西安电子科技大学 XIDIAN UNIVERSITY
计算机科学与技术学院 School of Computer Science and Technology
国家示范性软件学院 National Pilot School of Software Engineering

计算机安全导论

第1章 简 介

主讲人：张志为

二〇二四年秋季学期

提纲

PART 1 基本概念

PART 2 访问控制模型

PART 3 密码学相关概念

PART 4 实现与可用性问题

PART 5 作业

提纲

PART 1 基本概念

PART 2 访问控制模型

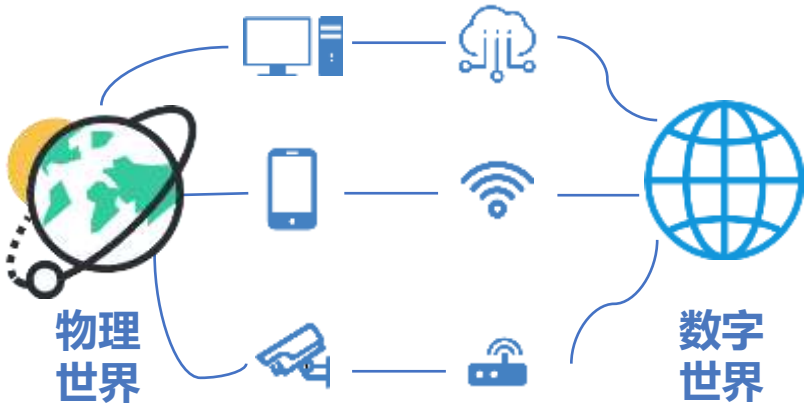
PART 3 密码学相关概念

PART 4 实现与可用性问题

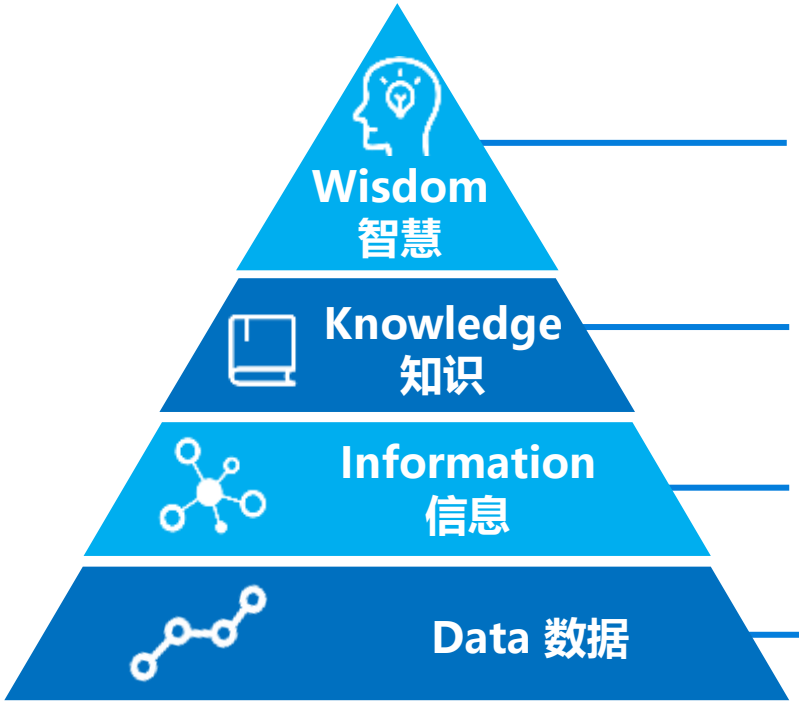
PART 5 作业

为什么“不安全”

数字化是物理世界向数字世界的迁移



DIKW金字塔层次结构



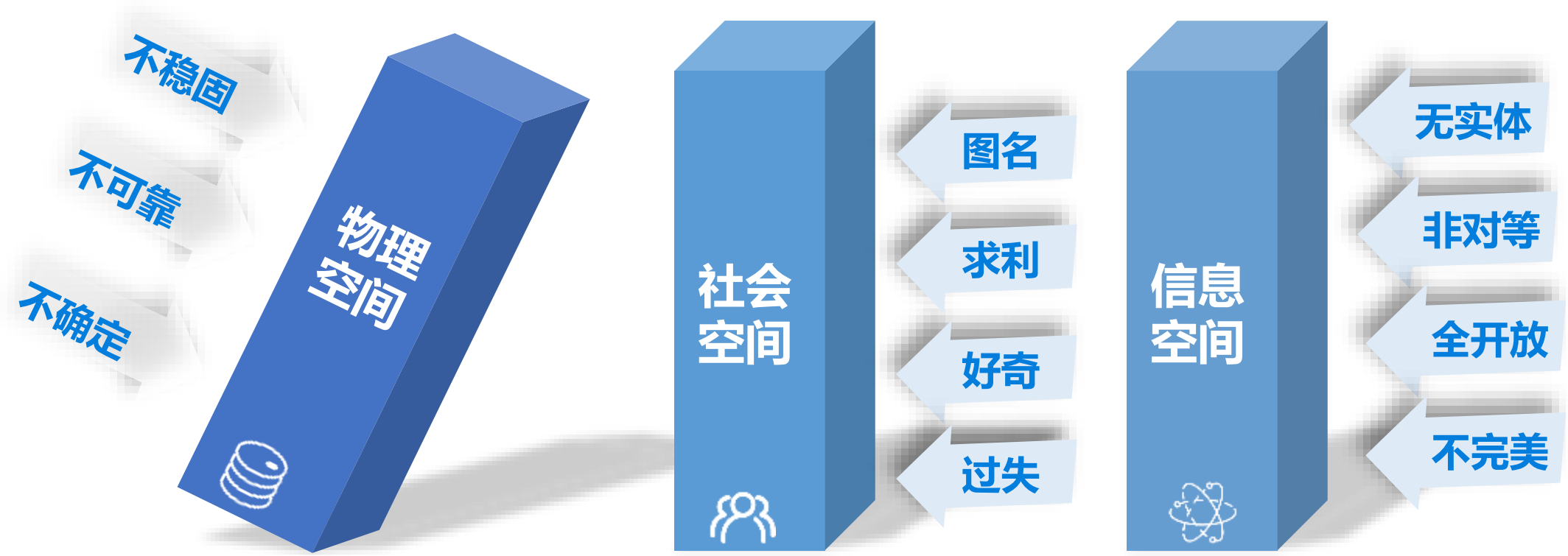
合理地应用应用知识并进行正确判断决策的能力

在对信息整合进行综合，提炼的基础上获得经验、判断与理解

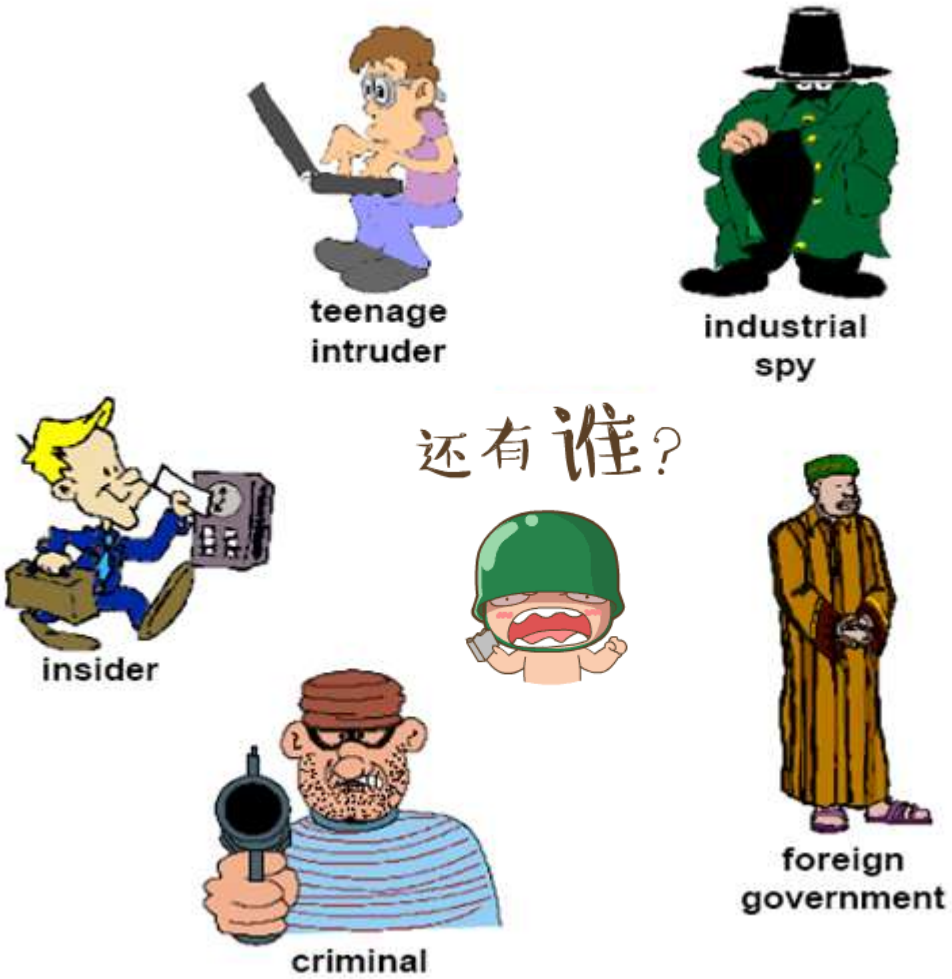
经加工后有逻辑，有意的数据

原始素材

为什么“不安全”



还有谁?


teenage
intruder
industrial
spy
还有谁?
insider
foreign
government
criminal

安全的基本概念

安全需求

Security Requirements

安全目标

C.I.A. and A.A.A.

