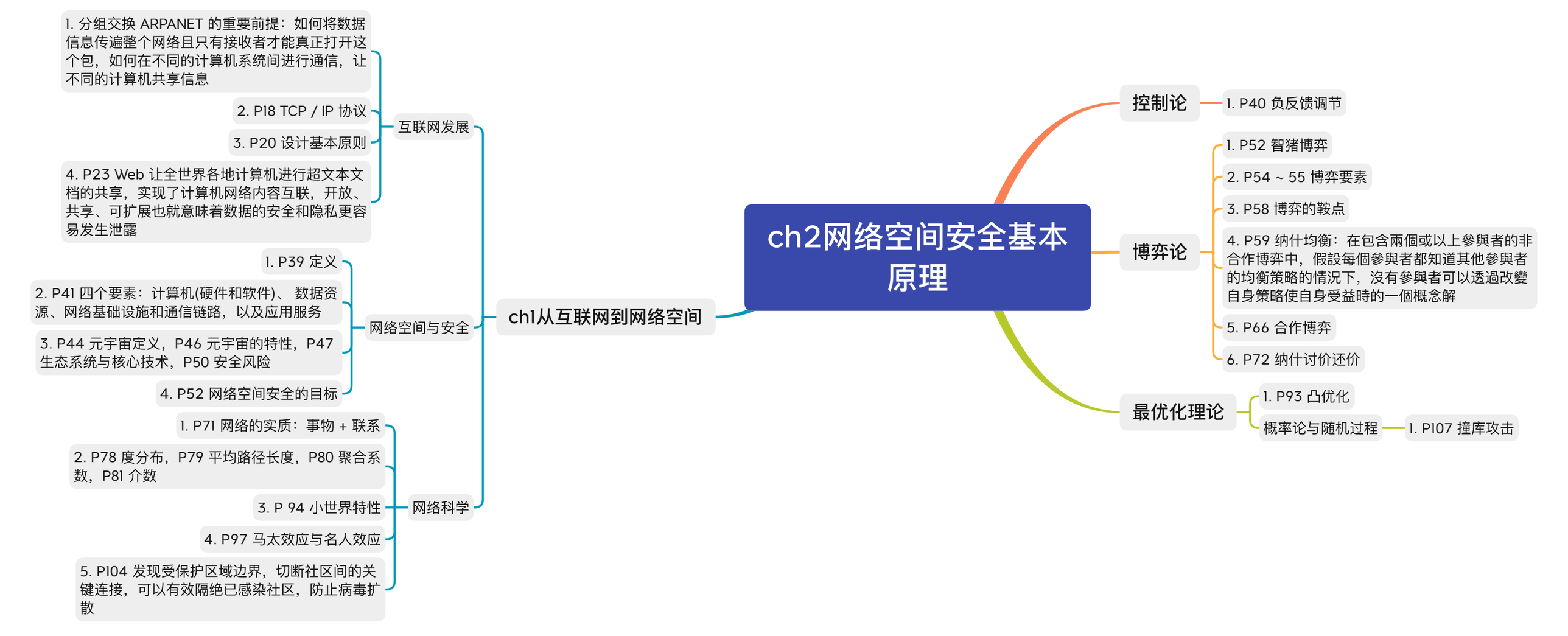

ch2网络空间安全基本原理
控制论
1. P40 负反馈调节
博弈论
1. P52 智猪博弈
2. P54 ~ 55 博弈要素
3. P58 博弈的鞍点
4. P59 纳什均衡：在包含兩個或以上參與者的非合作博弈中，假設每個參與者都知道其他參與者的均衡策略的情況下，沒有參與者可以透過改變自身策略使自身受益時的一個概念解
5. P66 合作博弈
6. P72 纳什讨价还价
最优化理论
1. P93 凸优化
概率论与随机过程
1. P107 撞库攻击
ch1从互联网到网络空间
互联网发展
1. 分组交换 ARPANET 的重要前提：如何将数据信息传遍整个网络且只有接收者才能真正打开这个包，如何在不同的计算机系统间进行通信，让不同的计算机共享信息
2. P18 TCP / IP 协议
3. P20 设计基本原则
4. P23 Web 让全世界各地计算机进行超文本文档的共享，实现了计算机网络内容互联，开放、共享、可扩展也就意味着数据的安全和隐私更容易发生泄露
网络空间与安全
1. P39 定义
2. P41 四个要素：计算机(硬件和软件)、 数据资源、网络基础设施和通信链路，以及应用服务
3. P44 元宇宙定义，P46 元宇宙的特性，P47 生态系统与核心技术，P50 安全风险
4. P52 网络空间安全的目标
网络科学
1. P71 网络的实质：事物 + 联系
2. P78 度分布，P79 平均路径长度，P80 聚合系数，P81 介数
3. P 94 小世界特性
4. P97 马太效应与名人效应
5. P104 发现受保护区域边界，切断社区间的关键连接，可以有效隔绝已感染社区，防止病毒扩散

Chap3——网络空间安全基本机制

沙箱

- 发展概况
  - **20世纪70年代**: 早期的沙箱技术开始出现。
  - **20世纪80年代至90年代末**: 沙箱技术得到了进一步的发展和应用。
  - **2000年以后**: 随着互联网的普及和技术的进步，沙箱技术得到了广泛的应用和发展。
- 安全目标
- 沙箱的基本思想和原理
  - **沙箱环境**: 一个受限的、与外部隔离的操作系统环境。
  - **程序的运行**: 在沙箱内部运行的程序只能对沙箱内部的资源进行访问，对沙箱外部的资源的访问则被限制。
  - **安全规则配置**: 通过配置安全规则，可以控制程序能够使用的资源集，例如内存空间，文件系统空间，网络等资源。
- 沙箱与其他技术的关系
  - **软件错误隔离**: *沙箱可以看成是软件错误隔离思想在网络防御中的应用*。软件错误隔离的目标是利用软件手段限制不可信模块造成的危害，通过隔离保证系统鲁棒性。
  - **访问控制**: 从访问控制的角度看，*沙箱的本质是面向程序的访问控制*。基于访问控制，沙箱可以限制程序的资源访问能力，既满足其正常的访问需求，又保证整体系统安全。
  - **虚拟化**: *沙箱也可以被视为虚拟化技术的一种特定实例*。虚拟化技术能够模拟完整的主机，如微软2019年推出的Windows Sandbox（又叫Windows沙盒）就是一种轻量化的虚拟机，它基于Windows容器技

入侵容忍

- 发展概况
  - **20世纪80年代**: 入侵容忍技术开始出现。
  - **20世纪90年代末**: 入侵容忍技术得到进一步的发展和应用。
  - **2000年以后**: 随着互联网的普及和技术的进步，入侵容忍技术得到了广泛的应用和发展。
- 安全目标
- 基本思想和原理
- 入侵容忍的安全能力和核心机制
  - **阻止和预防攻击**
  - **检测攻击、评估攻击造成的危害**
  - **在遭受攻击后，及时维护和恢复关键数据、关键服务或完全服务**
  - **安全通信机制**
  - **入侵检测机制**
  - **入侵遏制机制**
  - **错误容忍和处理机制**
- 互联网设计原则

