的CA列表内的认证机构发放。

2、SSL客户机认证：允许服务器确认用户身份。使用应用于服务器认证同样的技术，支持SSL协议的服务器软件能检查客户证书、公用ID是否有效和是否由在服务器信任的认证机构列表内的CA发放。

主要功能

3、机密性：一个加密的SSL连接要求所有在客户机与服务器之间发送的信息由发送方软件加密和由接受方软件解密，这样提供了高度机密性。

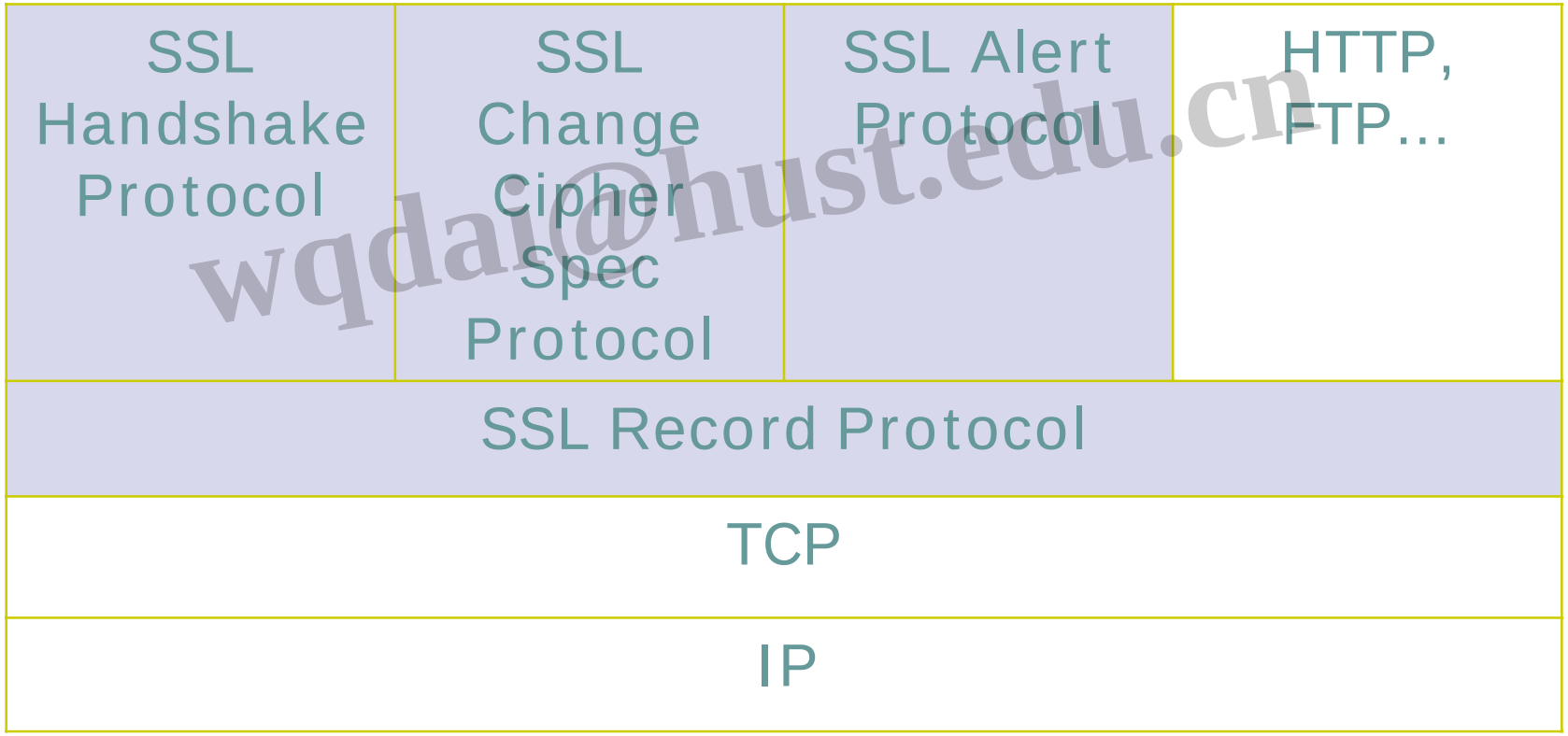
4、完整性：所有通过加密SSL连接发送的数据都被一种检测篡改的机制所保护，这种机制自动地决定传输中的数据是否已经被更改。

