16. 计算机系统安全

计算机系统工程导论 2024

16.2 隔离、身份与访问控制

本节相关的参考文献

1. Woo T Y C, Lam S S. Authentication for distributed systems[J]. Computer, 1992, 25(1): 39-52.
2. Woo T Y C, Lam S S. 'Authentication'revisited (correction and addendum to'Authentication'for distributed systems, Jan. 92, 39-52)[J]. Computer, 1992, 25(3): 10.
3. Dolev D, Yao A. On the security of public key protocols[J]. IEEE Transactions on information theory, 1983, 29(2): 198-208.

填空题 5分

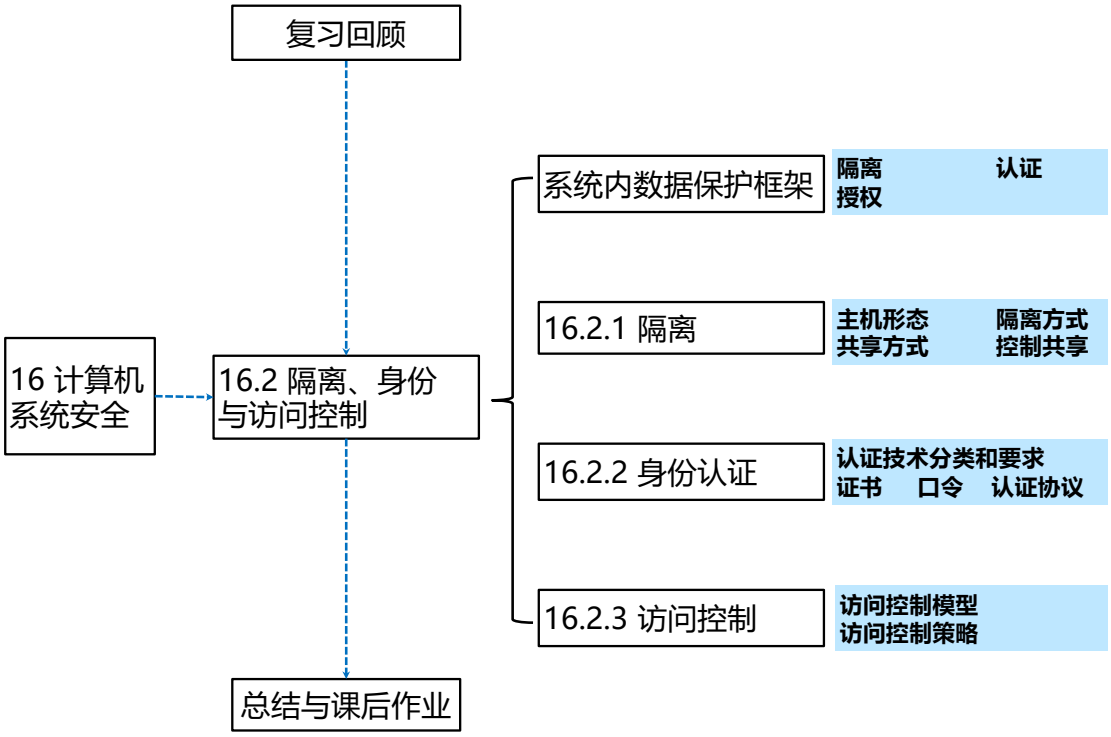
设置

复习回顾（2分钟）

填入括号中备选答案的**数字编号**:

1. 安全与可靠性的差别是：[填空1]（1.安全 2.可靠性）考虑了人为的因素。
2. 保护数据最根本的方法是：[填空2]（1.加密 2.口令 3.隔离）。
3. 将密文转换成明文的过程称为：[填空3]（1.加密 2.解密 3.密钥 4.分析）。
4. 加密密钥与解密密钥不同的算法为：[填空4]（1.对称 2.公钥 3.序列 4.分组）密码算法。
5. TLS的[填空5]（1.握手 2.记录 3.警告 4.心跳）协议用来进行算法和参数的协商。

作答



系统数据保护技术的历史发展

物理隔离是最初和最基本的安全机制 **(1950-1963)**

- 容易: 自己携带数据，自己独占机器，带走所有结果
- 今天

分时复 **3-1982)**

- 困难

安全变得越来越复杂和困难 **(1982-2015)**

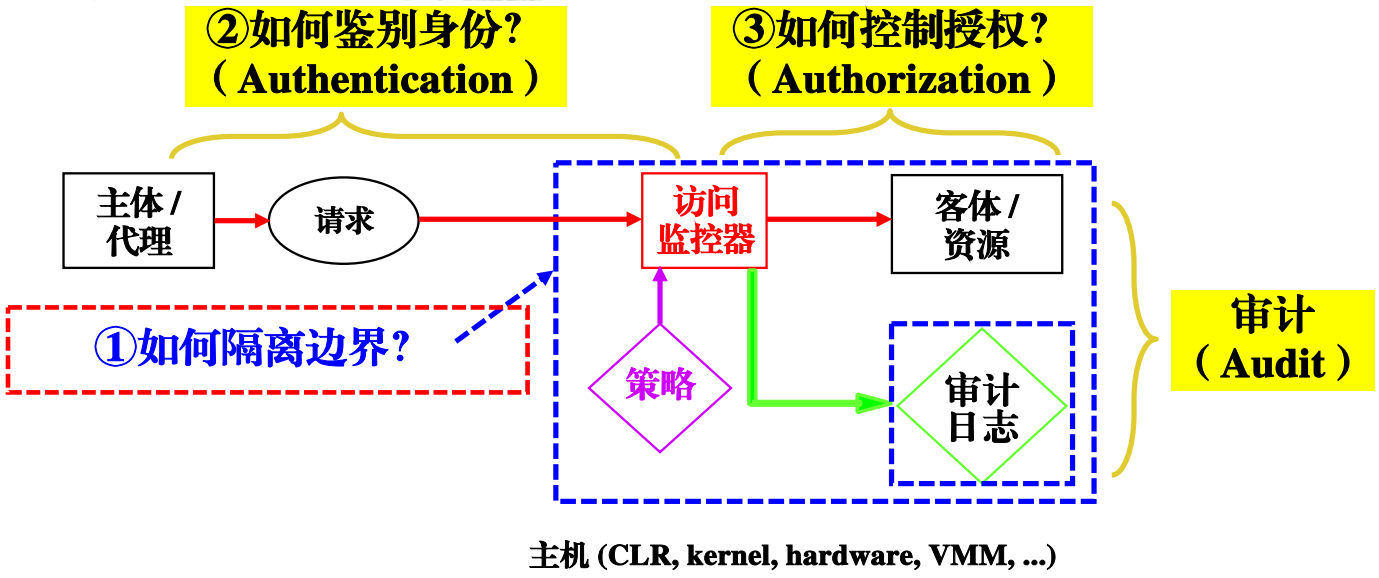
- 隔离越来越少，共享越来越多，中心化管理越来越困难
- 计算机中重要的资产越来越多
- 人们一直在追求完善安全（误区）