可信计算

- 发展概况
  - **1999年**: 可信计算平台联盟（Trusted Computing Platform Alliance, TCPA）成立。
  - **2003年**: 可信计算组织（Trusted Computing Group, TCG）成立，主推可信计算发展。
  - **2006年**: 可信平台模块（Trusted Platform Module, TPM）被广泛接受和应用。
  - **2007年以后**: 可信计算的理念和技术得到进一步推广和应用。
- 安全目标
- 基本思想和原理
- 关键技术概念
  - **背书密钥**
  - **安全输入和输出**
  - **内存屏蔽/受保护的执行**
  - **密封存储**
  - **远程认证**
  - **可信第三方**

移动目标防御

- 发展概况
  - **1970年代**: 初期的移动目标防御概念出现。
  - **2009年**: 移动目标防御技术逐渐受到关注。
  - **2011年**: 移动目标防御技术逐步实践应用。
  - **2014年至今**: 移动目标防御成为重要的网络安全防护手段。
- 安全目标
- 基本思想和原理
  1. **动态网络**: 通过不断在网络系统的多个配置之间进行转移变换，例如更改开放的网络端口，网络配置，软件等。
  2. **增加随机性和不确定性**: 通过在网络、平台、环境、软件和数据等多个层次增加随机性和不确定性，从而增加攻击难度、有效削弱攻击者对防御机制的适应和突破能力。
  - IP地址跳变
  - 端口跳变
  - 动态路由
  - 网络和主机身份随机化
  - 地址空间随机化
  - 指令集合随机化
  - 数据存放形式随机化

拟态防御

- 发展概况
  - **2013年**: 拟态防御理论开始产生。
  - **2016年**: 拟态防御理论得到进一步的发展。
  - **2018年**: 拟态防御理论得到广泛的应用和认可。
- 安全目标
  - **扰乱或阻断未知漏洞或后门利用的攻击链**，缩短外部攻击者和内网攻击者嗅探系统特征及规律的时间窗口，作为倍增器放大传统安全措施的效能。
  - **大幅增加了漏洞的利用难度，降低了攻击的有效性**；通过对未知漏洞或后门进行主动防御，有效抑制"有毒带菌"底层构件造成的安全威胁，解决不确定威胁的问题。
- 基本思想和原理
  - 通过网络、平台、环境、软件、数据等结构的主动跳变或快速迁移来实现**动态变化**、弹性可靠的拟态环境。
  - 扰乱攻击链的构造，使攻击的代价倍增，**难以生效**。
  - 对攻击者则表现为难以观测、无法预测，大幅度增加包括未知的可利用的漏洞和后门在内的攻击难度和成本，对**不确定性威胁形成主动防御**。
- 拟态安全主动防御体系基础架构
  - 以异构性、多样或多元性改变目标系统的**相似性、单一性**。
  - 以动态性、随机性改变目标系统的**静态性、确定性**。
  - 以异构冗余多模裁决机制识别和屏蔽**未知缺陷与未明威胁**。
  - 以高可靠性架构增强目标系统服务功能的**柔韧性或弹性**。
  - 以系统的**不确定属性防御针对目标系统的不确定性威胁**。
  - 随机化和多样化内核——动态异构冗余构造

Chap4

对称密码

- 对称密码算法的两条基本设计原则 (P.35)
  - 1. **扩散（Diffusion）**：重新排列消息中的每一个比特，使明文中的冗余度能够扩散到整个密文，将每一个比特明文的影响尽可能作用到较多的输出密文位中。
  - 2. **扰乱（Confusion）**：密文和密钥之间的统计特性关系尽可能复杂化。如果密钥的一位发生变化，密文的绝大多数位也发生变化。这两种设计原则是由信息论的创始人克劳德·艾尔伍德·香农提出的。
- 分组密码 (P.37)
  - ECB模式 (P.38)
  - CBC模式 (P.40)
  - CFB模式 (P.41)
  - OFB模式 (P.42)

DES(年年讲年年烂，建议看课件）

公钥密码

- **事先共享密钥**：在加密通信之前，以安全的方式将密钥交给接收方。
- **通过密钥分配中心（Key Distribution Center，KDC）**：当参与加密通信的人过多时，可以使用KDC。当需要进行加密通信时，KDC会生成一个通信密钥，每个人只需和KDC事先共享密钥即可。
- **通过Diffie-Hellman密钥交换**：进行加密通信的双方仅需要交换一些信息，即使这些信息被窃听，也无法对算法进行解密，而通信双方则可以通过这些信息各自生成相同的密钥。
- **通过公钥密码**：由于加解密使用不同的密钥，所以无需进行密钥分发。
- 公钥密码系统的加密原理 (P.64-65)
- 公钥密码系统的签名原理 (P.66)
- 公钥密码算法的表示 (P.67)
  - **对称密钥密码**
    - 密钥：会话密钥(Ks)
    - 加密函数：EKs[P]
    - 对密文C，解密函数：P=DKs[C]
  - **公开密钥**
    - (KUa，KRa)
    - 加密/签名：C= EKUb[P]，EKRa[P]
    - 解密/验证：P= DKRb[C]，DKUa[C]
- 对公开密钥密码算法的要求 (P.68-69)
  - 参与方B能容易地生成密钥对(KUb, KRb)。
  - 已知KUb，A的加密操作应该是容易的：C= EKUb[P]。
  - 已知KRb，B的解密操作应该是容易的：P= DKRb (C) =DKRb (EKUb (P))。
  - 已知KUb，求KRb应该是在计算上不可行的。
  - 已知KUb和C，欲恢复P应该是在计算上不可行的。
- 公钥密码算法的误解 (P.603)
  - 1. 公开密钥算法比对称密钥密码算法更安全：事实上，任何一种算法的安全性都依赖于密钥长度和破译密码的工作量。从抗分析的角度来看，没有哪一种算法是绝对优越的。
  - 2. 公开密钥算法使对称密钥成为过时的技术：事实上，公开密钥算法运行速度较慢，只能用在密钥管理和数字签名上。因此，对称密钥密码算法仍将长期存在。
  - 3. 使用公开密钥加密，密钥分配非常简单：事实上，密钥分配既不简单也不有效。