安全威胁

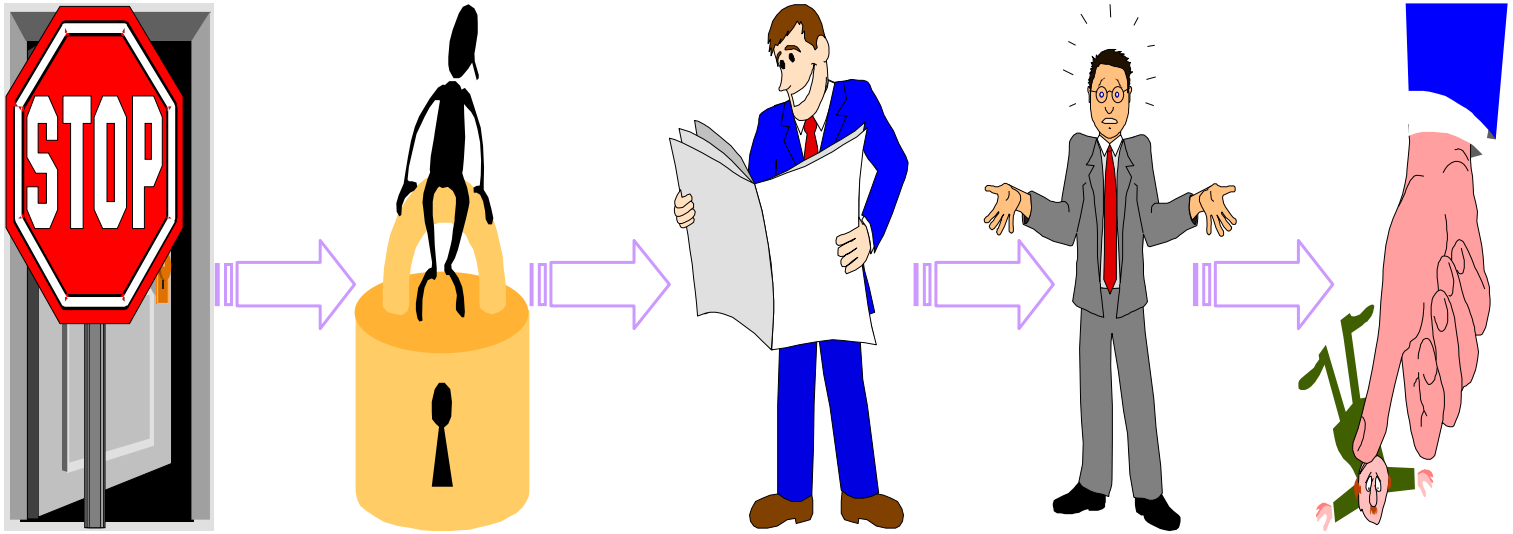
Threats and Attacks

安全原则

Security Principles

安全应达到的效果

进不来　　拿不走　　看不懂　　改不了　　跑不掉



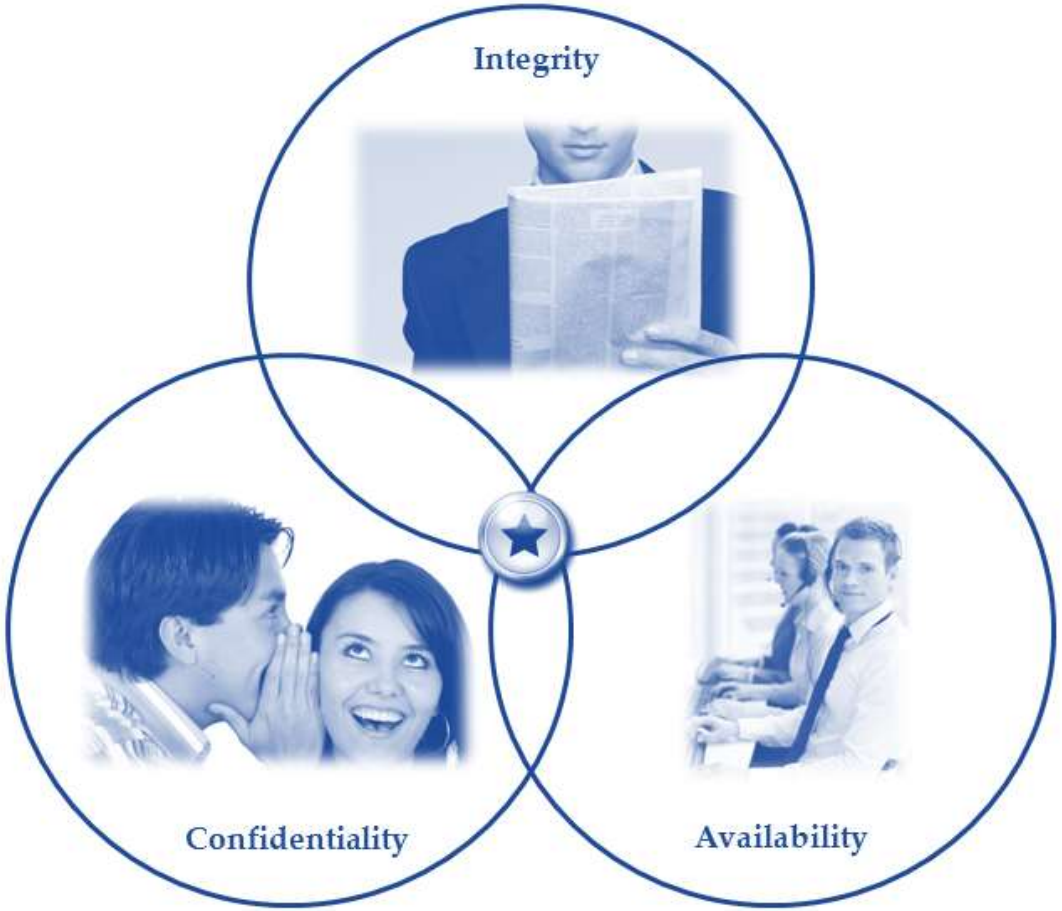
信息系统安全

- 数据安全
- 应用安全
- 主机安全
- 网络安全
- 物理安全

安全目标

CIA概念

- C: Confidentiality 机密性
- I: Integrity 完整性
- A: Availability 可用性



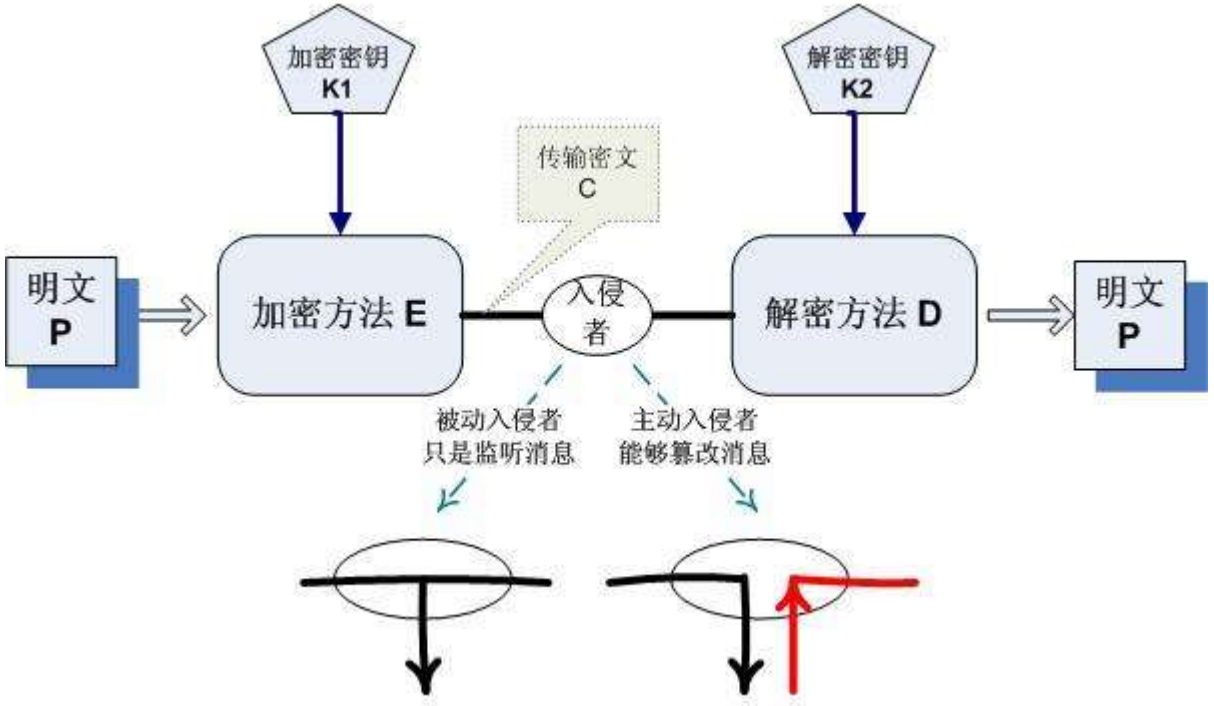
机密性Confidentiality

- **机密性(Confidentiality)定义**
  - 避免信息的非授权泄露，确保授权用户可以获取而非授权用户无法获取
- **保护机密性的工具(tools)**
  - 加密(Encryption)
  - 访问控制(Access control)
  - 认证(Authentication)
  - 授权(Authorization)
  - 物理安全(Physical security)

1997年电影《甲方乙方》李琦“打死不说” 梦

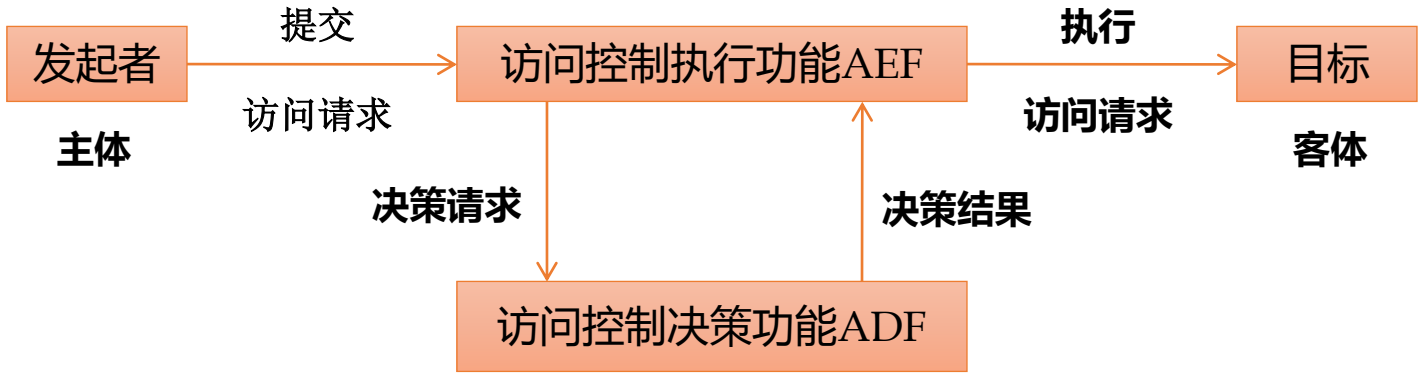
机密性Confidentiality

- **加密(Encryption)**
  - 使用秘密(加密密钥)传输信息，确保所传输的信息仅能用另一个秘密(解密密钥)获取
  - 加密和解密密钥可以一样也可以不一样



机密性Confidentiality

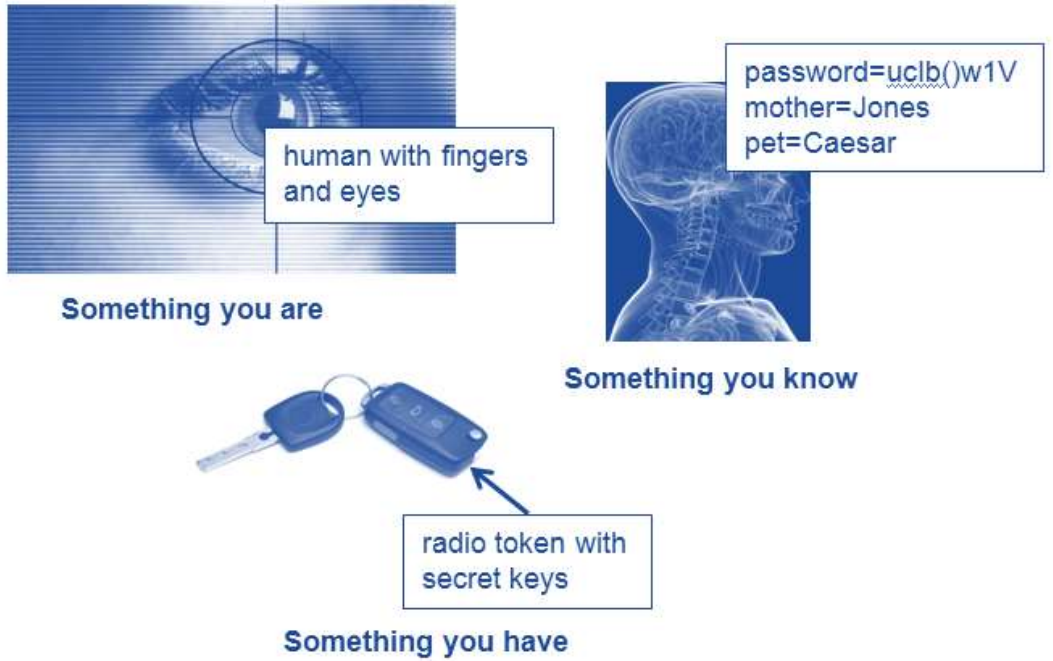
- **访问控制(Access control)**
  - 限制访问机密信息的一套规则和策略，需要提供身份信息或者角色以访问机密信息
  - 访问控制的目的是防止对信息资源的非授权访问和使用