连接安全

- SSL协议提供的连接安全有三个基本属性：
  - 连接是保密的。对称加密法用于数据加密（如用DES和RC4等）。
  - 对方的身份能够使用非对称或公钥密码进行认证（如用RSA和DSS等）。
  - 连接是可靠的。消息传输包括使用消息认证码（MAC）的消息完整性检查，安全哈希函数（如SHA和MD5等）用于消息认证码计算。

SSL协议的组成

SSL协议的四个子协议：SSL记录协议、SSL握手协议、SSL改变密码规格协议、SSL告警协议。



SSL的工作原理

- 采用握手协议建立客户与服务器之间的安全通道，该协议包括双方的相互认证，交换密钥参数
- 采用告警协议向对端指示其安全错误
- 采用改变密码规格协议改变密码参数
- 采用记录协议封装以上三种协议或应用层数据（记录类型20=改变密码规格，21=告警，22=握手，23=应用层数据）

SSL协议的组成

记录协议定义了要传输数据的格式，它位于一些可靠的的传输协议TCP之上，用于各种更高层协议的封装。记录协议主要完成分组和组合，压缩和解压缩，以及消息认证和加密等功能。所有传输数据包括握手消息和应用数据都被封装在记录中。

握手协议允许服务器与客户机在应用程序传输和接收数据之前互相认证、协商加密算法和密钥。

SSL握手协议

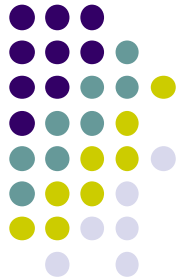
在初次建立SSL连接时使用SSL记录协议交换一系列消息。为如下操作做准备：

- 客户机对服务器的认证。
- 客户机与服务器都支持的加密算法或密码。
- 服务器对客户的认证（可选）。
- 使用公钥加密技术生成共享密钥。
- 建立加密SSL连接。

四个目标

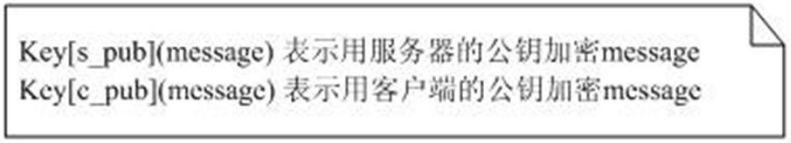
1. 加密安全性：在双方之间建立安全的连接。
2. 协同工作能力：独立程序员能够使用SSL3.0开发应用，然后能够在不知道他人代码的情况下成功地交换加密参数。
3. 可扩展性：SSL致力于提供一种框架，在必要时新的公钥和大批加密方法可以整合进来。
4. 相对效率：考虑到加密操作特别是公钥操作的速度问题，SSL协议使用可选的会话缓冲方案以降低连接数量，减少网络操作行为。