系统数据保护的基本问题：

1. 加密

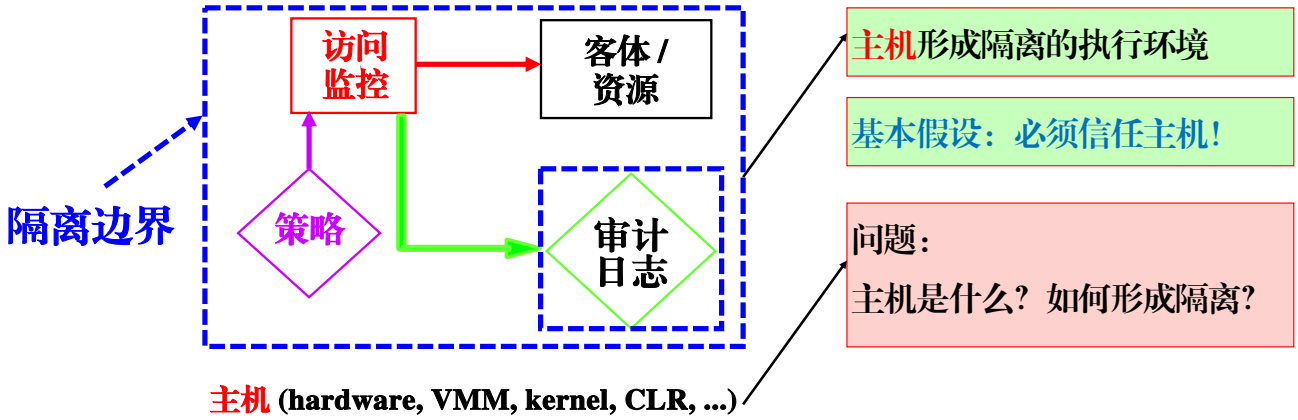
2. 隔离、认证和访问控制

系统内数据保护框架



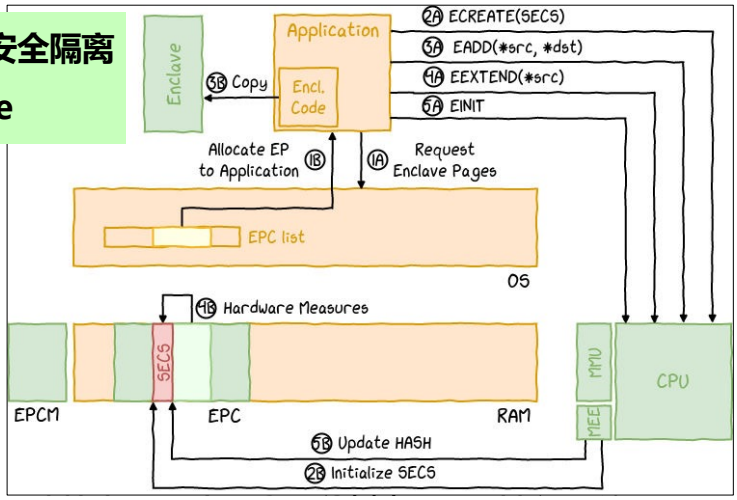
Anderson J P. Computer security technology planning study[R]. ESD-TR-73-51, 1972

16.2.1 隔离



主机的形态与隔离方式

形态	隔离方式
1. 物理机	计算环境硬件安全隔离 enclave
2. 虚拟机	
3. 进程	
4. 代码块	

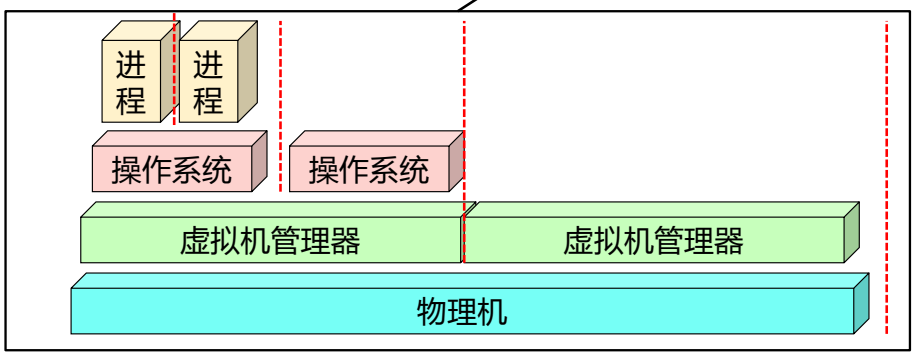


问题：如何在隔离的同时进行共享?