访问控制流程示例图

机密性Confidentiality

- **认证(Authentication)**
  - 确定某人的身份或角色
  - 可通过不同的方式及这些方式的组合来确定
  - 你拥有的
    - 智能卡、USB-Key（数字证书）……
  - 你知道的
    - 用户名-口令……
  - 你自身的
    - 指纹、虹膜、面容、声音、步态、行为特征……



- **授权(Authorization)**
  - 基于访问控制策略，确定一个人或者系统是否允许访问资源
  - 阻止攻击者利用欺骗手段使其能够访问受保护的资源

机密性Confidentiality

- **物理安全(Physical security)**
  - 建立物理障碍限制访问受保护的计算资源
  - 这些障碍主要包括
    - 在机房和门上使用锁
    - 将计算机置于无窗户的房间
    - 使用声音抑制材料
    - 建造大楼或房间使其墙面有铜网(法拉第笼)以确保电磁信号进入或泄露

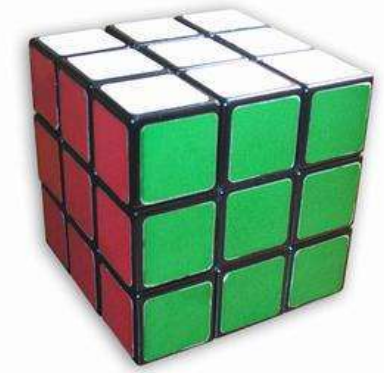
- **信用卡在线支付**
  - 当访问一个网页需要信用卡用来支付所购买的商品时，因特网浏览器在后台做了大量的工作确保信用卡号传输的机密性
  - 浏览器首先认证所访问的网站的真实性
  - 网站检查客户端浏览器是否可信，是否被授权
  - 浏览器和网站协商加密密钥
  - 浏览器将信用卡号等信息加密后发给网站服务器
  - 网站服务器还应确保信用卡号的安全存储

完整性Integrity

- **完整性(Integrity)定义**
  - 确保信息不被非授权修改的属性
  - 为什么会有完整性问题？恶意的修改或者无意的损坏
- **保护完整性的工具(tools)**
  - 备份(Backup)：定期归档数据
  - 校验和(Checksum)：计算一个函数将文件内容映射为一个数值。其**特性**是文件的小改变会使得函数输出有较大变化
  - 数据纠错码(data correcting codes)：在数据变化时可以检测并纠正
  - 保护元数据(metadata)，即描述数据的数据。更多信息请参考维基百科：https://en.wikipedia.org/wiki/Metadata。

思考题：

完整性、正确性与一致性是否相同？何时相同？



可用性Availability

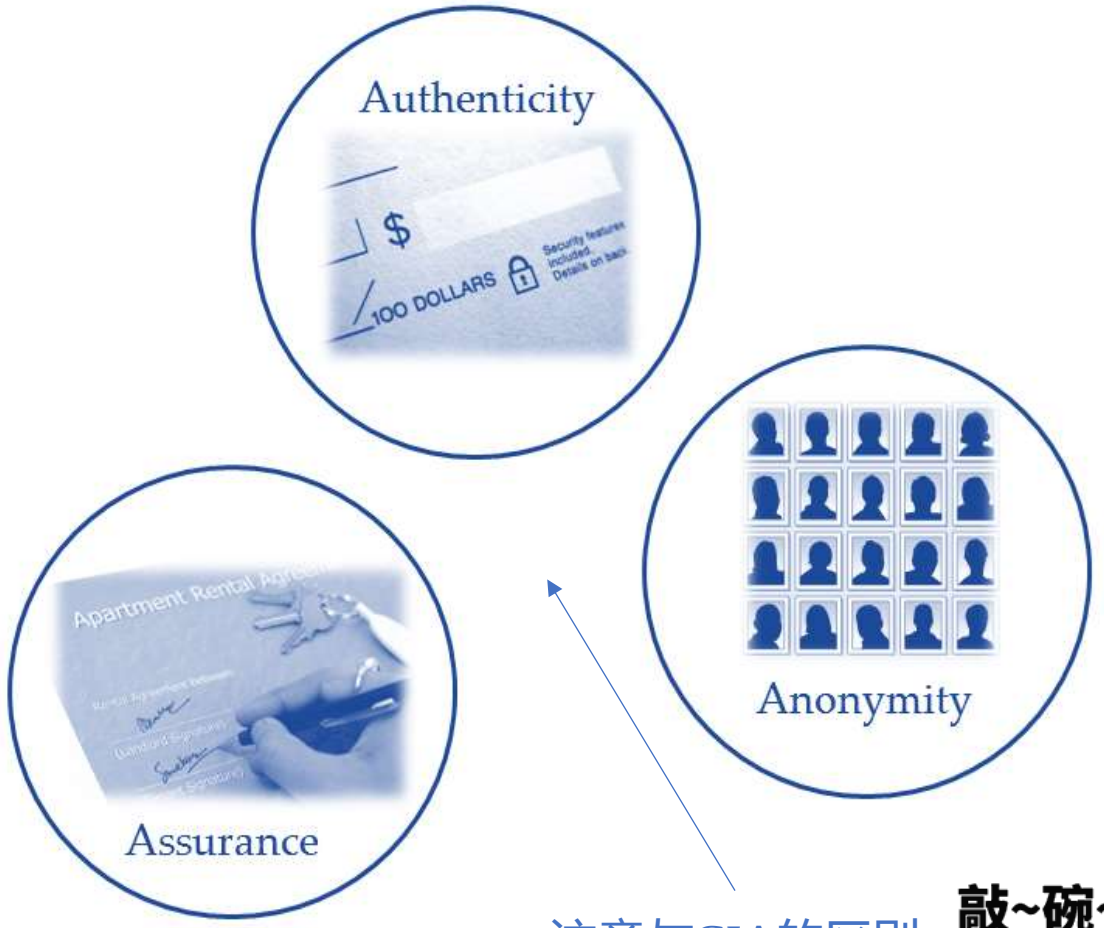
- **可用性(Availability)定义**
  - 确保授权用户能够对信息进行及时访问和修改的属性

- **保护可用性的工具(tools)**
  - 物理保护
    - 在物理环境恶劣变化时也能够保护信息的可用性
    - 例如将计算机系统置于特殊构建的楼房里以确保抵御暴风雨，地震和炸弹袭击等
  - 计算冗余
    - 在部分计算机或者存储设备失效的情况下依然可以提供信息
    - 例如采用磁盘冗余阵列(RAID)和多台计算机。关于RAID的更多信息请参考维基百科： https://en.wikipedia.org/wiki/RAID

安全目标

AAA概念

- A: Assurance 保证
- A: Authenticity 真实性
- A: Anonymity 匿名性



注意与CIA的区别

敲~碗~

- **保证(Assurance)定义**
  - 指计算机系统中信任如何被提供和管理
    - 系统资源的管理和保护：确保系统资源使用符合设定的策略
    - 信息使用方式的管理：确保信息如何被使用符合设定的策略
  - 信任管理
    - 策略(policy)： 指人或者系统对他们或者其他个体的行为预期
    - 权限(permission)： 指在与人和系统交互时所允许的行为
    - 保护(protection)： 指为了实施权限和策略所提供的机制
    - 例子：在线音乐系统(为用户如何访问歌曲设定**策略**；为购买特定歌曲的用户提供访问**权限**；为系统提供**保护**以防止用户未授权的访问)

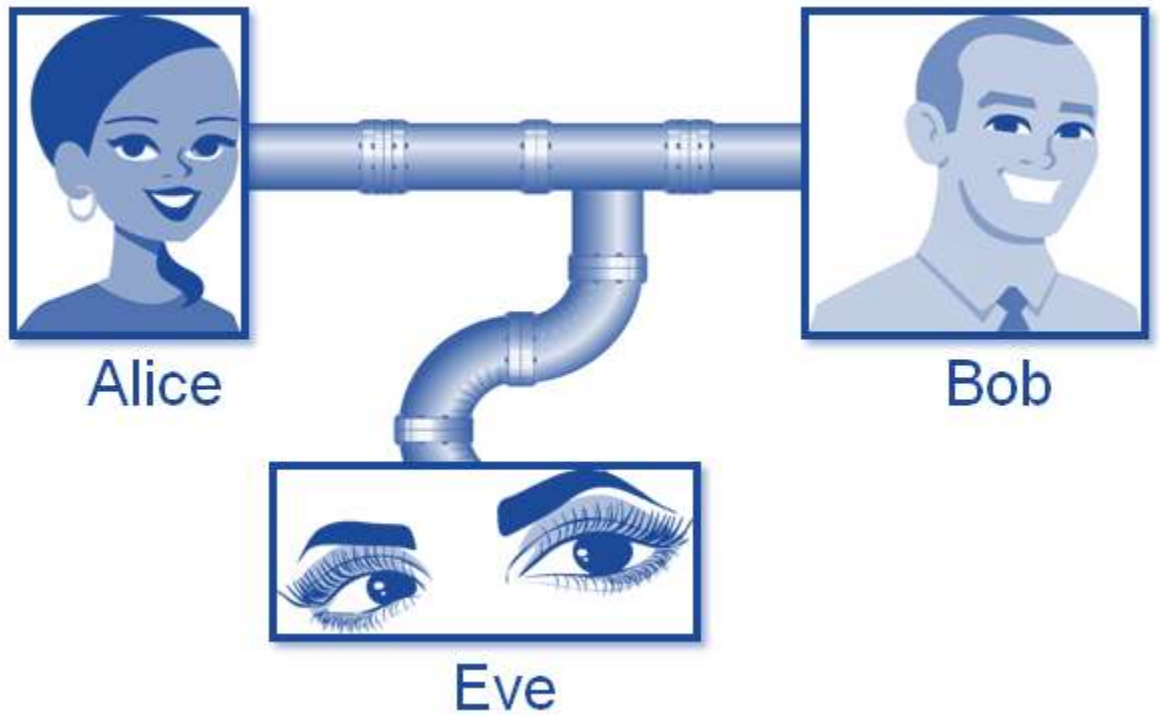
真实性Authenticity

- **真实性(Authenticity)定义**
  - 确定人或者系统所发布的声明、策略和权限真实性的能力
    - 真实性的满足能够确保在线交易的可行性
    - 不可抵赖性(Nonrepudiation)：一个人或者系统不能否认其所发布的真实声明
  - 主要工具
    - 数字签名(digital signatures)： 一种加密计算方法，允许人或者系统以唯一的方式认可与其相关数据的真实性。数字签名可实现不可抵赖性
    - 与纸质签名类比

匿名性Anonymity

- **匿名性(Anonymity)定义**
  - 确保特定的记录或者交易不归因于任何个体的属性
  - 不希望隐私泄露，如个体的医疗记录、购物记录等
  - 确保匿名性的方法
    - 数据聚集(Aggregation)
    - 数据混淆(Mixing)
    - 使用代理(Proxy)
    - 使用化名(Pseudonym)

- **窃听(Eavesdropping)**
  - 在通信信道中侦听发送给其他人的信息



Alice和Bob是密码学领域中的“著名人物”