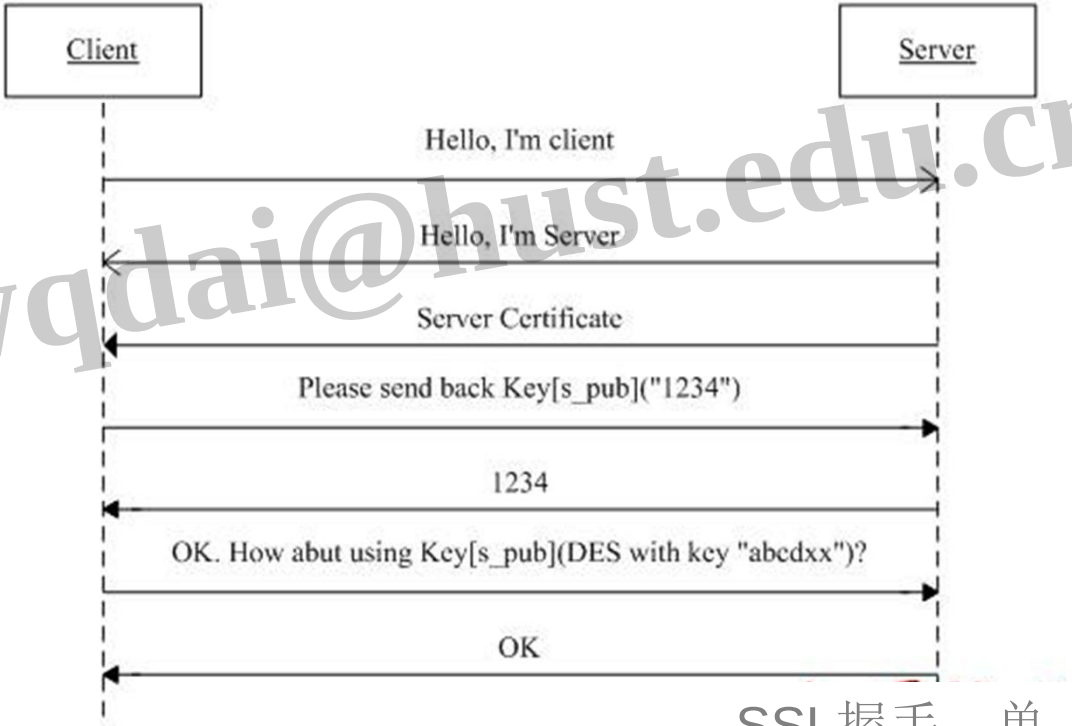
SSL 握手协议（Handshake Protocol）



- allows server & client to:
  - authenticate each other
  - to negotiate encryption & MAC algorithms
  - to negotiate cryptographic keys to be used
- comprises a series of messages in phases
  - Establish Security Capabilities
  - Server Authentication and Key Exchange
  - Client Authentication and Key Exchange
  - Finish