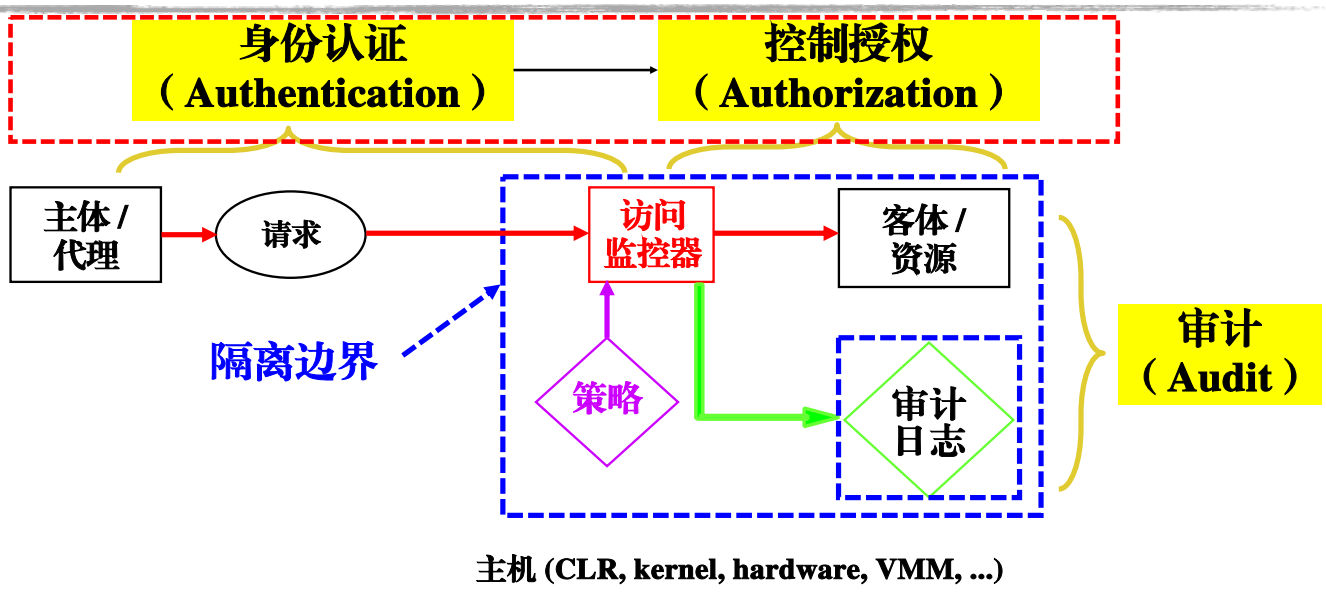
隔离下的共享

1. 物理机之间，用物理链路通信和共享
2. 虚拟机之间，用虚拟链路通信和共享
3. 进程之间，用IPC进行通信和共享
4. 代码块之间，用语言运行库/解释器共享

问题：如何控制共享?



控制共享的通用模式

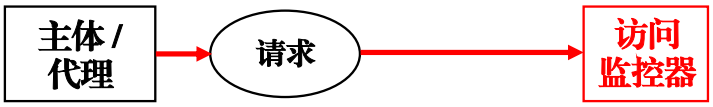


Anderson J P. Computer security technology planning study[R]. ESD-TR-73-51, 1972

认证：证明主体的身份

认证（Authentication）

证明某主体就是其声称身份的过程。

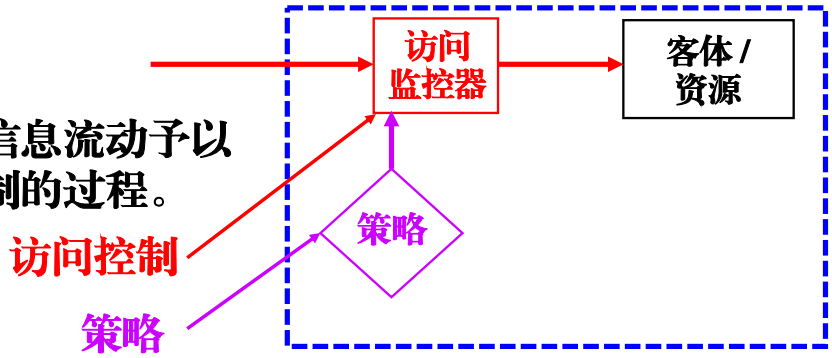


访问监视器可以鉴别访问者的身份

认证（Authentication） 16.2.2介绍

授权：按策略控制信息流动

授权
（Authorization）



对实体间的信息流动予以安全相关控制的过程。

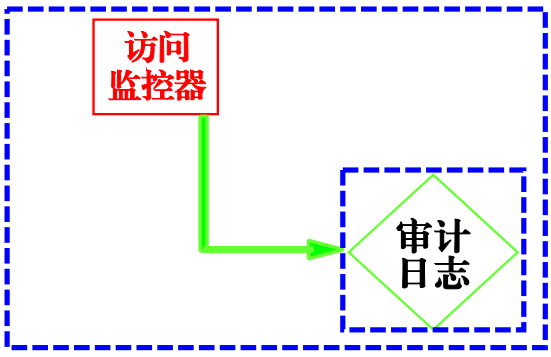
访问控制

策略

主机 (CLR, kernel, hardware, VMM, ...)