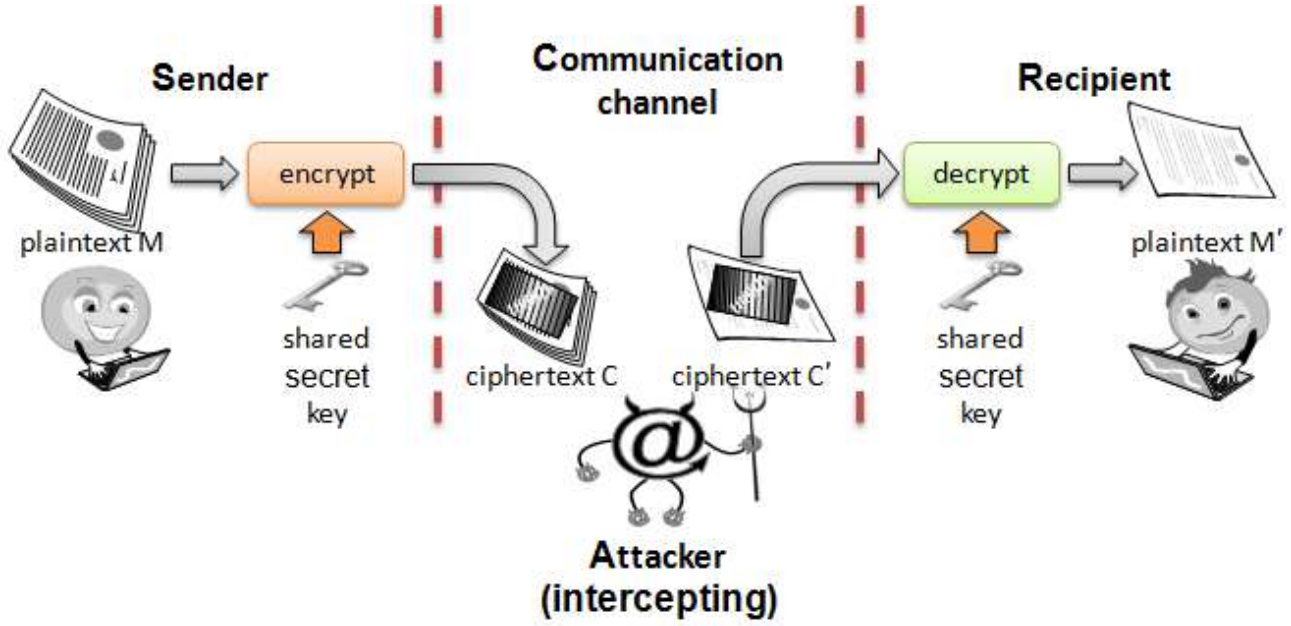
- 1978年，Rivest、Shamir、Adleman在论文中用Alice和Bob替代了A和B
- 根据Rivest回忆，使用Alice和Bob，一方面是避免使用枯燥无味的A和B，而这两个名字的英文首字母仍然维持A和B不变；另一方面Alice和Bob分别为女性名字和男性名字，在论文中就可以使用英语的她（she）和他（he）分别指代Alice和Bob，不至于让读者混淆

1985年，密码学家Bennett、Brassard、Robert在论文*How to Reduce Your Enemy's Information*中使用了Alice和Bob作为主人公，又给出了一个新的人物Eve

扩展：Alice、Bob 和他们的朋友们

爱丽丝（Alice）与鲍伯（Bob） 通例上，爱丽丝希望把一条讯息传送给鲍伯。
卡罗尔或查利（Carol或Charlie） 是通讯中的第三位参加者。
戴夫（Dave） 是通讯中的第四位参加者。
伊夫（Eve） 是一位偷听者（eavesdropper），但行为通常是被动的。她拥有偷听的技术，但不会中途篡改传送的讯息。在量子密码学中，伊夫也可以指环境（environment）。
艾萨克（Isaac） 是互联网服务提供者 (ISP)。
伊凡（Ivan） 是发行人，使用于商业密码学中。
贾斯汀（Justin） 是司法（justice）机关。
马洛里（Mallory） 是一位恶意攻击者（malicious attacker）。与伊夫不同的是，马洛里会篡改传送的讯息。对付马洛里所需的信息安全技术比对伊夫的高出很多。有时亦会叫作马文（Marvin）或马利特（Mallet）。
马提尔达（Matilda） 是一位商人（merchant），用于电子商务。
奥斯卡（Oscar） 是敌人，通常与马洛里一样。
帕特（Pat）或佩吉（Peggy） 是证明者（prover），维克托（Victor）是验证者（verifier）。两人会证实一项事件是否有实际进行，多使用于零知识证明。
普特（Plod或Officer Plod） 是执法官员。名称来自伊妮·布来敦所著的儿童文学《诺弟》（Noddy）中的角色“普特先生”。
史蒂夫（Steve） 代指隐写术（Steganography）。
特伦特（Trent） 是一位可信赖的仲裁人（trusted arbitrator），中立的第三者，根据存在的协议而判断。
特鲁迪（Trudy） 是侵入者（intruder），等同马洛里。
沃特（Walter） 是看守人（warden）。根据已存在的协议而保护爱丽丝和鲍伯。
佐伊（Zoe） 通常是一个安全协议中的最后参与者。

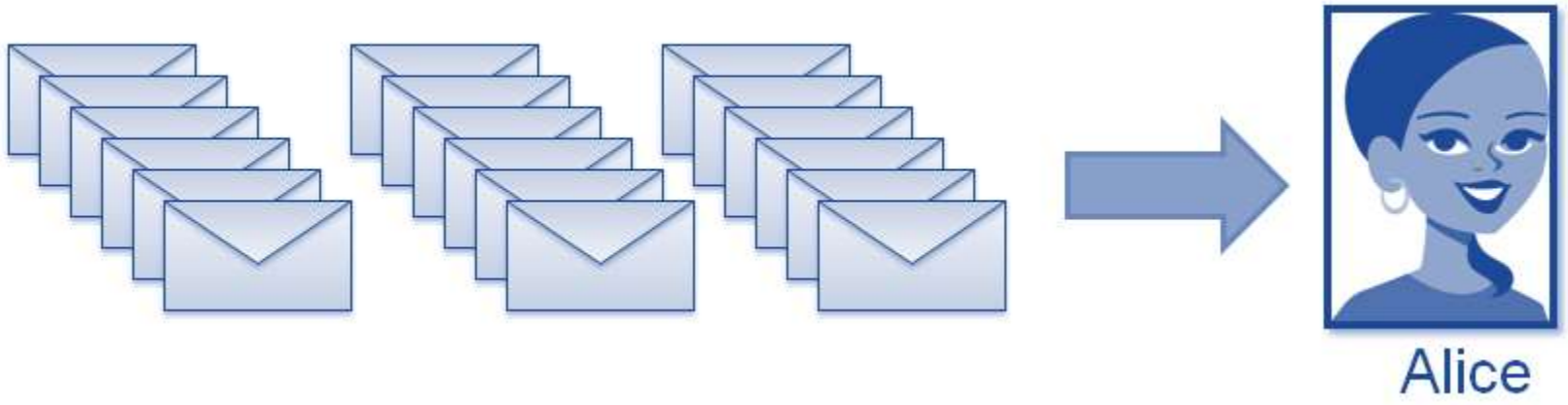
- **篡改(Alteration)**
  - 对信息的非授权修改
  - 例子：中间人攻击使得网络信息流被侦听、修改和重传



- **拒绝服务攻击(Denial of Service)**

DoS & DDoS

  - 对数据服务或者信息访问的中断或者降级
  - 例子：垃圾邮件，使邮件队列充满以降低邮件服务器的响应速度



- **伪装(Masquerading)**
  - 捏造消息并声称这些消息来自某人，但并非是消息的作者

"From: Alice"
(**really is from Eve**)

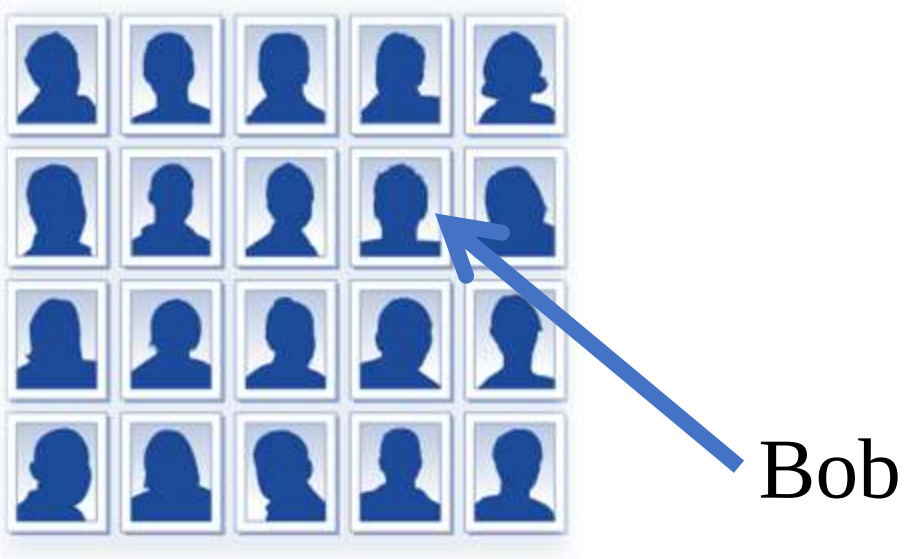
以假乱真的AI换脸术：
眼见未必为实

- **抵赖(Repudiation)**
  - 对承诺或者接收到数据的否认

- **相关性和追溯(Correlation and traceback)**

隐私保护

  - 结合多个数据源和信息流来确定一个特定数据流或信息的源头



- **十大安全原则(Security principles)**
  - J. H. Saltzer and M. D. Schroeder. The protection of information in computer systems. Proceedings of the IEEE, 63(9):1278-1308, 1975

安全原则

- **机制的经济性(Economy of mechanism)**
  - 着重强调在设计和实现安全措施时的简洁性(**simplicity**)
  - 一个简单的安全框架便于开发者和使用者理解，同时还可确保高效的开发和验证
- **故障安全默认配置(Fail-safe defaults)**
  - 所设计系统的默认配置应该具有保守的安全机制
  - 例如，当新用户进入一个操作系统时，其默认的用户组应该具有访问文件和服务的最小权限
  - 但操作系统或应用程序的默认选择往往是可用性>安全性。由于历史原因，一些流行的应用程序都是如此。例如web浏览器允许执行从web服务器端下载的代码

安全原则

- **完备调停(Complete mediation)**
  - 对于资源的每次访问必须检查以确保不违背安全机制
  - 由于访问权限可能随时间变化，在性能改善时需要谨慎对待，如对以前已经授权的检查不进行再次核查
  - 例如，一个在线银行web站点应该要求用户定期(如15分钟)重新登录
- **开放设计(Open design)**
  - 系统设计和安全体系结构应该公开，安全仅依赖于对密钥的保密
  - 开放设计使得系统可以被多方审查，从而尽早发现由于设计错误而导致的安全漏洞
  - 与通过隐匿来实现安全的方法相反。隐匿一般通过保密加密算法来实现，曾被一些组织使用，但经验表明该方法不够安全

安全原则

- **特权分离(Separation of privilege)**
  - 为了访问受限资源或者让程序执行某些操作，多个条件需要被同时满足
- **最小特权(Least privilege)**
  - 最小特权原则要求每个用户和程序在操作时应当使用尽可能少的特权
  - 可以限制特权滥用，对一个特定应用程序或用户账户的损坏降至最低
  - 例子：军方的信息须知(need-to-know)原则
- **最少公共机制(Least common mechanism)**
  - 在多用户系统中，允许多个用户共享资源的机制应该最小化
  - 例如，如果多个用户都需要访问文件或者应用程序，那么这些用户应该通过各自的通道来访问资源，以防止不可预见的结果引发安全问题