摘要与签名

- 散列函数 (P.80)
- 散列函数的性质 (P.82)
  - 1. 可以根据任意长度的消息计算出固定长度的散列值。
  - 2. 能够快速计算出散列值。
- 散列函数的安全性：HASH 碰撞 (P.84)
- 哈希算法实例 (P.87)
  - **MD5**: 生成128位的哈希值，广泛用于确保信息传输的完整无误。
  - **SHA-1**: 安全哈希算法，主要用于各种安全认证中，生成160位的哈希值。
  - **SHA-256**: 属于SHA-2标准下的哈希算法，生成256位的哈希值，被广泛用于比特币中。
- 散列函数的应用: 比特币 (P.88)
- SHA256算法 (P.89)
- 消息认证码 (MAC) (P.97)
- HMAC (P.97)
- 公钥密码与数字签名 (P.98)
  - 数字签名方法 (P.100)
  - 数字签名的分类 (P.101)
  - 数字签名的应用 (P.102)
  - 密码分析技术 (P.105)
  - 唯密文攻击(Ciphertext only) (P.106)
  - 已知明文攻击(Known Plaintext) (P.106)
  - 选择明文攻击(Chosen Plaintext) (P.106)
  - 选择密文攻击(Chosen Ciphertext) (P.106)
  - 相关密钥攻击(Related Key) (P.106)
  - 密码分析技术 (P.107)

密码学简史

- **古典密码**：主要是通过替换和移位等简单手段来加密信息。
- **近代密码**：包括复杂的机械或电子密码机，如二战期间的德国恩尼格玛密码机。
- **现代密码**：主要指的是计算机时代的密码技术，比如公钥密码和哈希函数等。
- 隐写术与密码学
  - **隐写术**：源自希腊语，意为"隐秘书写"。它是一门关于信息隐藏的技巧和科学，主要是将信息隐藏在其他无害的信息中，以达到传递秘密信息的目的。
  - **密码学**：源自希腊语，意为"隐藏的书写"。它是一门研究如何隐密地传递信息的学科。密码学的目的不仅仅是隐藏信息，而是确保信息的安全性，包括保密性、完整性和可用性等。
  - 密码学的基本概念
    - **密码编码**：通过信息编码使信息保密。
    - **密码分析**：用分析方法解密信息。
    - **基本术语**：
      - 明文(plain text)：原始要加密的信息。
      - 密文(cipher text)：经过加密的信息。
      - 加密(encrypt, encryption)：将明文转化为密文的过程。
      - 解密(decrypt, decryption)：将密文转化为明文的过程。
      - 密码算法(Algorithm)、密码(Cipher)：用来加密和解密的数学函数。
      - 密钥(Key)：密码算法中的一个变量。
    - **代替**：每个明文元素映射为另一个元素。
    - **换位**：明文中的元素重新排列。
    - **对称加密**：同一个密钥既用于加密又用于解密。
    - **非对称加密**：使用一对密钥，一个用于加密，另一个用于解密。
    - **块加密**：一次处理一个数据块进行加密。
    - **流加密**：一次处理一个数据单元进行加密。
  - 密码技术的主要用途
    - **数据保密**：通过数据加密和解密保护数据的私密性。
    - **认证技术**：实体身份认证和数据源发认证。
    - **信息完整性保护**：保证信息在传输过程中没有被插入、篡改、重发。
    - **数字签名和抗抵赖**：源发抗抵赖和交付抗抵赖。
- 古典密码
  - **代替密码 (Substitution Cipher)**
  - **换位密码 (Transposition Cipher)**
  - **代替密码与换位密码的组合**
  - 受限算法的缺陷
  - 不适合大规模使用
  - 不适合人员变动较大的组织
  - 用户无法了解算法的安全性
  - 例子：凯撒密码（PPT第17页）
  - 例子：猪圈密码（PPT第18页）
  - 例子：纵行换位密码（PPT第19页）
- 1949年~1975年：戚继光反切码（PPT第20-21页）
- 近代密码 (P.22)
  - ENIGMA机 (P.23-P.26)
    - **工作原理** (P.24-P.26): 它的工作方式基于三个核心要素：转子的初始设置，转子之间的相互位置，以及连接板连线的状况，这三者组合构成了所有可能的密钥。此外，由于每个转子的旋转，使得可能的加密变化在每次输入时都会变化。
    - **工作过程** (P.27): ·根据密码本取得当日密钥·首先发送一个新的密钥·随机地选择三个字母，比如说PGH·把PGH在键盘上连打两遍，加密为比如说KIVBJE（注意两次PGH被加密为不同的形式）·把KIVBJE放在在电文的最前面·重新调整三个转子的初始方向到PGH·正式对明文加密
  - 对称密钥密码系统的缺陷 (P.31)
    - 1. 密钥必须经过安全的信道分配。
    - 2. 无法用于数字签名。
    - 3. 密钥管理复杂，随着用户数量的增加，密钥的数量将呈指数级增长，这是因为每个用户都需要与每个其他用户有一个唯一的密钥。
  - 现代密码 (P.31)
  - 量子密码 (P.32-P.33)
    - **工作原理**：量子密码利用光的偏振状态来编码信息。例如，"水平垂直方向"偏振和"对角方向"偏振可以被用来代表二进制的0和1。
    - **量子密钥生成方法**：在量子密码中，发送方生成并发送偏振，接收方则进行测量。由于量子力学的原理，任何未经授权的测量都会改变量子态，从而被检测到。

RSA算法简介 (P.70)

- RSA算法的诞生 (P.71)
- RSA算法的性能表现 (P.73, 74)
- RSA算法的操作过程 (P.604-608)
- 公钥密码系统的应用 (P.77)