授权（Authorization） 16.2.3介绍

审计：发现和追溯问题



对操作记录进行安全相关审查的过程。

审计
（Audit）

审计（Audit） 《网络空间安全导论》课程介绍

16.2.2 身份认证

① **认证技术的分类和要求**

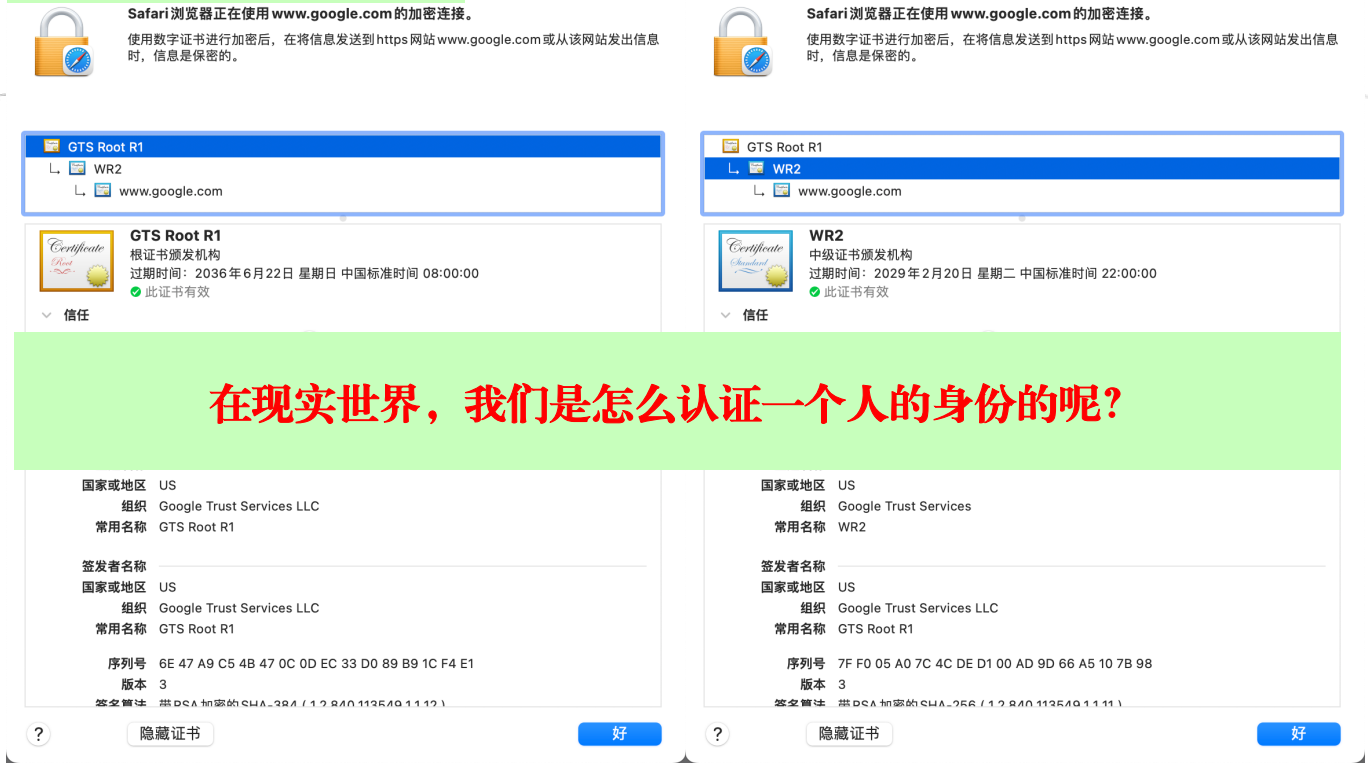
② **基于证书的认证**

③ **基于口令的认证**

④ **认证协议**

身份认证：
证明某主体就是其声称身份的过程。

认证技术的分类和要求



在现实世界，我们是怎么认证一个人的身份的呢？

现实世界的身份认证

1. 知识（know something）
2. 物品（have something）
3. 生物特征（be something）

身份认证： 证明某主体就是其声称身份的过程。

你能举出相应的例子吗？这些例子在什么情况下会失败？

数字世界的身份认证

思考：认证机制通过网络进行，如何实施？有何风险？

1. know sth
2. have sth
3. be sth

环境特点：穿越不安全的区域

风险：可能被监听和重放

↓

重放（replay attack）：重复或延迟使用有效的数据，起到欺骗的目的。

应对思路：每次使用的认证信息都不同

网络认证的威胁假设

中间攻击者的能力

- 数据
  - 能够：窃听、重放、猜测、修改数据
  - 不能：获得终端设备中存放的数据
- 算法
  - 能够：使用已知和可行的分析算法
  - 不能：突破基本密码原语的安全性
- 算力
  - 能够：具有合理算力
  - 不能：使用无限算力

"...the attacker carries the message."

Dolev D, Yao A. On the security of public key protocols[J]. IEEE Transactions on information theory, 1983, 29(2): 198-208.

网络认证的基本要求

认证消息

1. 不可伪造性
2. 新鲜性

认证系统

1. 数据传输完整性与认证相结合
2. 完全介入

16.2.2 身份认证

① **认证技术的分类和要求**

② **基于证书的认证**

③ **基于口令的认证**

④ **认证协议**

互联网的认证需求的复杂性

如何进行大规模的认证?

- 网络上大量的服务器
- DNS和IP都不可靠
- 如何命名？如何确定信任?

1. 命名问题

2. 信任问题

如何进行跨域的认证?

- 互联网用户众多
- 用户来自于各个域：工作、生活、社交
- 如何将身份唯一确定？如何将信任进行传递?