- **心理可接受性(Psychological acceptability)**
  - 用户接口应该精心设计且非常直观，所有与安全相关的设置都按普通用户的期望进行配置
- **工作因素(Work factor)**
  - 在设计安全方案时，绕过安全机制的代价应与攻击者的资源做对比
  - 例如，用于保护大学数据库系统所需要的安全措施与用于保护军事机密的系统相比要简单的多
- **危害记录(Compromise recording)**
  - 有时候记录入侵细节比采用更复杂的措施来预防入侵更为理想
  - 例如，互联网连接的监控摄像头是一个有效记录危害系统的典型例子，部署监控摄像头来代替防盗门窗来保护大楼

提纲

PART 1 基本概念

PART 2 访问控制模型

PART 3 密码学相关概念

PART 4 实现与可用性问题

PART 5 作业

访问控制模型

- **预防攻击最好的方法之一**
  - 从源头上阻止攻击
  - 通过确定谁有权限访问各种信息(即访问控制)，就能够防御针对机密性、完整性和匿名性的攻击

- **如何管理访问控制(Access control)**
  - 访问控制矩阵(Access control matrices)
  - 访问控制列表(Access control lists, ACL)
  - 能力列表(Capabilities)
  - 基于角色的访问控制(Role-based access control, RBAC)

- **访问控制模型的假设**
  - 假设有数据管理者、数据所有者或者系统管理员来定义访问控制
  - 限制用户只能访问和修改与他们相关的信息，即采取最小权限原则

访问控制矩阵

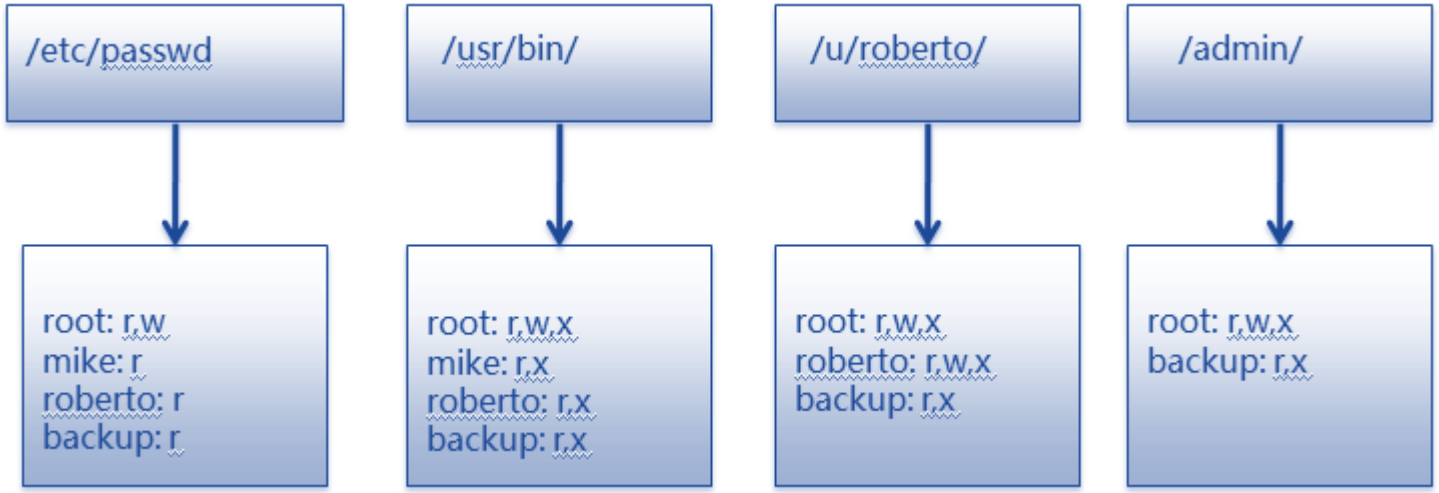
- **访问控制矩阵：定义访问权限的表**
  - 矩阵每一行与一个主体关联，主体包括用户，组，执行操作的系统等。
  - 矩阵每一列与一个客体关联，客体有文件，文件夹，文档，设备等需要定义访问权限的实体
  - 矩阵的每一个单元格填入主客体对的访问权限，包括读，写，复制，执行，删除和注释等；空单元格表示不授予任何权限

	/etc/passwd	/usr/bin/	/u/roberto/	/admin/
root	read, write	read, write, exec	read, write, exec	read, write, exec
mike	read	read, exec		
roberto	read	read, exec	read, write, exec	
backup	read	read, exec	read, exec	read, exec
. . .	. . .	. . .	. . .	. . .

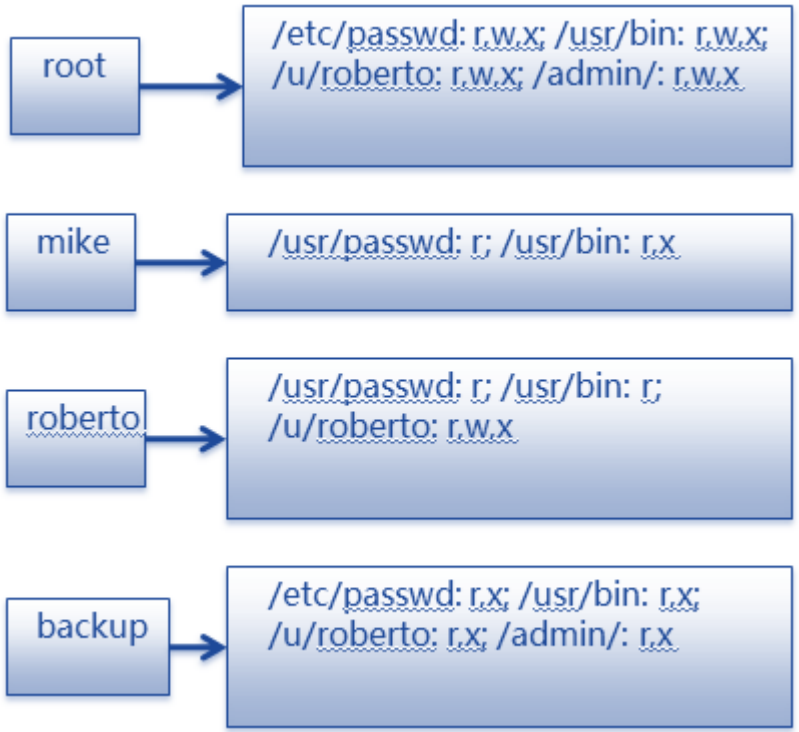
访问控制矩阵

- **访问控制矩阵的优点**
  - 很容易确定主客体对的访问控制权限
  - 管理者能够简单、直观地观察访问控制关系
  - 控制粒度细，针对每一个主客体对
- **访问控制矩阵的缺点**
  - 最大的缺点：缺乏可扩展性，访问控制矩阵有可能会很大。
  - n个主体和m个客体，则访问控制矩阵有n×m个单元。若一个服务器有1,000个主体和1,000,000个客体，那么访问控制矩阵有1亿个单元格
  - 如何克服？访问控制列表，能力列表和基于角色的访问控制

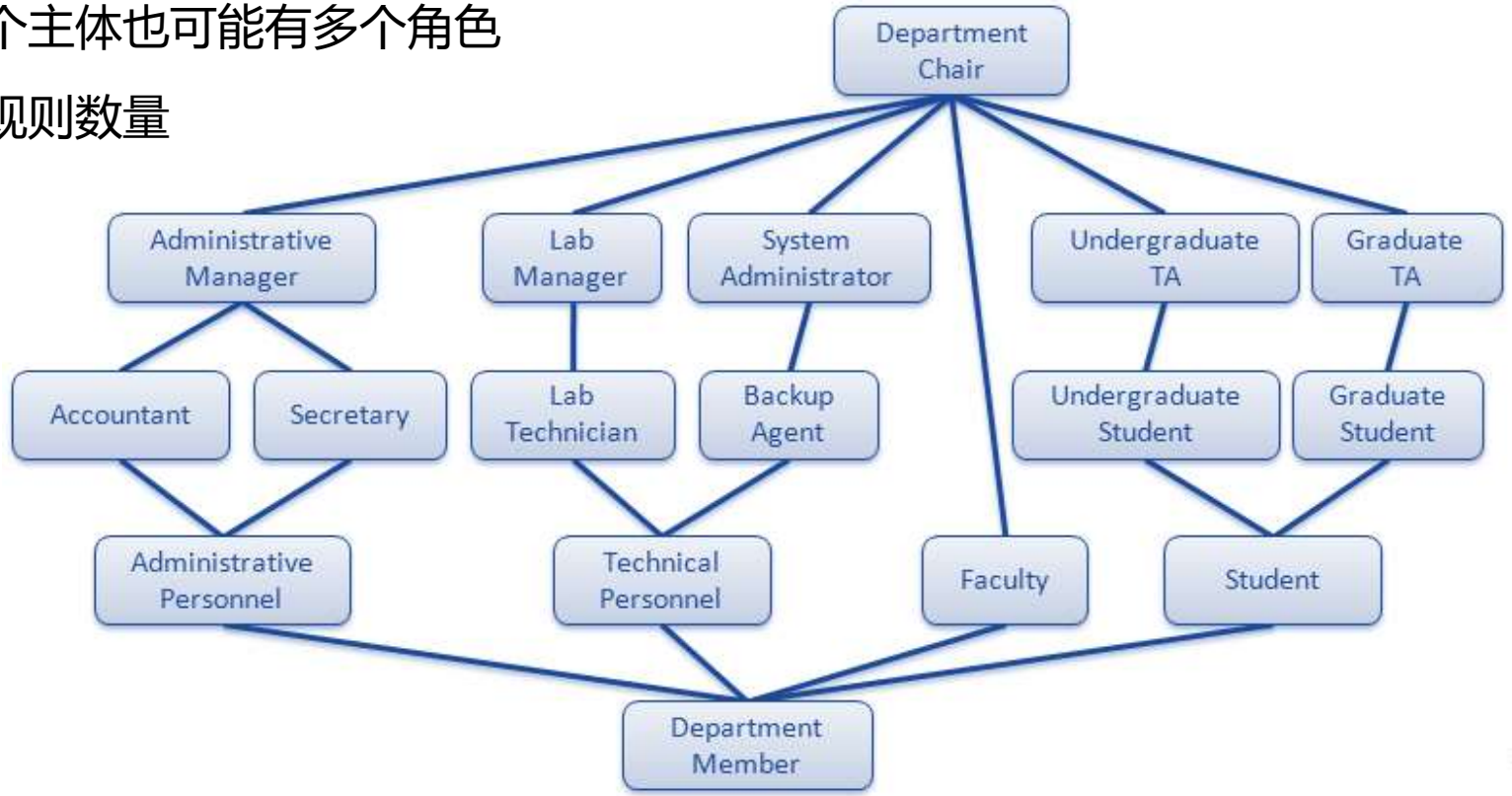
- **访问控制列表**
  - 对于每个客体o，定义列表L， L为客体o的访问控制列表，该列表列举了所有对客体o有访问权限的主体
  - 本质上，ACL模型使用访问控制矩阵的每一列并忽略其空白单元格
  - 优点：与访问控制矩阵比**小很多**，可将客体的列表以**元数据形式**存储
  - 缺点：枚举每个主体的访问权限复杂度高。这种枚举是必须的，例如将一个主体从系统中移除的时候



- **能力列表**
  - 对于每个主体s，定义其客体列表，且s具有非空的访问控制权限
  - 本质上，能力列表使用访问控制矩阵的每一行并忽略其空白单元格
  - 优缺点与访问控制列表类似