- Chap5
  - 隐私保护技术初探
    - 隐私保护技术概述
    - 面向数据发布的隐私保护
      - 1. **基于限制发布的隐私保护**：这种技术在将数据公布给数据挖掘者之前，对数据进行扰动、加密、匿名等处理，将数据中的隐私藏起来。研究主要集中于数据匿名化，例如有选择的发布原始数据、不发布或者发布精度较低的敏感数据。
      - 2. **基于数据失真的隐私保护**：该技术通过对原始数据进行扰动，目的是隐藏真实数据，只呈现出数据的统计学特征。失真后的数据满足两个条件：保持原本的某些特性不变，且攻击者不能根据失真数据重构出真实的原始数据。此技术主要包括随机化、阻塞、变形、交换等。
      - 3. **基于数据加密的隐私保护**：这种技术对原始数据进行加密，通过密码机制实现其他参与方对原始数据的不可见性以及数据的无损失性。由于加密技术可解决安全通信的问题，因此多应用于分布式应用。
    - 面向数据挖掘的隐私保护
      - 1. **关联规则的数据挖掘**：关联规则挖掘是数据挖掘领域研究的重点之一，是从大量数据中挖掘数据项之间隐藏的关系，发现数据集中项集之间的关联和规则的过程。例如购物篮分析，寻找商品之间隐藏的关联规则。
      - 2. **隐私保护的关联规则挖掘**：包含两类方法，一种是修改支持敏感规则的数据，使得规则的置信度和支持度小于一定的阈值而实现规则的隐藏；另一种是不修改数据，而是隐藏生成敏感规则的频繁项集，尽可能降低敏感规则的置信度或者支持度
    - 匿名化隐私保护模型
      - 传统的匿名化方法
      - 链接攻击
      - k-anonymity（PPT第34页）
      - 同质性攻击和背景知识攻击（PPT第35-36页）
      - l-diversity（PPT第37-39页）
      - t-closeness（P41）
  - 数据匿名化方法（页码：42）
    - **泛化（Generalization）**：这是通过使用更抽象、概括的值或区间来替代精确值的方法。
    - **抑制（Suppression）**：这种方法是将数据表中的数据直接删除或隐藏。
    - **聚类（Clustering）**：这种方法根据给定的规则将数据集分成各种类簇，尽可能保证类簇内对象相似，不同类簇的对象相异。
    - **微聚集（Micro-aggregation）**：这种方法是将相似的数据划分在同一个类中，每个类至少有k条记录，用类质心代替类中所有记录的准标识符属性值。
    - **分解（Decomposition）**：这种方法根据敏感属性值对数据表分组，尽量使得同一组的敏感属性值不同，将分组后的数据表拆分为分别包含准标识符属性信息和包含敏感属性信息的两张表。
    - **置换（Permutation）**：这种方法是对数据表分组，把每组内的敏感属性值随机交换，打乱顺序，再拆分数据表，对外发布。
    - 泛化（页码：43-44）
      - 泛化树（页码：44）
        - **数值型数据泛化树**：对于数值型属性，可以使用泛化树进行泛化。底层的具体取值被一个覆盖精确数值的区间代替，而上层的取值则逐渐变得更加抽象和概括。这样，可以保持数据的真实性和一致性，同时减少敏感信息的泄露。
        - **分类型数据泛化树**：对于分类型属性，可以使用泛化树进行泛化。底层的具体取值被一个更一般的值代替，而上层的取值则逐渐变得更加抽象和概括。这样，可以保持数据的真实性和一致性，同时降低敏感信息的泄露风险。
      - 域泛化（全局泛化）与值泛化（局部泛化）（页码：45-46）
        - **域泛化**：域泛化将一个给定的属性域从底层开始同时向上泛化，直到满足隐私保护要求，然后停止泛化。在域泛化中，同一个属性的全部值必须在同一层上进行泛化。域泛化分为全域泛化和子树泛化两种方式。
          - **全域泛化**：在泛化树中，同一个父节点下的所有子节点要么全部泛化，要么全部不泛化。全域泛化可以确保一致性，但可能导致较大的信息损失。
          - **子树泛化**：在泛化树中，对部分子节点进行泛化，而其他兄弟节点不要求泛化。父节点代替泛化的子节点进行数据发布。子树泛化相比全域泛化可以减少信息损失，但可能引入一些不一致性。
        - **值泛化**：值泛化将原始属性域中的每个值直接泛化成一般域中的唯一值。在值泛化中，可以对多个属性的值同时进行泛化，只需要对不符合限制要求的等价类进行泛化，要求一个等价类中所有记录都泛化成相同的值。值泛化可以保持数据的一致性，但可能引入一些不一致性。
    - 抑制（P48）
      - **记录抑制**：对数据表中的某条记录进行抑制处理，即完全删除该记录，使其无法识别。
      - **值抑制**：对数据表中某个属性的部分值进行抑制处理，即用符号或其他模糊化方法替换一部分属性值，以减少敏感信息的泄露。
      - **单元抑制**：对数据表中某个属性的所有值进行抑制处理，即完全删除该属性，使其无法被使用或分析。
  - 差分隐私(P49)
    - **严格定义的隐私保护**：差分隐私对隐私保护进行了严格的定义，并提供了量化评估方法，使不同参数处理下的数据集所提供的隐私保护水平具有可比较性。
    - **抵抗背景知识攻击**：差分隐私假设攻击者掌握最大的知识背景，即能够获得除目标记录外所有其他记录的信息。因此，差分隐私的设计考虑了更强的攻击模型，能够更好地保护隐私。
    - **严格的隐私保护证明**：差分隐私提供了严格和科学的方法来证明其隐私保护水平。当模型参数改变时，可以对其隐私保护水平进行定量分析，以便在隐私与数据可用性之间进行权衡。
    - 差分隐私基础(P53)
    - 数值型差分隐私和非数值型差分隐私匿名化与差分隐私
    - 差分攻击
    - 差分隐私思想
    - 差分隐私定义
      - 当ε = 0时，攻击者无法区分相邻数据集，保护程度最高，但数据可用性最差。
      - 当ε增大时，保护程度逐渐降低，ε过大会造成隐私泄露。
      - 通常，ε取较小的值，如0.01、0.1或ln 2、ln 3等。ε的取值应结合具体需求设定，以平衡输出结果的安全性和可用性。
    - 差分隐私的实现(P56)
      - 拉普拉斯机制
      - 高斯机制
      - 全局敏感度和局部敏感度
        - 全局敏感度是指在任意一对邻近数据集D和D'上，查询函数f的输出结果之间的最大变化范围。它与数据集无关，由查询函数本身决定。
        - 局部敏感度是指对于给定的数据集D和它的任意邻近数据集D'，查询函数f在D上的局部敏感度。它由查询函数和给定数据集中的数据共同决定。
      - 数值型差分隐私：拉普拉斯和高斯机制
      - 非数值型差分隐私：指数机制
  - 同态加密(P69)
    - 同态加密基础
    - 同态加密的应用场景：安全的数据外包
    - 组成
      - **KeyGen算法**：通过计算安全参数生成一对公私钥。
      - **Encrypt算法**：使用公钥将明文加密为密文。
      - **Evaluate算法**：在密文上进行运算，例如加法或乘法。
      - **Decrypt算法**：使用私钥将密文解密为明文。
    - 同态加密的发展
      - 1. **仅支持加法同态的加密体制**：最早的同态加密体制只支持加法同态或乘法同态，但不能同时满足两者。
      - 2. **半同态加密**（Partially Homomorphic Encryption，PHE）：半同态加密体制同时满足加法同态和乘法同态的性质，但只能进行有限次的加和乘运算。
      - 3. **浅同态加密**（Somewhat Homomorphic Encryption，SWHE）：浅同态加密体制也同时满足加法同态和乘法同态的性质，但可以进行任意多次加和乘运算。
      - 4. **全同态加密**（Fully Homomorphic Encryption，FHE）：全同态加密体制是最理想的同态加密形式，它可以在不解密的条件下对加密数据进行任何可以在明文上进行的运算，实现了深度和无限的数据分析，对加密信息进行深入分析而不影响其保密性。
    - 半同态加密
      - ⊗
        - 乘法同态加密：当 表示乘法时，称为乘法同态加密。典型的乘法同态加密算法有：
          - RSA公钥加密算法（1977年提出）
          - ElGamal公钥加密算法（1985年提出）
        - 加法同态加密：当 表示加法时，称为加法同态加密。典型的加法同态加密算法有：
          - Paillier公钥加密算法（1999年提出），是最常用且最具实用性的加法同态加密算法。
      - ElGamal乘法同态加密
      - Paillier加法同态加密
      - 全同态加密
    - 同态加密的应用
      - **医疗机构的数据分析**：在医疗机构中，数据处理能力较弱，可以借助云服务商提供的计算服务。使用同态加密，医疗数据可以在加密的状态下存储和计算，而不泄露隐私信息。云服务商可以进行数据搜索、分析和处理等功能，同时保护数据隐私。
      - **电子投票**：同态加密可以用于设计安全的电子选举系统。统计方可以在不知道投票者投票内容的情况下对投票结果进行统计，既保证
    - 同态加密的优势与挑战
      - **降低计算代价**：同态加密可以对多个密文进行计算后再解密，降低了计算代价。
      - **降低通信代价**：同态加密可实现无密钥方对密文的计算，无需经过密钥方，降低了通信代价。
      - **保证数据安全性**：同态加密可以实现让解密方只能获知最终的结果，而无法获得每个密文的消息，从而保证了信息的安全性。
      - **计算效率**：当前的同态加密方案的计算复杂度较高，如何设计高效的全同态加密方案仍然是一个问题。
      - **安全性**：同态加密方案大多基于未论证的困难问题，寻找可论证的困难问题仍然是一个挑战。
      - **噪音消除**：同态加密需要额外的消除噪音算法，如何设计具有自然同态性的全同态加密方案仍然是一个问题。
  - 安全多方计算(P93)
    - 安全多方计算基础
    - 安全多方计算的提出
    - 安全多方计算的形式化描述
      - **输入独立性**：各方能独立输入数据，计算时不泄露本地数据。
      - **计算正确性**：计算结束后各方能够得到正确的计算结果。
      - **去中心化性**：各参与方地位平等，提供了去中心化的计算模式。
    - 安全多方计算的威胁模型
      - **诚实模型**：参与者按照协议要求行动，不提供虚假数据，不泄露、窃听数据，不终止协议，完全按照协议执行。
      - **半诚实模型**：在诚实模型基础上保留所有收集到的信息，推断其他参与者的秘密信息。
      - **恶意模型**：无视协议要求，可能提供虚假数据、泄露数据、窃听甚至终止协议。
    - 安全多方计算的计算模型
      - **基于"可信第三方"的计算模型**：参与方得到计算结果，可信第三方得到参与方的输入信息和计算结果，信息的保密性由可信第三方来保证。然而，在实际情况下很难找到完全可信的第三方，所以这种模型很少使用。
      - **交互计算模型**：参与方按照协议步骤执行计算，按协议的要求将中间结果发送给其他参与方，同时接收其他参与方计算的中间结果，信息的保密性由协议的安全性来保证。这是安全多方计算中最常用的模型，提供了一种去中心化的计算方式。
      - **外包计算模型**：各个参与方希望使用云计算提供的计算资源，但不想直接将信息委托给云计算服务提供商，也不想让其得知计算结果。参与方将信息处理后存储在外包服务器上，由外包处理器对所有参与方的秘密信息进行计算，并将结果发送给各参与方。信息的保密性由协议的安全性来保证。
    - 基本密码协议
    - 安全多方计算的应用
      - **门限签名**：将私钥拆分为多个秘密分片，只有在达到门限值的参与者共同协作时才能生成有效的签名。
      - **电子拍卖**：在不直接公开竞拍者的出价情况下，能够计算出所有参与者输入的最大值或最小值，使得在线拍卖成为现实。
      - **联合数据查询**：多个数据库可以共同进行数据查询，使用安全多方计算保护各数据库的私有信息或知识版权。
      - 其他领域：安全多方计算涉