解决命名问题

方案：用公钥命名各类主体（标识） ——————→ **命名问题**

- 不易碰撞，具有唯一性
- 可携带，可穿越管理域
- 可签名、认证，可传递、分发

信任问题

问题：信任

- 如何将**公钥**关联到**可辨识名称**（例如：中国海洋大学）？ 证书
- 如何确认**名称**和**物理实体**关系（例如：申请人的身份）？ 注册

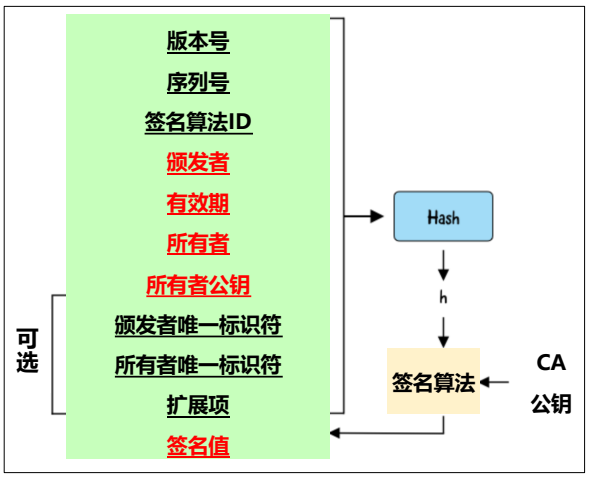
公钥密码证书：数字世界的身份证

例

- 组织证书、网站证书
- 个人证书、设备证书

证书的格式

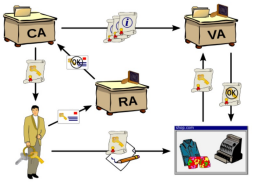
- 所有者
- 公钥
- 签发者
- 签名
- 有效期
- ……



证书的信任体系

问题：如何将公钥关联到名称？如何建立广泛的信任？

1. 基于物理会面的P2P公钥分发
   - 初始信任建立：物理会面，互相信任
   - 信任体系建立：P2P传递方式
   - 特点：成本低、扩展性差、效力弱



公钥密码基础设施PKI
（public key infrastructure）

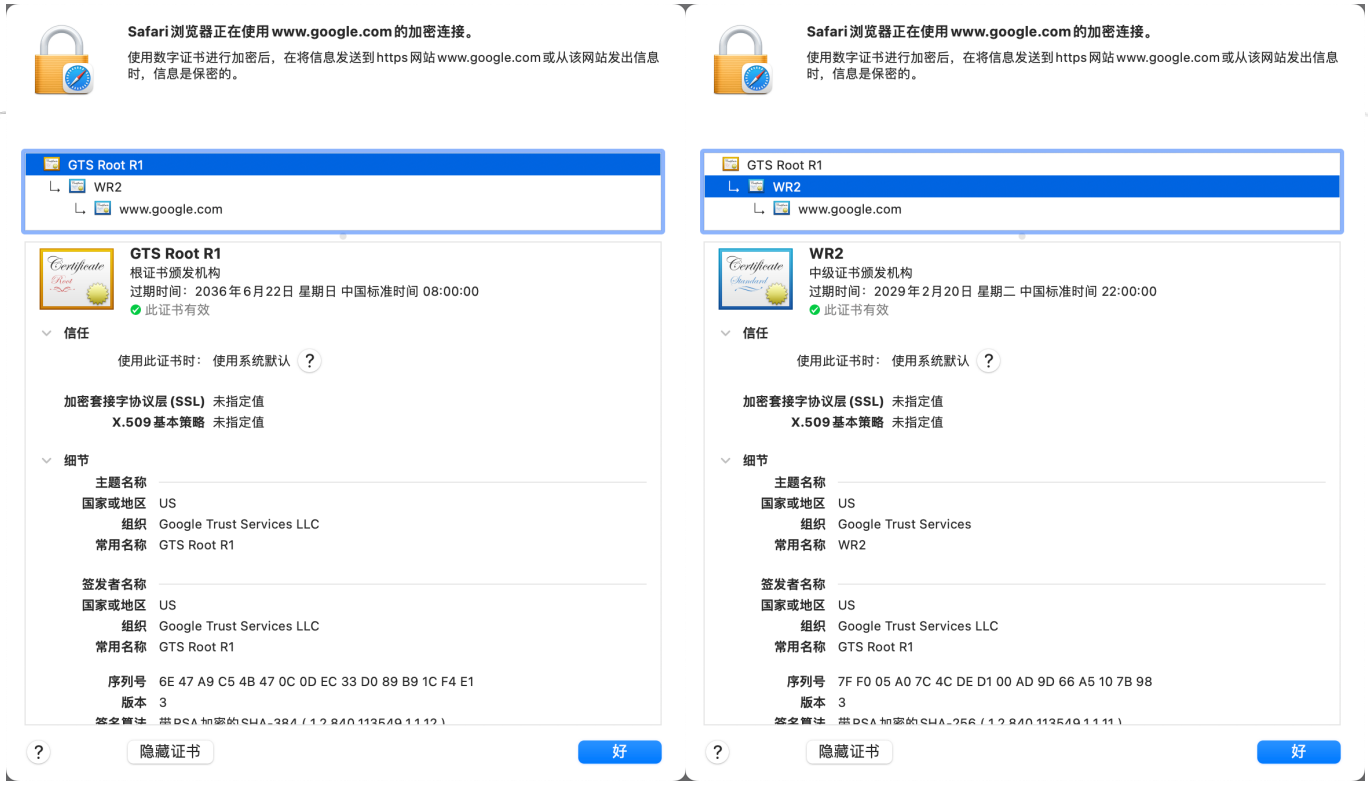
组成

证书库

证书撤销列表

系统

2. 基于权威机构的公钥分发
   - 初始信任建立：证书机构CA（Certificate Authority）颁发
   - 信任体系建立：CA之间建立信任关系
   - 特点：有成本、扩展性好、效力强



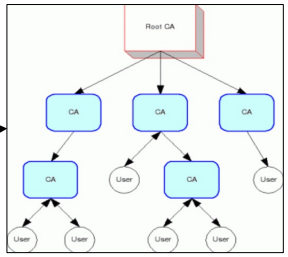
基于PKI的信任链

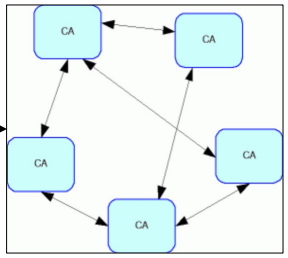
对证书的信任来自于信任链

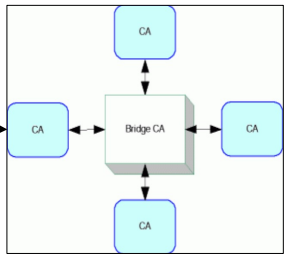
- A 信任 B 信任 C …… 信任 X

CA信任链的建立方式

- 层次
  - 例：政府认证机构、DNSSEC
- 网状
  - 例：商业认证机构
- 桥状
  - 例：FBCA







注册：确认名称和物理实体关系

策略

- 先来先服务（FCFS）
- 安全引导（BootsTrap）
- 行政体系（By Admin）

#讨论，现实例子、安全性/便利性、缺陷?

The Achilles heel

身份认证的秘密丢失了该怎么办?

- 恢复：安全问题？重新注册？email或电话号码?
- 有哪些风险?

16.2.2 身份认证

① **认证技术的分类和要求**

② **基于证书的认证**

③ **基于口令的认证**

④ **认证协议**

口令

口令是最常见的Know Sth认证方式

#思考:

- 过程
- 优点
- 缺点
- 一般事项

问题：如何提升口令机制的安全性?