- **基于角色的访问控制**
  - 定义角色，并对角色指派访问控制权限，而不是直接对主体
  - 一个角色可能包含若干个主体，一个主体也可能有多个角色
  - 优点：存储高效，降低需要记录的规则数量
  - 缺点：在目前操作系统中未实现



提纲

PART 1 基本概念

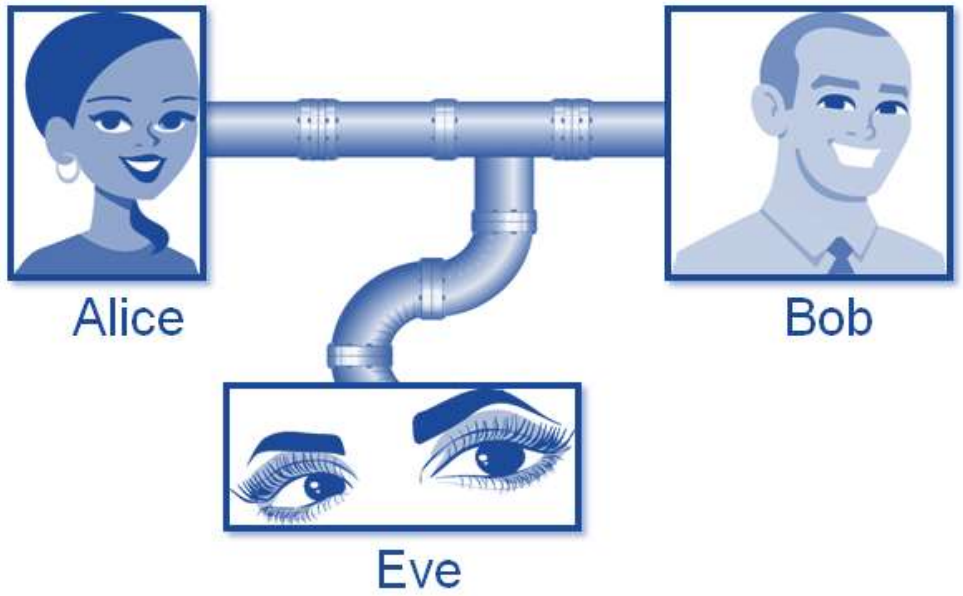
PART 2 访问控制模型

PART 3 密码学相关概念

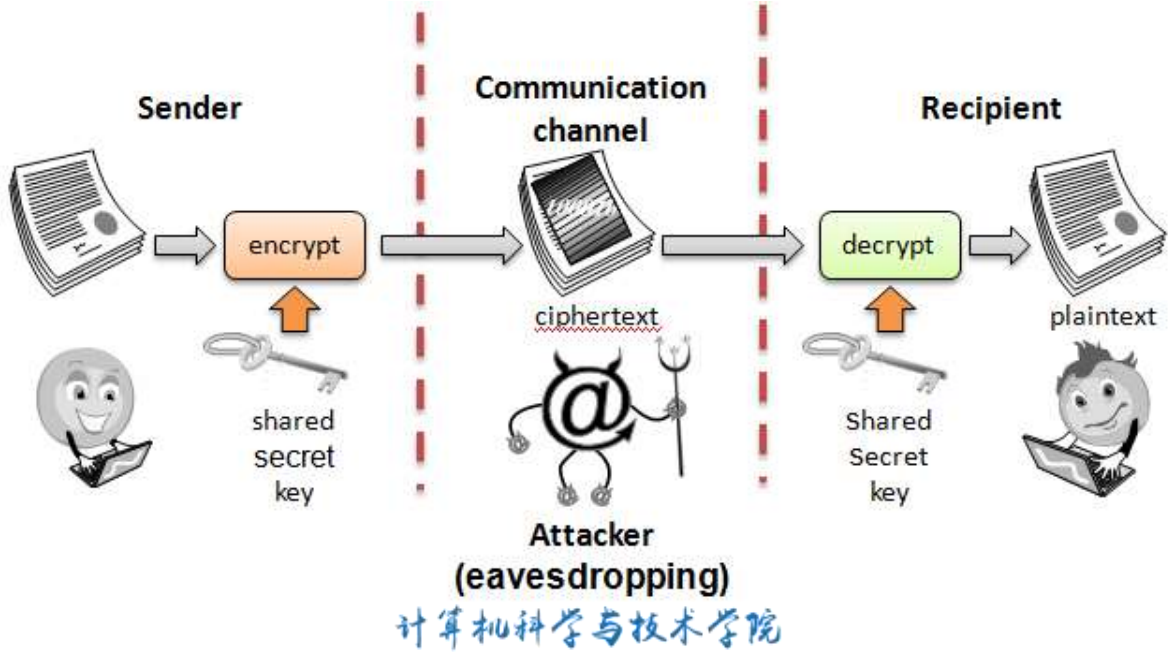
PART 4 实现与可用性问题

PART 5 作业

- **密码学技术是实现安全的主要技术手段**
  - 加密Encryption（密码体制、对称加密、非对称加密）
    - 一种手段使得通信双方(Alice和Bob)在易受窃听的不安全信道上建立保密通信
  - 数字签名、哈希函数、消息认证码、数字证书



- **加密和解密(Encryption, Decryption)**
  - 消息M被称作明文，Alice用加密算法E将明文M转换为密文C输出
  - Bob用解密算法D将密文C转换为明文M
  - 公式表示：$C = E(M), M = D(C)$
  - 加解密算法需要确保除了通信双发外其他人无法从密文C推出明文M



- **密码体制(Cryptosystem)包括以下七部分**
  - 可能的明文集合
  - 可能的密文集合
  - 加密密钥集合
  - 解密密钥集合
  - 加密密钥和解密密钥间的对应关系
  - 所使用的加密算法
  - 所使用的解密算法
- 一般密码体制中，加解密算法是公开的，需要保密的部分是加解密密钥。为什么？

保护密码算法还是密钥?

- **柯克霍夫斯原则（Kerckhoffs's principle）**
  - 在 19 世纪后期，奥古斯特·柯克霍夫斯发表了一篇论文来阐述这个问题。他认为**加密方法不必保密，需要保密的是密钥**。原因有三个：
    - **与维护算法的保密性相比，维护密钥的保密性更容易**。一个密钥可能只有100bit，而安全地保存一个长数千倍的加密算法要难得多。而且算法的细节可能泄露，不止是内部专家，逆向工程也可能学习到一些细节
    - **万一密钥暴露了，Alice 和 Bob 可以很容易地更换密钥；而万一算法被攻破，重新设计一个很好的加密算法要难得多**。事实上，我们需要定期更换密钥，这种情况下保证加密算法的保密性简直不可能
    - **万一有多对人员需要加密其通信，这些人可以使用相同的算法和不同的密钥，而使用不同的程序则太难**

Auguste Kerckhoffs

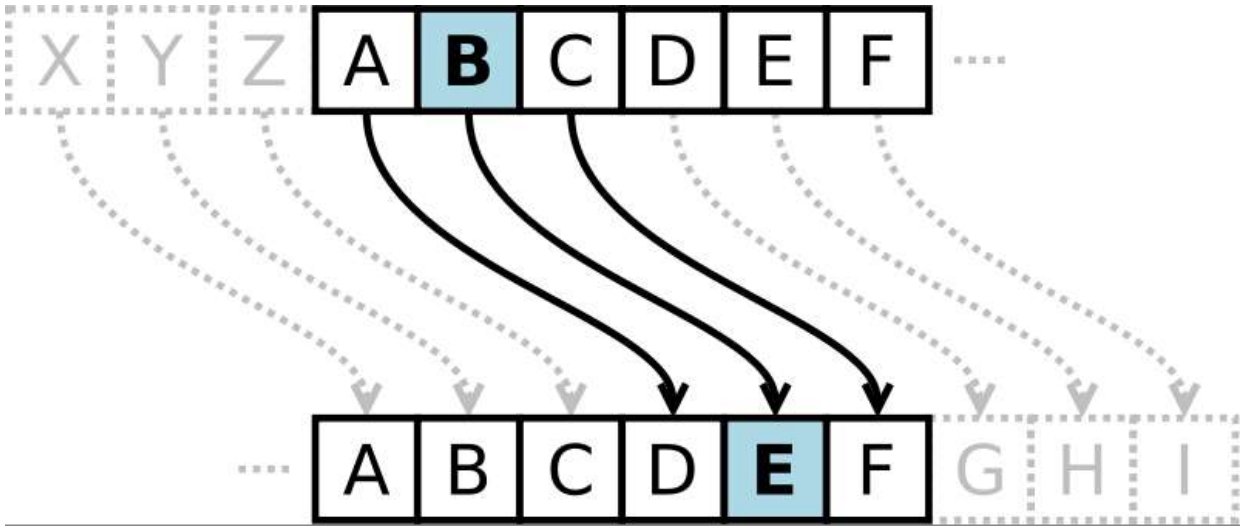
Kerckhoffs's principle was reformulated (perhaps independently) by Claude Shannon as *"The enemy knows the system"*. In that form it is called **Shannon's maxim**.

保护密码算法还是密钥?

- **与之相比，开放密码学设计则有如下四点优势**
  - 公开的设计可以承受公开的钻研和分析，因此可以**更加强壮**。构造良好的密码学方案非常困难，更广泛地被研究可以证明其安全性
  - 公开后有更大的可能被**正义黑客发现**，**比被敌人发现要好**
  - 系统的安全取决于算法的保密性，对**代码的逆向抵抗力很差**。密钥不是代码的一部分，不存在这个问题
  - 公开设计使**标准更容易建立**

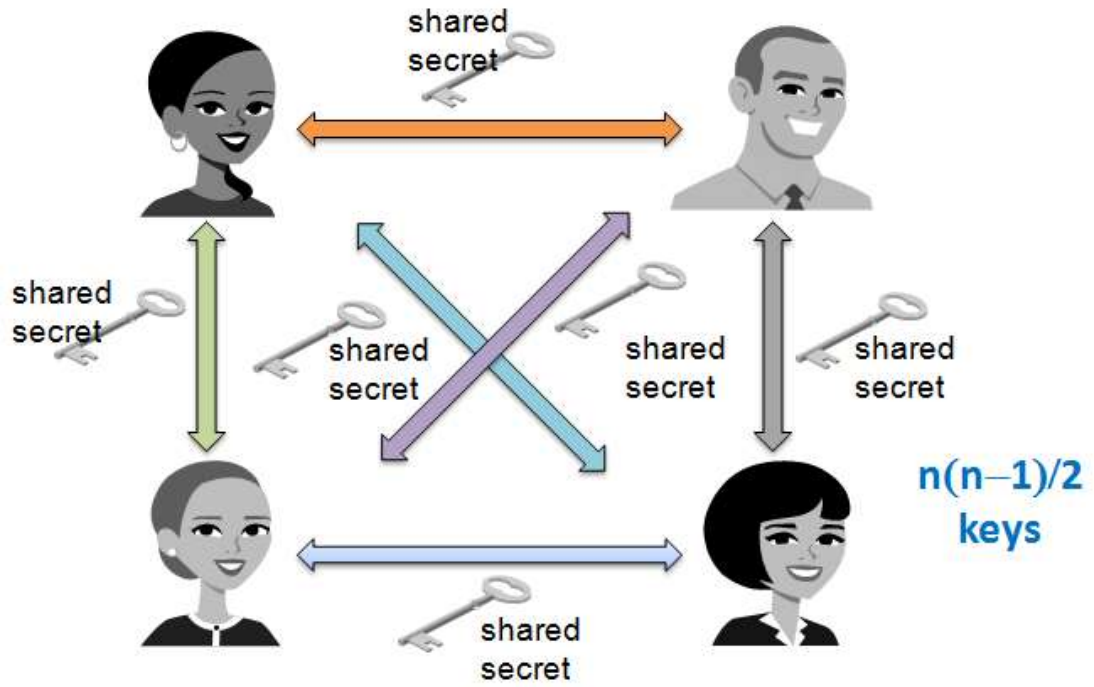
凯撒密码

- 将每个字符替换为字母集中它后面的第三个字符(k = 3)
- 加密：$c = m + k \bmod 26$
- 解密：$m = c - k \bmod 26$