Chap6

硬件攻击技术

- 漏洞攻击
  - **指令集漏洞**：不同系统架构对相同指令集的实现方式有所不同，指令集漏洞是由于处理器开发人员在指令集的硬件实现过程中，对一些特殊情况考虑不周导致的。各大处理器厂商会定期发布产品的勘误表，列出指令集漏洞并提供修复措施。
  - **浮点除（FDIV）的实现漏洞**：在某些处理器中，浮点除法的实现存在错误，导致计算结果不准确。例如，因特尔奔腾处理器中的FDIV错误就是一个经典的实现漏洞。
  - **性能优化缺陷**：Meltdown和Spectre源于乱序执行和分支预测（提高CPU性能）
- 硬件木马(P21)
  - 不可信的产业链（页码：22）
  - 木马分类（页码：23）
    - **插入阶段**：指木马的植入阶段，可以是在制造过程中的偶然插入（由制造过程的缺陷引起），或者是蓄意插入（由攻击者植入）。
    - **抽象级别**：指木马在电路中的位置，可以是在较高级别的模块中（如处理器、芯片组），也可以是在较低级别的模块中（如外围接口芯片）。
    - **激活机制**：指木马被触发的机制，可以是在已知的功能状态下触发，也可以是在特定状态的组合或序列下触发。
    - **影响**：指木马对系统的影响程度，可以是功能参数错误，也可以是泄露敏感信息等。
    - **位置**：指木马在电路中的具体位置，可以是在信号路径上，也可以是在控制路径上。
  - 木马与故障的区别（页码：23）
    - **电路故障**是指在电路的生产加工过程中由于工艺或流程不完美导致的电路缺陷。
    - **硬件木马**是指人为蓄意加入的恶意电路模块，其功能超出了指定功能之外。与电路故障不同，硬件木马通常较难检测到。
  - 软件木马与硬件木马对比（页码：24）
    - **软件木马**是一种带有恶意代码的软件，在代码运行过程中激活。与硬件木马攻击目标相似，但软件木马通常可以通过软件更新进行移除。
    - **硬件木马**存在于集成硬件中，在硬件运行过程中被激活。一旦植入，很难移除，通常需要流片（重新设计芯片）才能解决。
  - 硬件故障（页码：25）
- 故障注入攻击
  - 故障注入分类
  - 光故障注入
  - 电磁故障注入
  - 时序电路与故障注入
  - 影响因素

侧信道攻击与旁路侧信道（页码：39）

- 物理侧信道攻击（页码：41）
  - **时间侧信道**：通过系统在不同行为流程下所表现出的执行时间差异，推断有用的信息，以达成或辅助攻击。
  - **功耗侧信道**：通过分析设备随时间的功耗曲线，推测CPU执行过程中的某些信息。
- 时间侧信道攻击（页码：42-43）
- 功耗侧信道攻击（页码：44-46）
  - **简单功耗分析（Simple Power Analysis, SPA）**：通过直接分析设备随时间的功耗曲线，推断CPU执行的顺序或指令。
  - **差分功耗分析（Differential Power Analysis, DPA）**：通过比较不同输入
  - **相关功耗分析（Correlation Power Analysis, CPA）**：通过建立功耗曲线与已知输入的相关性模型，推断出密钥或敏感数据。
- 电磁侧信道攻击（页码：49）
- 微架构侧信道攻击（页码：50）
- Cache侧信道攻击（页码：51-55）
  - **Prime-Probe**：通过填充特定的缓存组，并测量读取时间来推测缓存中的信息。
  - **Flush-Reload**：通过驱逐缓存中的数据，并测量重新加载的时间来推测缓存中的信息。
  - **Evict-Reload**：与Flush-Reload类似，通过驱逐共享内存中的数据块并重新加载来推测缓存中的信息。
  - **Flush-Flush**：通过执行flush指令清空缓存中的原始数据，并测量刷新时间来判断原始数据是否被缓存。