基于口令的认证方案1

C → Server：我是A，我的口令是P ←—— **问题：窃听**

Server：

- 查询 Name=A且Password=P的记录，如果有则成功

基于口令的认证方案2

C → Server：我是A，我的口令的hash值是H

Server：

- 查询 Name=A且Hash（Password）=H的记录，如果有则成功

问题：窃取

基于口令的认证方案3

C → Server：我是A，我的口令的hash值是H

Server：

- 在数据库中仅保存用户名和口令的Hash值
- 查询 Name=A且Hash=H的记录，如果有则成功

问题：查表

基于口令的认证方案4

C → Server：我是A，我的口令+Salt的hash值是H

Server：

- 在数据库中仅保存用户名和口令+Salt的Hash值
- 查询 Name=A且Hash=H的记录，如果有则成功

问题：重放

口令安全问题

如何防止口令被窃取和暴力猜测?

- 应对窃取
  - Hash √
- 应对猜测
  - 限制尝试次数

什么样的口令容易被猜测? 弱口令

- 为什么弱口令问题不容易解决?

挑战：如何在用户易接受的前提下，改善口令安全?

口令安全性方案1：口令管理器

功能

1. 自动生成、自动填充
2. 高熵口令、多个口令

安全性:

- 本地运行，攻击面少

易用性:

- 用户只需记住1个口令

这一方案的弱点是什么?

口令安全性方案2：双因子

SMS、扫描二维码等

- 安全分析：依赖电话号码真实、基站安全、终端安全、终端在线
- 易受攻击：钓鱼

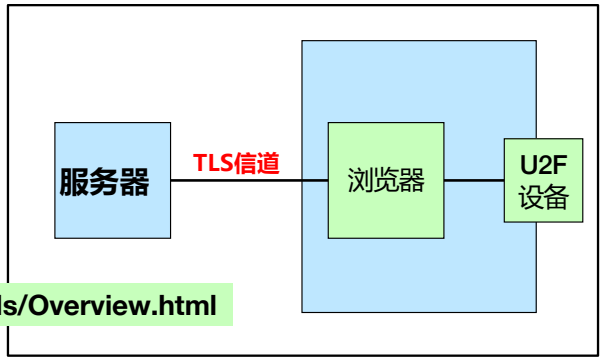
你能举出对应的例子吗?

U2F(Universal 2 Factor）by FIDO（Yubico & Google发起)

- 模式：基于挑战-应答（challenge-response）
- 设备：适配器（dongle）+公私钥对

U2F的挑战-应答 机制

1. **服务器S->浏览器B: 挑战challenge = 随机数**
2. **浏览器B->设备D：挑战challenge**
3. **设备D->浏览器B：对challenge)签名，得到→ 应答s**
4. **浏览器B->服务器S: 应答s**
5. **服务器S: 验证 (challenge, s)**

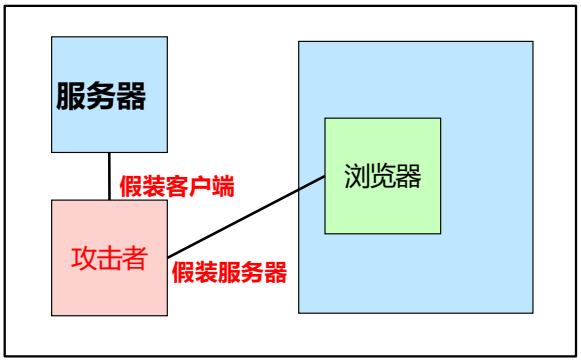


https://developers.yubico.com/U2F/Protocol_details/Overview.html

U2F的中间人攻击和防范

在数据保护和认证之间寻找漏洞

- 攻击者在认证之前，伪装成另一方在中间建立2个TLS会话
- 攻击者在认证的时候，在双方之间转发挑战与应答
- 成功后在转发时窃取和修改信息
- **本质：应答被重用于另外的会话**



如何应对?

- 客户端在应答时，对当前TLS会话的唯一标识也进行签名

为什么有效?

如何产生唯一性的标识?

完整性与认证结合!

16.2.2 身份认证

① **认证技术的分类和要求**

② **基于证书的认证**

③ **基于口令的认证**

④ **认证协议**

简单的鉴别协议（对称）

假设：P、Q存在长期有效的共享密钥k

P　　：消息 m = "我是P"
P　　：对 {m, Q} 加密得m'
P → Q：发送m和m'
Q　　：验证m和m'关系
　　：正确的过则成功

存在什么问题?

被窃听后，m和m'可以用来假冒P

改进的鉴别协议（对称）

假设：P、Q存在长期有效的共享密钥k

P → Q：我是P
Q　　：产生随机数n
Q → P：n
P　　：对 {P, Q, n} 加密得m
P → Q：发送m
Q　　：验证m
　　：正确的过则成功

问题：提前约定密钥，可扩展性差，密钥爆炸……

改进方向：设立认证中心?

Woo–Lam协议

假设：P、Q都与认证中心A存在长期有效的共享密钥kpa和kqa

P → Q：我是P
Q　　：产生随机数n
Q → P：n
P　　：对 {P, Q, n} 使用kpa加密得x
P → Q：发送x
Q　　：对 {P, Q, x} 使用kqa加密得y

Q → A：y
A　　：解密两层得到n
　　：对 {P, Q, n} 使用kqa加密得z
A → Q：z
Q　　：验证z
　　：正确的过则成功

论文：Thomas Y.C. Woo, Simon S. Lam, "Authentication for Distributed Systems", Computer, vol. 25, no. , pp. 39-52, January 1992, doi:10.1109/2.108052

Woo–Lam协议：攻击

假设：P、Q都与认证中心A存在长期有效的共享密钥kpa和kqa

P → Q：我是P
Q　　：产生随机数n
Q → P：n
P　　：对 {P, Q, n} 使用kpa加密得x
P → Q：发送x　假如将x换成n'?
Q　　：对 {P, Q, x} 使用kqa加密得y
{P, Q, n'}　kqa

Q → A：y
A　　：解密两层得到n
　　：对 {P, Q, n} 使用kqa加密得z
A → Q：z　{P, Q, n'}　kqa
Q　　：验证z
　　：正确的过则成功

Dolev-Yao 假设!