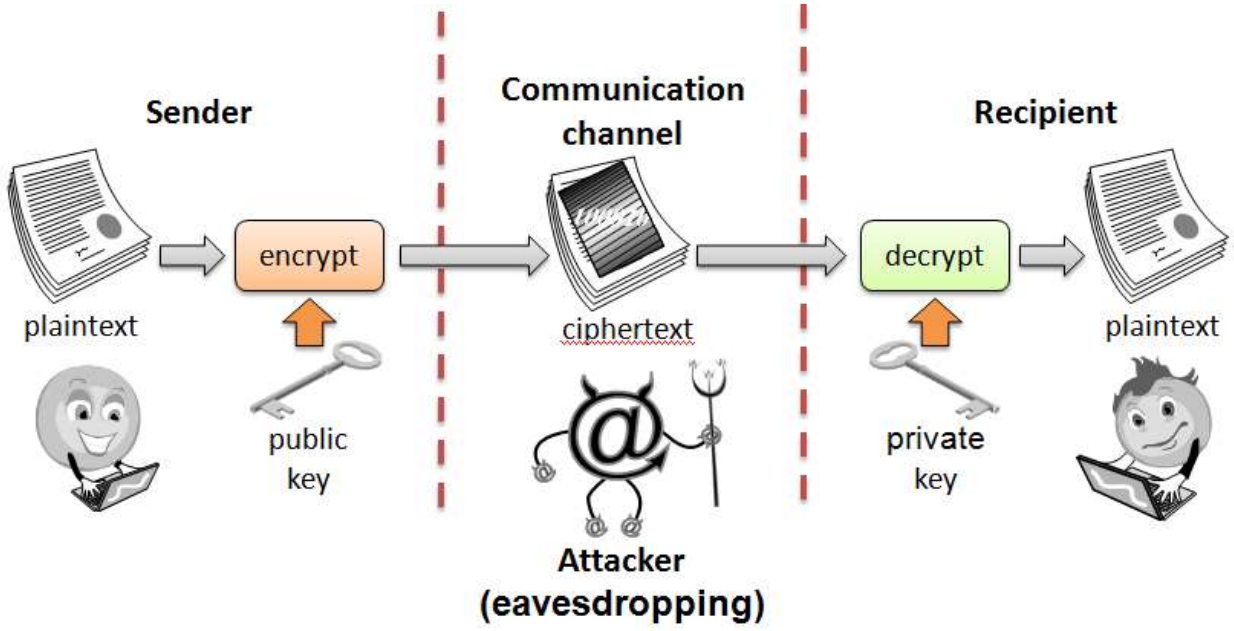


对称密码体制

- **对称加密(Symmetric Encryption)**
  - Alice和Bob共享一个密钥，同时用在加密和解密中
- **对称密钥分发(Symmetric Key Distribution)**
  - 要求每一对通信实体独立的共享一个密钥

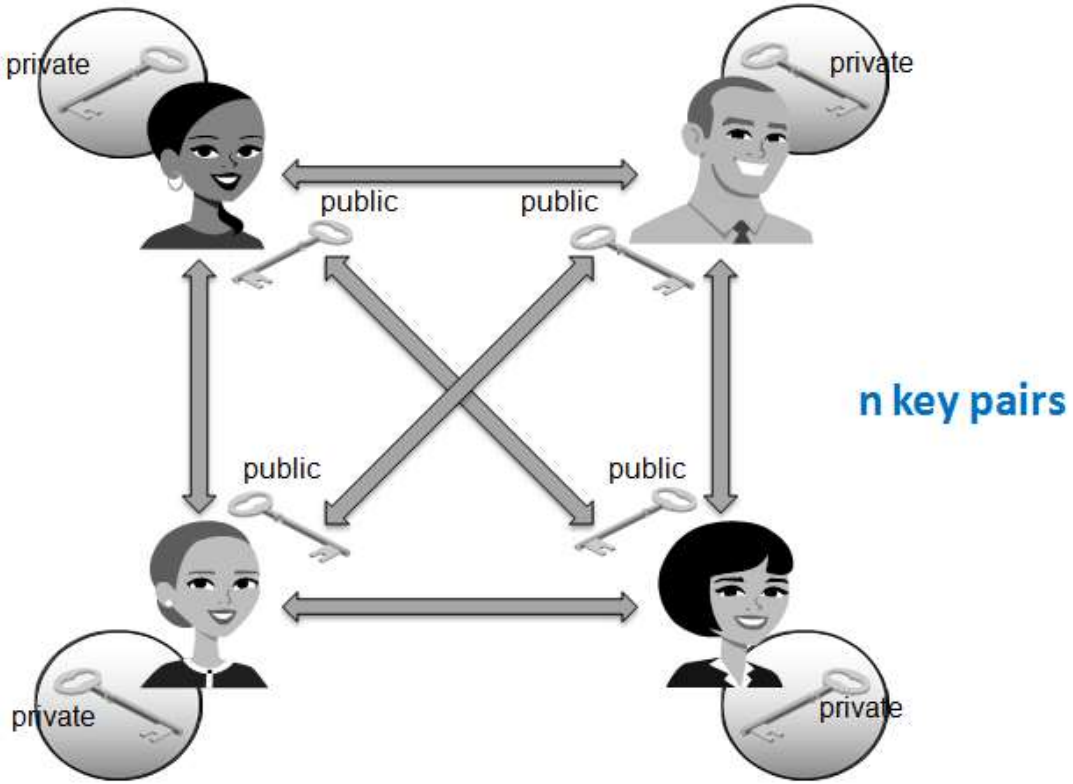


- **公钥加密(Public-key Encryption)**：加密和解密使用不同的密钥
  - Bob有两个密钥，私钥SK_B由Bob保密，公钥PK_B由Bob公开
  - 假设Alice向Bob发送加密的消息，Alice首先需要获取Bob的公钥PK_B，使用PK_B加密消息M：$C = E_{PKB}(M)$，将加密结果C发给Bob，最后Bob使用私钥解密得到明文M： $M = D_{SKB}(C)$



公钥密码体制

- **公钥加密体制密钥分配**
  - 每个接收方仅需要一个密钥
  - 与对称密码体制相比较

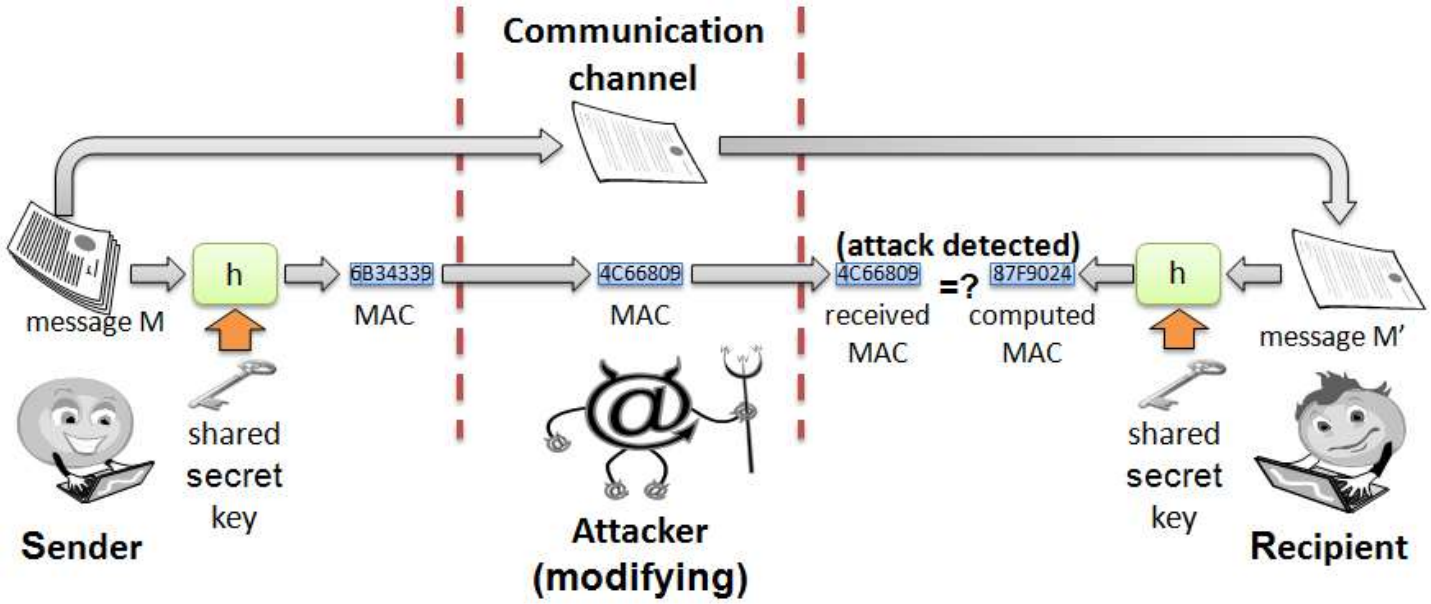


- **公钥加密体制缺点**
  - 加解密复杂度高
  - 密钥长度较长
  - 实际应用中，对称密码体制和公钥体制结合使用

- **数字签名(Digital Signatures)**
  - 公钥加密体制可以用来构造数字签名
  - 数字签名文件的完整性容易验证的，而且数字签名具有不可抵赖性
  - 与传统的纸质签名功能类似
- **数字签名过程**
  - 加解密复杂度高为了对一个消息M签名，Alice用自己的私钥SK_A对M加密：$C = E_{SKA}(M)$.
  - 任何人可以使用Alice的公钥PK_A解密密文C： $M' = D_{PKA}(C)$，并与消息M进行比较
  - 缺点：若待签名消息M很长的话，签名也会与M一样长，导致效率低下

- **密码学哈希函数(Cryptographic Hash Function)**：消息M的一个校验和，需要满足单向性和抗碰撞性
  - 单向性：计算Y=H(M)很容易，但从Y得到M很难
  - 抗碰撞性：很难找到两个消息M和N,使得H(M)=H(N)
  - 哈希函数: SHA-1, SHA-256
  - 将可变长度的消息M映射为定长的二进制消息H(M)
  - 应用1：数字签名中减少代签名消息的长度，首先对消息M应用哈希函数，对其输出应用数字签名算法
  - 应用2：保证操作系统中文件系统的完整性

- **消息认证码(Message Authentication Codes)**
  - 如果Alice和Bob共享一个密钥，MAC允许Alice和Bob之间数据的完整性。
  - 给定消息M，Alice计算H(K||M)并将M和所计算的哈希值发送给Bob
  - 哈希函数Bob用共享的密钥K和接收的消息M′ 计算哈希，并与收到的哈希值进行比较



数字证书

- **数字证书(Digital Certificates)**
  - Alice如何确认获取的Bob的公钥确实是Bob的?
  - 数字证书将用户身份和其公钥绑定并与第三方可信机构(CA)的私钥签名.