硬件防护技术

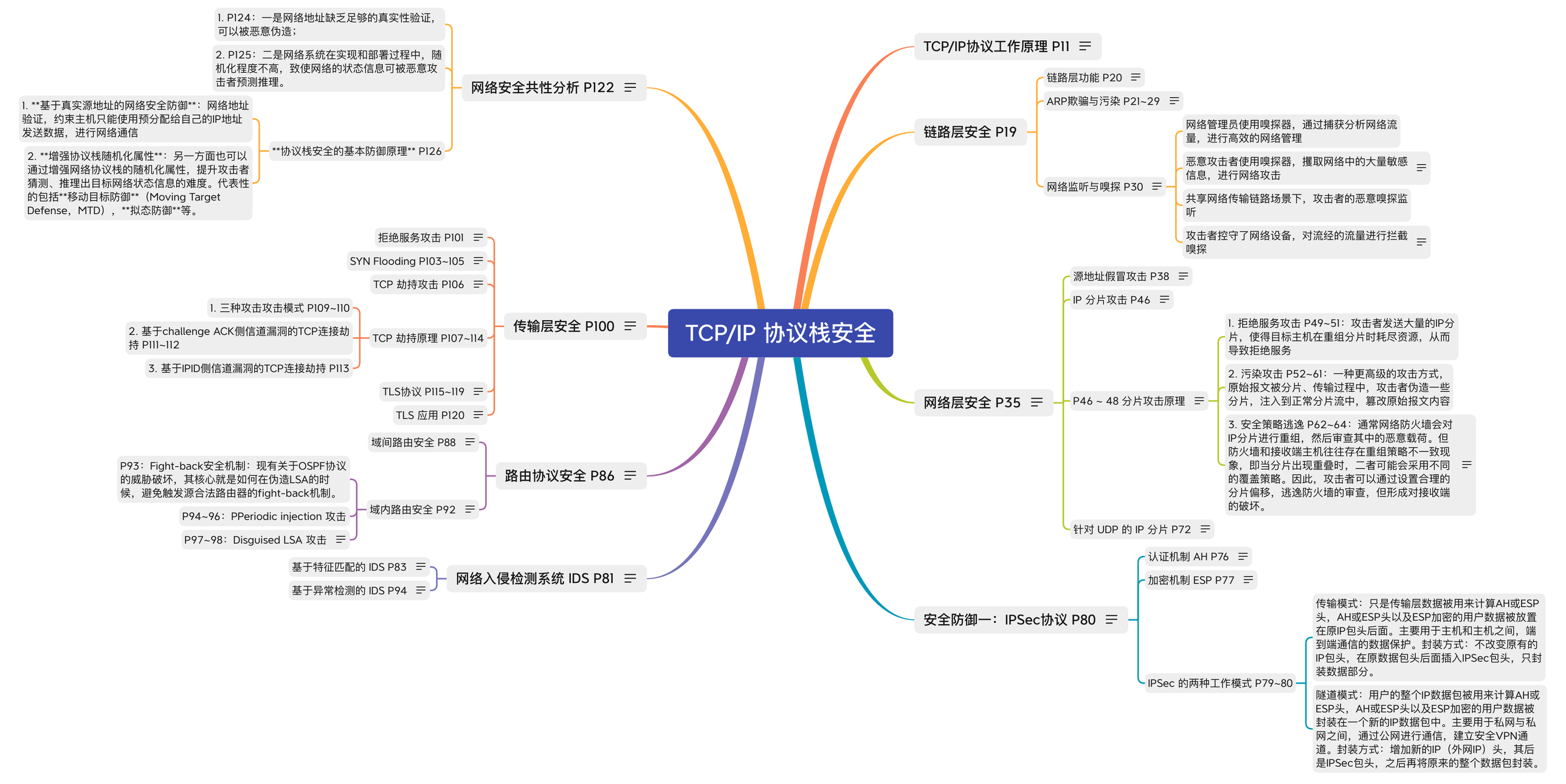
- 木马检测技术
  - 检测挑战
    - 选择合适的木马模型。
    - 生成测试向量来激活木马或增加侧信道测量中的木马敏感度。
    - 消除/校准测试中的环境噪声和测量噪声。
  - 集成电路生命周期（页码：62）
- 方法分类（页码：63）
  - 破坏性检测
    - 化学去封
    - 电路扫描
    - 逆向分析
    - 芯片重构
    - 模板匹配
  - 非破坏性检测
    - 实时监测（在线监测）
    - 芯片测试
    - 测试向量
    - 特征检测
    - 结果分析
- 逻辑测试和侧信道分析（页码：64）
  - 逻辑测试
  - 侧信道分析
- 隔离技术
  - 存储器隔离（页码：66）
  - EPT硬件虚拟化技术（页码：67-68）
  - ARM TrustZone（页码：69-70）
  - Intel SGX（页码：71-72）
- 密码技术（页码：75）
- 旁路信息隐藏与掩码技术（页码：79-81）

典型漏洞分析

- MOLES（页码：83）
- Meltdown漏洞（页码：83）
  - 乱序执行（页码：84）
  - 攻击过程（页码：85）
    1. 将目标内核地址加载到寄存器rcx和rbx。
    2. 对寄存器rax进行清零。
    3. 从目标内核地址中读取一个字节并加载到寄存器al中（步骤1）。
    4. 将寄存器rax左移12位（相当于乘以4096）。
    5. 从rbx和rax的和地址中加载一个qword（8字节）。
  - 攻击场景
- Spectre漏洞（页码：87-89）
- VoltJockey漏洞（页码：89-96）
- COVID-19家庭检测套件的漏洞（页码：96）