重放于n'所在的会话

如何改进?

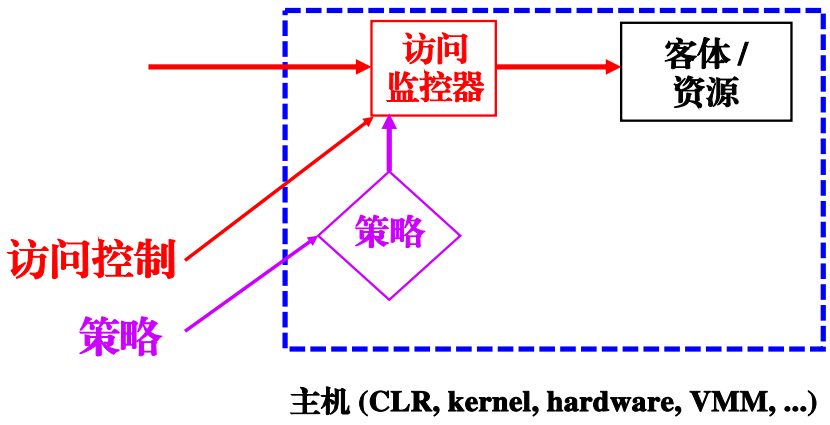
作者自己的修正：S.S. Lam, T.Y.C. Woo, "Authentication Revisited", Computer, vol. 25, no. , pp. 10, March 1992, doi:10.1109/2.121502
维基：https://en.wikipedia.org/wiki/Woo–Lam

Woo–Lam非对称协议（自学）

$$
\begin{array}{lcl}
P \rightarrow Q & : & \text{“I am } P\text{.”} \\
Q & : & \text{generate nonce } n \\
Q \rightarrow P & : & n \\
P & : & \text{compute } m = \{P, Q, n\}_{k_P^{-1}} \\
P \rightarrow Q & : & m \\
Q \rightarrow A & : & \text{“I need } P\text{'s public key.”} \\
A & : & \text{retrieve public key } k_P \text{ of } P \text{ from key database} \\
 & : & \text{create certificate } c = \{P, k_P\}_{k_A^{-1}} \\
A \rightarrow Q & : & P, c \\
Q & : & \text{recover } (P, k_P) \text{ from } c \text{ by decrypting with } k_A \\
 & : & \text{verify } (P, Q, n) \stackrel{?}{=} \{m\}_{k_P} \\
 & : & \text{if equal then accept; otherwise reject}
\end{array}
$$

16.2.3 访问控制

访问控制：对实体间的信息流动予以安全相关控制的过程。



访问控制模型

1）简单防护模型

2）看守者模型

3）信息流控制模型

1）简单防护模型

函数：permissions = policy(subject, object)

矩阵（authorization matrix）：

主体	客体/资源	
	文件 f	数据库 payroll
A	*读, 写*	*写 paychecks*
B	*读*	–

两种实现方式

- 按列实现：list system
- 按行实现：ticket system

谁来规定策略?

- 资源所有者：自主策略（DAC）
- 系统管理员：强制策略（MAC）

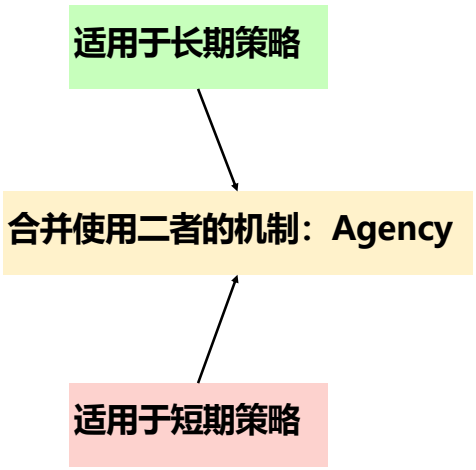
两种实现方式

list system：访问控制列表（ACL）

- 权限存于客体
- 优点：易于集中管理、审计、撤销权限
- 例：文件系统
- 问题：权限检查需查询列表

ticket system：能力（Capability）

- 权限存于主体
- 优点：权限检查较为简单
- 例：文件描述符、对象的名称
- 问题：如何枚举？如何撤销权限？
- 特点：能力可以允许传递（优点/缺点？）



例：ACLs in UNIX

客体	进程通过文件系统API，用路径名访问文件或设备。
主体	内核为进程保存： ruid、 euid、suid 即：真实uid、有效uid、保存的uid
制定策略者	文件的inode保存了所有者的UID，创建时默认为创建者。
访问监视器	所有文件操作，必须通过操作系统内核
实现方式	文件的inode保存了3条按组划分的ACL： 所有者、同组者、其他人，每条ACL包含3个权限位。

谁来规定策略

自主访问控制 (DAC)：

- 客体所有者
- CTSS、商用操作系统

强制访问控制 (MAC)：

- 管理员
- 军事、涉密操作系统

基于角色的访问控制(RBAC):

- 应用程序设计者规定角色的策略
- 管理员分配用户的角色
- 应用信息系统

看守者模型

简单保护模型的面向对象设计

- 特点
  - 能够设定任意访问：不仅仅是read、write和execute
  - 能够设定上下文：时间、访问者属性等
  - 能够设定复杂方案：门限访问等
- 例
  - Chinese-Wall 模型
- 挑战
  - 复杂，从而难以理解、难以正确实现

无论简单模型还是看守者模型，进程都可以在不同客体间传递信息，从而导致被保护的信息泄露。

能解决吗?

信息流控制模型

问题：恶意程序可在文件间复制信息，突破控制边界

- 能否控制信息的流动?