对密码体制的简单攻击

- **中间人攻击(Man-in-the-middle Attack)**
  - 对加密机制的中间人攻击
  - 对数字签名的中间人攻击

对密码体制的简单攻击

- **暴力破解攻击(Brute-Force Decryption Attack)**
  - 对于能够识别的密文有效，如果密文是不可识别的，暴力破解无法实施
  - 例如可以用暴力破解来攻击对自然语言的加密
- **对英文文本暴力破解攻击的分析**
  - 假设传送一段英文文本有t个字符，这些字符使用标准的8比特ASCII编码，那么消息长度为$n=8t$比特.
  - 所有可能 n长比特的二进制串有$(2^8)^t=2^n$个。
  - 据估计，英文文本中的每个字符携带的信息大约为1.25比特，那么具有t个字符的英文文本的数量有$(2^{1.25})^t=2^{1.25t}$个。
  - 那么随机选择的n长比特二进制串为一个英文文本的概率为 $\frac{2^{1.25t}}{2^n}=\frac{2^{0.16n}}{2^n}=\frac{1}{2^{0.84n}}$

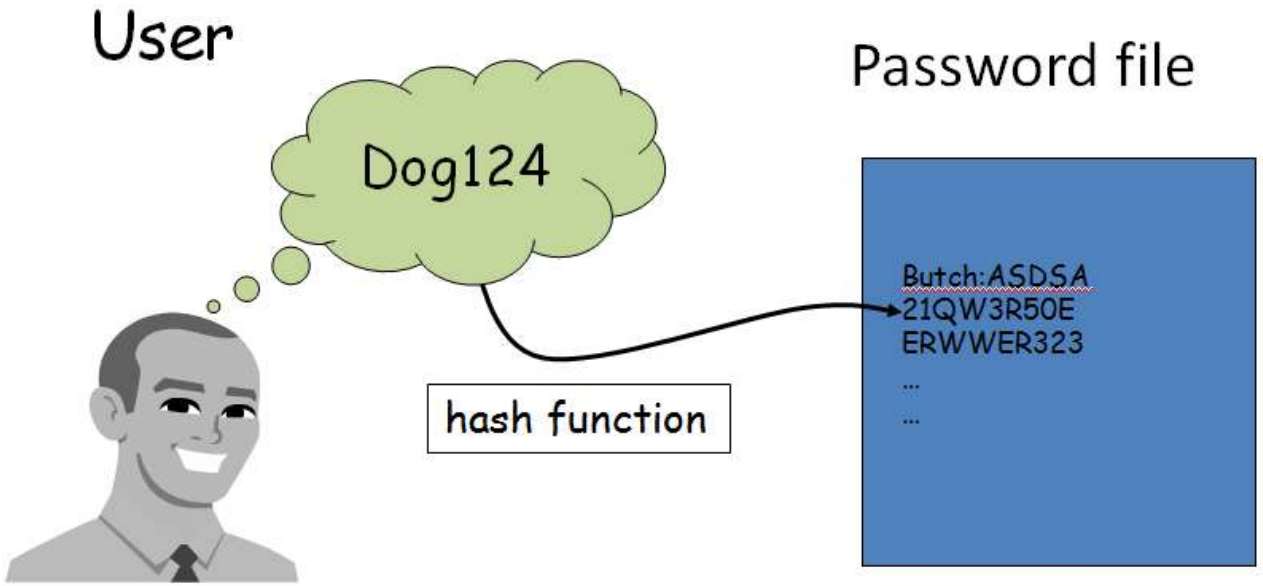
提纲

实现和可用性相关问题

- **实现和可用性相关问题**
  - 效率和可用性
  - 密码(password)
  - 社会工程(Social engineering)
  - 程序代码的脆弱性
- **效率和可用性**
  - 计算机安全解决方案应该高效
  - 用户不喜欢用很慢、且操作复杂的系统
  - 例如：经常用公钥体制进行密钥交换，而在随后通信中采用对称加密
  - 例子：一组学生共同开发一个软件项目中的可用性和访问控制问题

实现和可用性相关问题

- **口令(password)**
  - 一个短的字符序列，使得系统通过一个用户知道的秘密来认证用户
  - Userid: _____
  - Password: _____
- **口令存储：不能以明文形式**



实现和可用性相关问题

- **何为强的口令(strong password)**
  - 混合使用大/小写字符、特殊字符和数字
  - 例如： Seattle1， M1ke03， P@$$w0rd
  - 缺点：容易忘记
- **设置强口令的一个方法**
  - 首先选择自己容易记住的一个句子：

    Mark took Lisa to Disneyland on March 15
  - 然后选取首字母：MtLtDoM15
  - 做一定的修改： MtL+DoM15

实现和可用性相关问题

- **对口令的字典攻击(Dictionary Attack)**
  - 在破解口令时，逐一尝试用户自定义词典中的可能口令的攻击方式。与暴力破解的区别是，暴力破解会逐一尝试所有可能的组合口令，而字典式攻击会使用一个预先定义好的单词列表(可能的口令)
  - 对于英文：有50000个常用单词，1000个常用名，10000个常用姓，1000个典型的宠物名，36525个生日和节日，一共不到100000条目
- **如何防御字典攻击**
  - 口令的设置更加强壮（具有足够长度，含有字母、数字、符号等各种类型），更新更加频繁。这样可以减少被字典攻击猜测成功的几率
  - 采取针对字典攻击更为有效的入侵检测机制，如某个客户端向系统频繁发起认证请求并失败时，系统发出告警并在必要时更换新口令
  - 采用更加健壮的加密算法和策略，使得常规的字典攻击难以生效

实现和可用性相关问题

- **社会工程(Social Engineering)**
  - 指利用人们内心深处的弱点，用计谋来绕过计算机安全解决方案
  - 间谍的三种行为：偷窃、贿赂和勒索同样适用于计算机安全
  - 社会工程攻击是对计算机安全解决方案的最强攻击手段之一
  - 假托(Pretexting)：编造一个故事是管理员信服，从而揭露秘密消息
  - 诱饵(Baiting)：提供一个“好处”让用户或者代理执行不安全的操作
  - 相等补偿(Quid pro quo)：提供一个操作或者服务，并期望有所回报
- 一般来说，社会工程攻击是绕过计算机安全解决方案的一种非常有效的方法。因此，系统设计师在设计安全系统时应该谨记可能的社会工程攻击

实现和可用性相关问题

- **源自编程错误的脆弱性**
  - 再好的设计没有安全的实现，系统依然容易受到攻击
  - 程序员应该知道如何实现安全系统和对安全需求进行形式化描述
  - 社会应该针对所有的安全需求进行测试，特别是对网络通信的程序段和用户提供输入的处理
  - 例如缓冲区溢出攻击，它是通过利用常见的编程错误，即没有检查应用程序读取的输入字符串是否大于缓冲区的长度
  - 攻击者可以利用缓冲区溢出恶意用户编写的代码注入运行的应用程序

提纲

PART 1 基本概念

PART 2 访问控制模型

PART 3 密码学相关概念

PART 4 实现与可用性问题

PART 5 作业

小结

- 安全基本概念
  - CIA、AAA
  - 威胁和攻击、十大安全原则
- 访问控制模型
  - 访问控制矩阵、访问控制列表、能力表和基于角色的访问控制
- 密码学基本概念
  - 加密、数字签名、密码学哈希函数和数字证书
- 实现和可用性问题
  - 效率和可用性、密码、社会工程、程序脆弱性

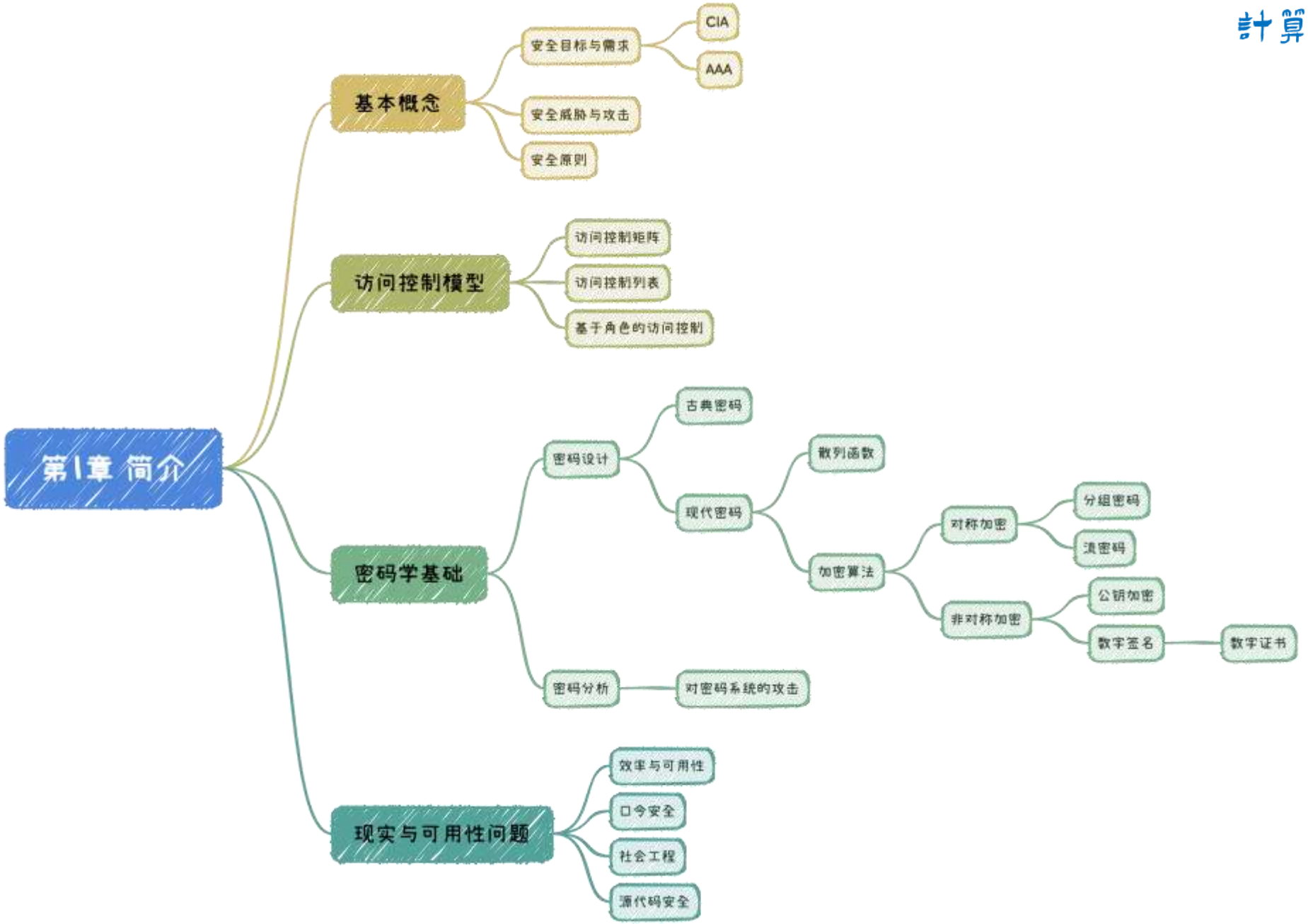
作业

- **思考题**
  - 阅读计算机与网络安全相关资料
- **习题**
  - R-1.1、1.2、1.6~1.11
  - **通过“学在西电”xdspoc.xidian.edu.cn查收作业并提交**

本章结束

~End~

但行好事，莫問前程。
Those that can, do.
Those that cannot,
complain.


第1章 简介
基本概念
安全目标与需求
CIA
AAA
安全威胁与攻击
安全原则
访问控制模型
访问控制矩阵
访问控制列表
基于角色的访问控制
密码学基础
密码设计
古典密码
现代密码
散列函数
加密算法
对称加密
分组密码
流密码
非对称加密
公钥加密
数字签名
数字证书
密码分析
对密码系统的攻击
现实与可用性问题
效率与可用性
口令安全
社会工程
源代码安全