CH7操作系统安全

- 为什么存在安全问题 P5-8
  - 1. 现代操作系统是规模庞大的软件系统，现代操作系统是**系统之系统**，各个模块之间的依赖关系复杂
  - 2. 其二，现代操作系统的设计以性能为最主要目标，而非安全性
- 攻击前提 P9
  - 攻击者可以构造任意输入并接受受害进程输入校验；
  - 攻击者无法**直接**读写系统下进程的内存，无法**直接**干预处理器上指令的执行
- 攻击目标 P10
- 操作系统基础攻击方案
  - Linux 内核为每一个进程维护一个**独立的**线性逻辑地址空间，以便于实现进程间内存的相互隔离
  - 这一线性逻辑地址空间被分为**用户空间**和**内核空间**；用户态下仅可访问用户空间，系统调用提供接口以访问内核空间；内核态下亦无法访问用户空间
  - 1. 文本段：进程的可执行二进制源代码
  - 2. 数据段：初始化了的静态变量和全局变量
  - 3. BSS 段 ： 未初始化的静态变量和全局变量
  - 4. 堆区 ： 由程序申请释放
  - 5. 内存映射段：映射共享内存和动态链接库
  - 6. 栈区 ： 包含了函数调用信息和局部变量
  - 栈区内存的作用 P19
  - 密切相关的寄存器 P20
  - 栈区溢出攻击 P30
    - 1. 返回至溢出数据 P32
    - 2. 返回至库函数 P35
  - 总结 P37
  - 基础堆区攻击 P38
  - 堆溢出攻击 P44
    - 1. P45 直接覆盖malloc_chunk首部为无意义内容，在堆管理器处理管理元数据时将造成崩溃
    - 2. P46~49 构造堆块重叠：堆块重叠是一种病态堆区内存分配状态，同一堆区逻辑地址被堆管理器多次分配。如右图所示，造成Heap Overlap之后攻击者可以通过写入一个堆块，实现对另一堆块内容的写入；同理，读出被覆盖堆块当中的数据。
    - 3. P50 更加复杂的堆区溢出攻击：利用堆管理其他机制。例如，基于unlink机制的堆区溢出攻击
    - 4. Use-After-Free 51是进程由于实现上的错误，使用已被释放的堆区内存。被free函数释放的堆块内存仍然可以被继续使用，当再次调用malloc分配内存时，会同时有两个指针指向同一堆块造成 堆块重叠。
    - 5. Double-Free 52 进程多次释放统一堆块，被多次释放的堆块将被堆管理器分配多次，最终产生堆块重叠。
    - 6. Heap Over-Read 53 直接越界读出堆区数据，造成信息泄露
    - 7. Heap Spray 堆喷：堆喷申请大量的堆区空间，并将其中填入大量的滑板指令（NOP）和攻击恶意代码；堆喷使用户空间存在大量恶意代码，若EIP指向堆区时将命中滑板指令区，受害进程最终将"滑到"恶意代码。堆喷对抗地址的随机浮动类型的防御方案 ，并实现了恶意代码的注入。
- 操作系统基础防御方案 P55
  - 防御技术的缺陷
    - 1. 对于Stack Canary，作为Canary的内容可能被泄露给攻击者，或被暴力枚举破解
    - 2. 对于ASLR已有去随机化方案，泄露内存分布信息
    - 3. ROP等进程控制流劫持方案亦可以绕过ASLR、NX的保护机制
- 高级控制流劫持方案 P64
  - P93 两种具体实现
    - 1. 直接改写vtable指向的函数指针，可通过构造堆块覆盖完成
    - 2. 覆盖vtable字段，使其指向攻击者控制的内存，然后在其中布置函数指针
  - P90 虚假vtable劫持Fake vtable Hijacking
  - P88 ~ 89具体步骤与 GOT Hijacking 的总结
  - P82 全局偏置表劫持GOT Hijacking
    - 1. ASLR将随机浮动共享库的基地址，导致共享库的位置无法被硬编码
    - 2. 并非动态链接库当中的所有库函数都需要被映射（降低内存开销）
  - ROP 总结
  - P71 面向返回地址编程 ROP Gadget
  - 共享库机制 P67
  - 进程执行的更多细节 P65
- 高级操作系统基础防御方案 P94
  - P95~99 控制流完整性保护
    - 1. 通过二进制或者源代码程序分析得到控制流图（CFG），获取转移指令目标地址的列表
    - 2. 运行时检验转移指令的目标地址是否与列表中的地址相对应；控制流劫持会违背原有的控制流图，CFI 则可以检验并阻止这种行为
  - 指针完整性保护 P101
  - 信息流控制 P103
  - I/O子系统保护 P107


TCP/IP 协议栈安全
TCP/IP协议工作原理 P11
链路层安全 P19
链路层功能 P20
ARP欺骗与污染 P21~29
网络监听与嗅探 P30
网络管理员使用嗅探器，通过捕获分析网络流量，进行高效的网络管理
恶意攻击者使用嗅探器，攫取网络中的大量敏感信息，进行网络攻击
共享网络传输链路场景下，攻击者的恶意嗅探监听
攻击者控守了网络设备，对流经的流量进行拦截嗅探
网络层安全 P35
源地址假冒攻击 P38
IP 分片攻击 P46
P46 ~ 48 分片攻击原理
1. 拒绝服务攻击 P49~51：攻击者发送大量的IP分片，使得目标主机在重组分片时耗尽资源，从而导致拒绝服务
2. 污染攻击 P52~61：一种更高级的攻击方式，原始报文被分片、传输过程中，攻击者伪造一些分片，注入到正常分片流中，篡改原始报文内容
3. 安全策略逃逸 P62~64：通常网络防火墙会对IP分片进行重组，然后审查其中的恶意载荷。但防火墙和接收端主机往往存在重组策略不一致现象，即当分片出现重叠时，二者可能会采用不同的覆盖策略。因此，攻击者可以通过设置合理的分片偏移，逃逸防火墙的审查，但形成对接收端的破坏。
针对 UDP 的 IP 分片 P72
安全防御一：IPSec协议 P80
认证机制 AH P76
加密机制 ESP P77
IPSec 的两种工作模式 P79~80
传输模式：只是传输层数据被用来计算AH或ESP头，AH或ESP头以及ESP加密的用户数据被放置在原IP包头后面。主要用于主机和主机之间，端到端通信的数据保护。封装方式：不改变原有的IP包头，在原数据包头后面插入IPSec包头，只封装数据部分。
隧道模式：用户的整个IP数据包被用来计算AH或ESP头，AH或ESP头以及ESP加密的用户数据被封装在一个新的IP数据包中。主要用于私网与私网之间，通过公网进行通信，建立安全VPN通道。封装方式：增加新的IP（外网IP）头，其后是IPSec包头，之后再将原来的整个数据包封装。
网络安全共性分析 P122
1. P124：一是网络地址缺乏足够的真实性验证，可以被恶意伪造；
2. P125：二是网络系统在实现和部署过程中，随机化程度不高，致使网络的状态信息可被恶意攻击者预测推理。
协议栈安全的基本防御原理 P126
1. **基于真实源地址的网络安全防御**：网络地址验证，约束主机只能使用预分配给自己的IP地址发送数据，进行网络通信
2. **增强协议栈随机化属性**：另一方面也可以通过增强网络协议栈的随机化属性，提升攻击者猜测、推理出目标网络状态信息的难度。代表性的包括**移动目标防御**（Moving Target Defense，MTD），**拟态防御**等。
传输层安全 P100
拒绝服务攻击 P101
SYN Flooding P103~105
TCP 劫持攻击 P106
TCP 劫持原理 P107~114
1. 三种攻击攻击模式 P109~110
2. 基于challenge ACK侧信道漏洞的TCP连接劫持 P111~112
3. 基于IPID侧信道漏洞的TCP连接劫持 P113
TLS协议 P115~119
TLS 应用 P120
路由协议安全 P86
域间路由安全 P88
域内路由安全 P92
P93：Fight-back安全机制：现有关于OSPF协议的威胁破坏，其核心就是如何在伪造LSA的时候，避免触发源合法路由器的fight-back机制。
P94~96：PPeriodic injection 攻击
P97~98：Disguised LSA 攻击
网络入侵检测系统 IDS P81
基于特征匹配的 IDS P83
基于异常检测的 IDS P94