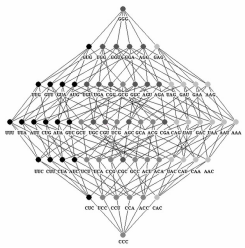
方案：信息流控制

- 动态分析数据在进程中的流动
  - 将进程的内存和通信也纳入到控制中
- 用于军事（TCSEC，1985），也用于个人信息保护（污点追踪）
- 潜信道问题仍没有完全解决

信息流控制的策略

主客体分级分类，级+类组成复合的密级（need-to-know）：

- 分级（全序）：非密→秘密→机密→绝密……
- 分类（偏序）：海军|空军|后勤|空海|……



信息流控制策略

- 将权限组成格(Lattice)模型　(Adept-50; Denning 1976)
- 设定规则，例：
  - 机密性模型：禁止上读、禁止下写 (Bell/LaPadula 1973)
  - 完整性模型：禁止下读、禁止上写（Biba）

《离散数学》回顾：格（lattice）

[序理论的格定义]

- 如果偏序（L,≤）中任意2个元素a和b总存在最大下界(a∧b)和最小上界(a∨b)，则（L,≤）是一个格
  - 上界和下界不一定在格中
- 有界格：包含最大元素（1）和最小元素（0）
  - 任何格 + 最大最小元素 → 有界格

[集合论的格定义]

- 不易理解？想想集合的幂集的元素之间的包含关系

如何分析流动?

直接流动

- b = a

间接流动

- if (a) b
- b = X [a]

新信息的密级

- m = a+b+c+d：在格上取原有信息的各个密级的集合的上确界

问题：不断产生比原有信息更高密级的信息，没有信息降级。怎么解决?

这就体现了秘密信息降级与解密（declassification）的必要性!

安全从业者所面临的挑战：

你以为你以为的，
就是你以为的吗？

突破信息流控制

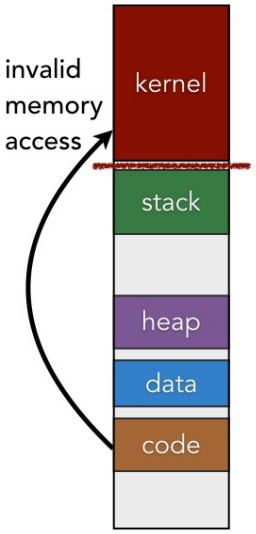
思考：只有数据才能传信息吗?

- 信息是不确定性而非数据，有共享资源就可以传递信息！

潜信道（Covert Channel)

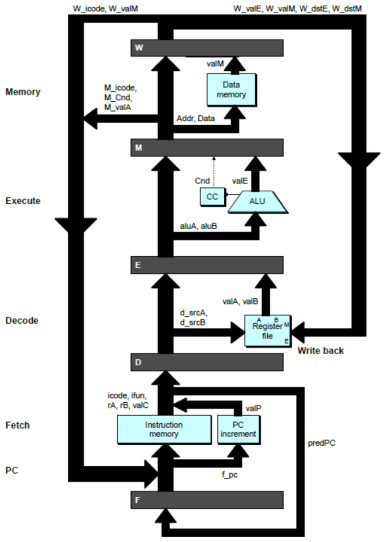
- 如何编码
  - 时间
  - 空间
  - 缓存（时间、空间、效率）

Meltdown 与 Spectre



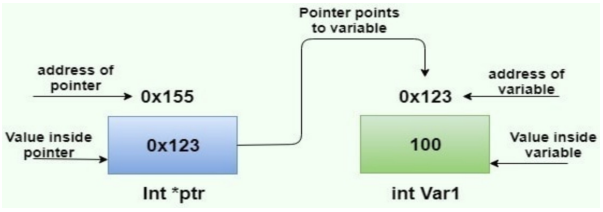
第7章

虚拟内存-内核

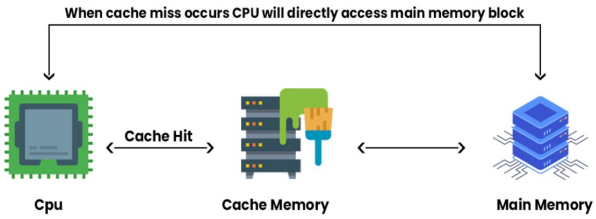


第12章

性能-流水线、推测



第3章 命名-地址类名称



第2章 计算机系统抽象

第12章 快路径/慢路径

现实：全生命周期访问控制的挑战

剩余信息（residue）问题：

- CTSS：内存的复用
- Kerberos：数据包填充携带内存信息
- dean of the Harvard Divinity School 1998：磁盘
- 备份
- 磁残余
- ……

现实：更多的安全问题

可用性问题

- 拒绝服务
  - 使用：botnet（例，Mirai）、放大/反射
  - 攻击：带宽、路由器CPU/内存、服务器内存(状态)/CPU
- 拒绝服务为什么难以防御？如何区分攻击？

实现问题

- 如何实现可信的内核？测试、验证、还是认证？

权衡问题

- 细粒度带来灵活性，但粗粒度易于管理
- Perfect is the enemy of the good!

拓展阅读

Google是如何为巨大的云计算机房实现认证和访问控制的：

- "Google Infrastructure Security Design Overview" by Google Cloud.
  - 通过多种方式实现了最小权限：划分、隔离各个活动
  - 加解密保护数据、数字签名保护完整性
  - 运行了名称服务，关联服务与其公钥
  - 自己设计主板和安全芯片，自己维护数据库：安全芯片的公钥，BIOS、OS等的身份
  - 安全芯片检查BIOS、OS是否有签名，并签署
  - 安全芯片签名BIOS、OS身份，提供给其他服务

课后建议自学

RM的实现

- https://web.ecs.syr.edu/~wedu/seed/Labs/Reference-Monitor/

Setuid安全问题

用BAN逻辑分析认证逻辑的正确性

本节相关的参考文献

1. Woo T Y C, Lam S S. Authentication for distributed systems[J]. Computer, 1992, 25(1): 39-52.
2. Woo T Y C, Lam S S. 'Authentication'revisited (correction and addendum to'Authentication'for distributed systems, Jan. 92, 39-52)[J]. Computer, 1992, 25(3): 10.
3. Dolev D, Yao A. On the security of public key protocols[J]. IEEE Transactions on information theory, 1983, 29(2): 198-208.

16.2 结束