SSL 握手协议



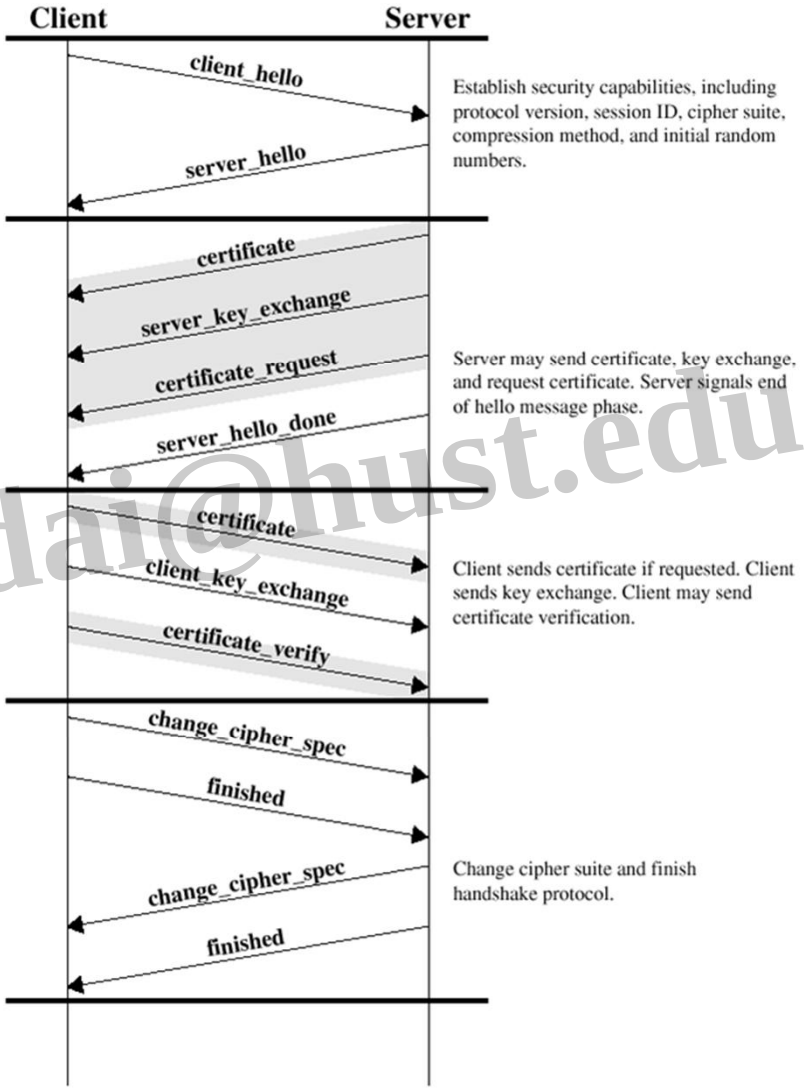


SSL握手，单方认证

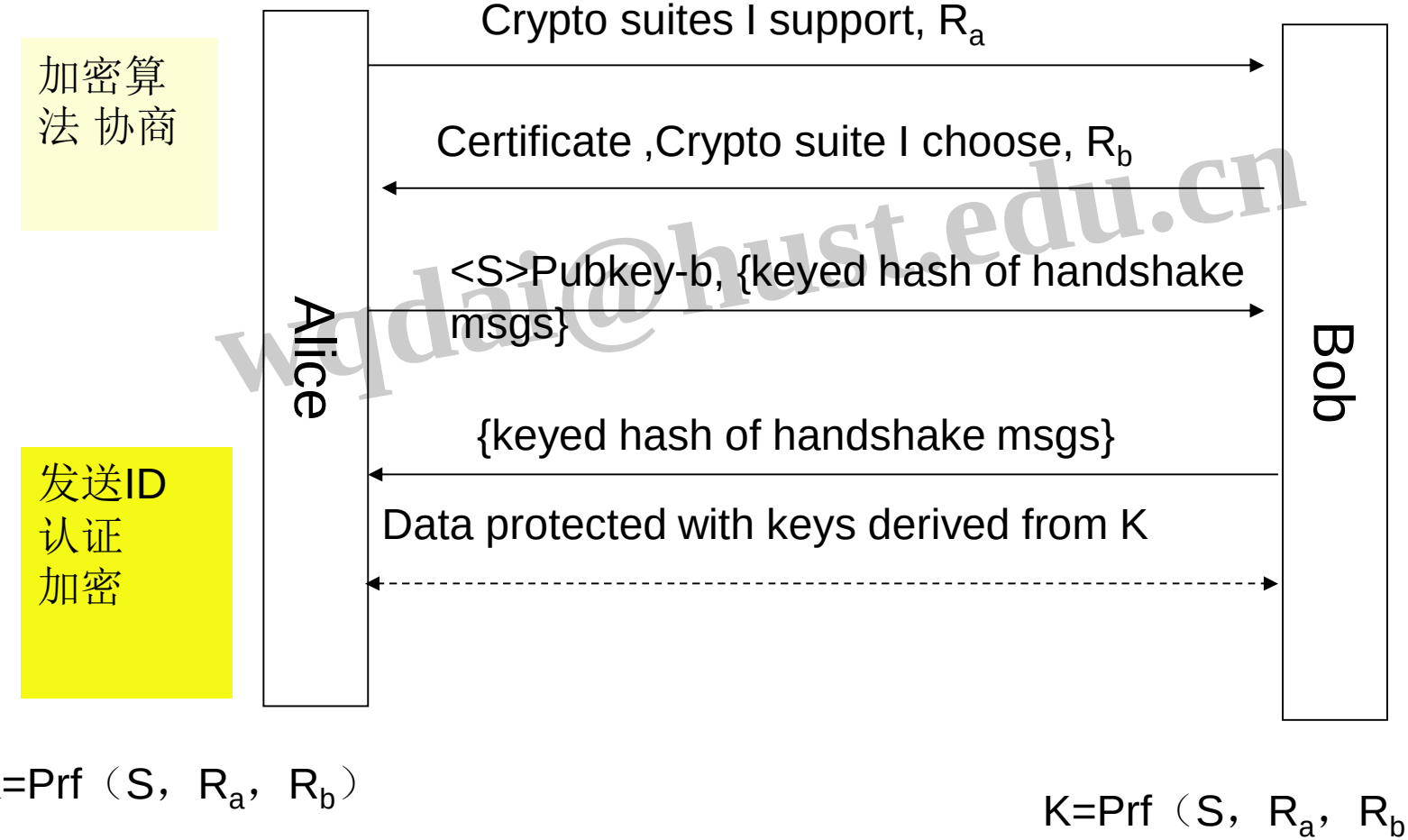
SSL 握手协议

SSL握手，双方认证

SSL 握手协议



The SSL Handshake Protocol



K=Prf（S，R_a，R_b）

K=Prf（S，R_a，R_b）

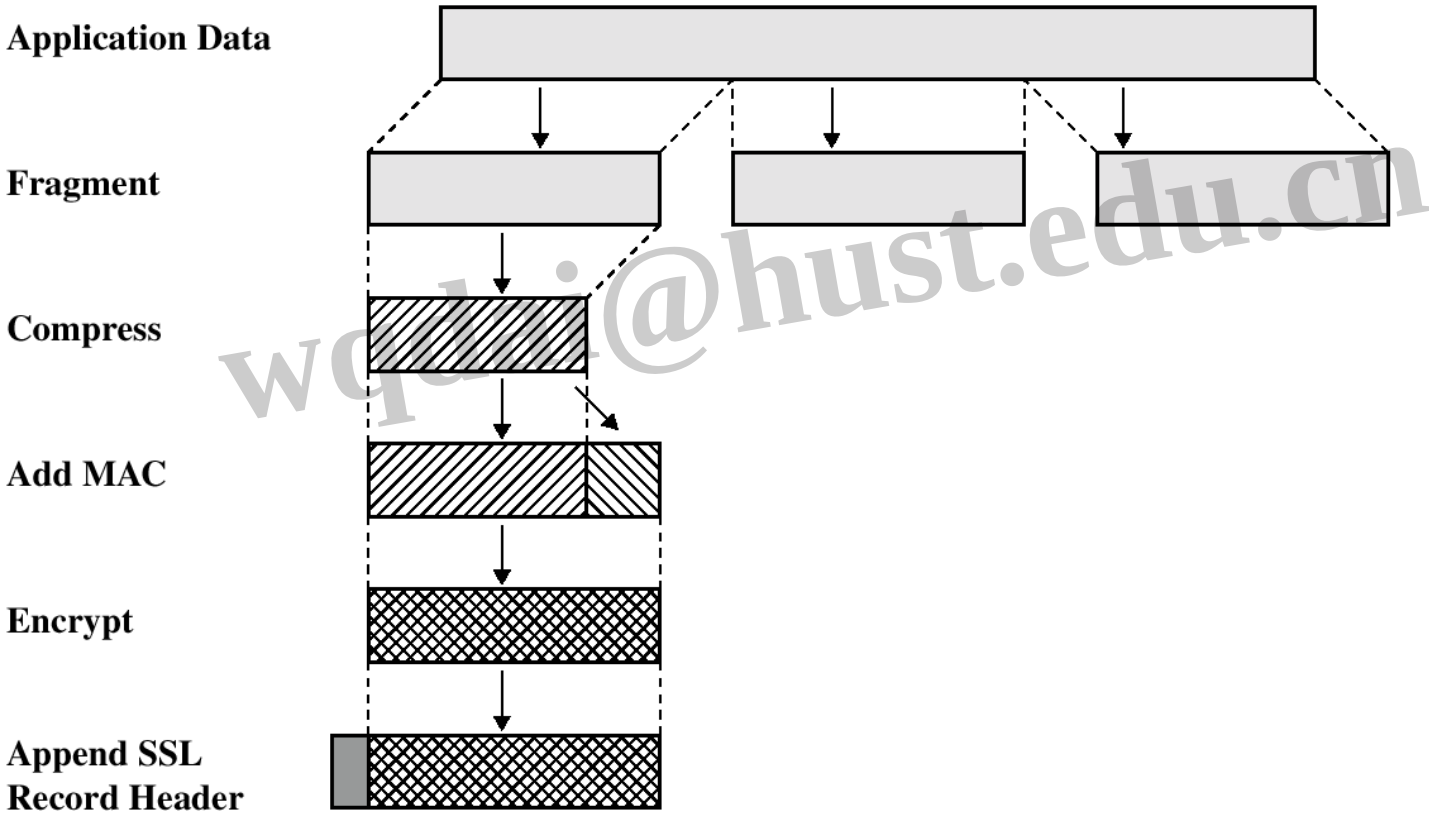
SSL改变密码规格协议和告警协议

- SSL Change Cipher Spec Protocol
  - 改变密码规格（参数）
- SSL Alert Protocol
  - 通知SSL对端有关安全错误警报，警报得级别有警告和致命两种（warning or fatal）
  - 错误的原因有：
    - unexpected message, bad record mac, decompression failure, handshake failure, illegal parameter
    - close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown

SSL记录协议（Record Protocol）

- 机密性（confidentiality）
  - using symmetric encryption with a shared secret key defined by Handshake Protocol
  - IDEA, RC2-40, DES-40, DES, 3DES, RC4-40, RC4-128
  - message is compressed before encryption
- 报文完整性（message integrity）
  - using a MAC with shared secret key
  - similar to HMAC but with different padding

SSL记录Protocol



SSL/TLS实现

- OpenSSL
- NSS (Network Security Services)
- GnuTLS
- Microsoft Win2k SSL implementation