DNS安全

- DNS 概述：P9
  - DNS 演进 P10
  - DNS域名结构：P12
  - DNS区域组织形式：P13
  - 权威域名服务器 P14
- DNS使用及解析过程 P17
  - 1. P19~21 DNS使用 & 请求过程：并不是每一次域名解析都完成整个查询流程：按照域名空间层次化结构自上而下应答，本地DNS - Root - 顶级域 - 二级域名
  - 2. DNS迭代查询过程**交互信息格式**：P22~26
  - 3. DNS 反向查询 P27
- DNS攻击 P29
  - DNS 各个层面的攻击 P30
  - 操控用户机 P31
    - 1. 修改/etc/resolv.conf：将主机DNS服务器修改成攻击者控制的服务器，攻击者回复恶意内容
    - 2. 修改/etc/hosts：直接修改地址与域名映射关系，修改为攻击者控制的IP地址
    - 3. 嗅探回复：监听主机发出的DNS查询请求，注入恶意回复
  - 本地缓存中毒攻击 P32~35
    - 1. 只针对回复部分伪造，影响面：一个主机名；在伪造回复中，把主机名映射到攻击地址，告诉本地DNS服务器域名对应 地址是攻击者的计算机。
    - 2. 针对授权部分攻击，影响面：整个域；将ns.attack.com放在授权部分，查询目标域内任何一个主机名时，本地 DNS服务器把请求发给ns.attack.com。
  - 远程缓存中毒攻击 P36
  - **如何伪造一个被受害者接收的应答包** P38
    - 1. 缓存时限约束：域名不能在缓存中
    - 2. 应答包匹配约束：端口号、交易ID匹配
    - 3. 时间约束：伪造包比正常DNS应答快
  - P47~49 来自恶意DNS服务器的污染
  - 拒绝服务攻击 DDoS：P50
- DNS攻击预防策略
  - 通过非对称加密验证身份 - DNSSEC：P55~60
  - DMSSEC 的不足 P60~62
    - 1. 经济因素制约其部署，大量签名和验证带来严重的负载，影响效率，验证能力取决于递归服务器以及客户端是否对数字签名记录做校验。
    - 2. 离线存储与频繁使用密钥，私钥的安全无法保证；没有提供机密性和一致性检验，无法抵御重放攻击；只提供单向认证，没有对客户的认证，不能解决缓存中毒； 没有提供DoS防护。
    - 3. 引入了新的实现和配置错误。
    - 4. 用 DoT / DoH 改进 P61~62
  - 基于系统管理：P63~64
    - 1. 规范并梳理DNS配置过程中出现的漏洞
    - 2. 限制用户在短时间内发起大量DNS查询
    - 3. 增加端口猜测难度，如用于查询的UDP端口不再是默认的53，而在UDP端口范围内随机选择(排除预留端口)
    - 4. 分布式部署、恶意流量过滤等方案
    - 5. 严格检查：附加区记录和问题区问题不在同一域管辖，则不会采信，防范恶意权威DNS虚假记录污染缓存
  - 新型架构设计 Blockstack：P65~66
- 典型案例分析
  - Kaminsky攻击：P68~72
    - 缺乏端节点身份和发布内容验证
    - 数据未采用加密传输
    - 协议自身检查采信机制不足
    - 针对协议内容和通信实体设计可靠的验证方式
    - 通过加密等策略确保对用户身份和发布内容进行鉴别，实现有效防御
  - 恶意服务器回复伪造攻击：P73~76
    - 域名系统存在复杂的解析依赖
    - DNS管理方面缺乏验证配置内容能力，存在错误输入的记录
    - 规范域名应用管理，减少使用环节引入的安全威胁
    - 对配置内容建立审核机制，确保每一条域名记录的正确性，从根上防止接受恶意信息
  - 拒绝服务攻击：P77
    - 大量恶意请求到达被攻击服务器并被攻击服务器接受
    - 安全的网络体系架构，如采用真实网络地址保障每一台接入网络计算机 的安全性，建立快速DDoS溯源机制
    - 专用DNS请求过滤系统，针对DNS数据包进行恶意流量进行过滤；分 布式部署降低攻击对性能的影响

真实源地址验证

- 设计背景 P8
  - 基于 IP 地址伪造的攻击 P11
  - 危害严重 P13
  - 现有防御方法 P15
  - 现有方法的缺陷 P16
    - 算法复杂，协议开销大
    - 缺乏部署激励
    - 完备性不足
    - 可扩展性不足
  - 真实 IP 原地址的三重含义 P19
    - 1. 经授权的：IP源地址必须是经互联网IP地址管理机构分配授权的，不能伪造
    - 2. 唯一的：IP源地址必须是全局唯一的
    - 3. 可追溯的：网络中转发的IP分组，可根据其IP源地址找到其所有者和位置
  - 真实源地址验证体系结构 SAVA P20
    - 1. 接入子网层：该层位于 SAVA 体系结构的最底层，负责对直接接入网络的主机进行源地址验证。在这一层，SAVA 设备会绑定主机和其IP地址，并验证从主机发出的数据包是否使用了正确的源地址。这一层的作用是防止本地网络中的源地址伪造攻击，确保只有合法的主机才能发送具有正确源地址的数据包。
    - 2. 自治系统内部层：该层位于 SAVA 体系结构的中间层，负责在自治系统内部进行源地址验证。在自治系统内部的路由器上进行源地址验证，以确认数据包的源地址确实来自预期的接入点。这一层的作用是防止自治系统内部的源地址伪造，确保数据包的源地址符合预期的接入点。
    - 3. 自治系统间层：该层位于 SAVA 体系结构的最顶层，负责在不同自治系统之间进行源地址验证。在自治系统间的路由器上进行源地址验证，以确认数据包的源地址确实来自预期的自治系统。这一层的作用是防止不同自治系统之间的源地址伪造，确保数据包的源地址符合预期的自治系统。
    - 1. 安全责任明确：易于追踪攻击事件，定位攻击者；杜绝基于伪造源地址的攻击
    - 2. 网络管理简化：简化身份认证；流量可审计不可抵赖；支持基于IP地址的网络计费
- 真实源地址验证体系结构设计原则
  - 1. IPv4地址结构：P23~25
  - 2. IPv6地址结构：P26~27
  - 3. 自治域 P28
  - 4. 域内路由 IGP-OSPF：P29~33
  - 5. 域间路由 BGP：P34~37
- SAVA设计原则：P38
  - 可扩展性 P38~45
    - 层与层之间相独立，每层可以独立进行技术的演化
    - 每层拥有自己的优化目标
    - 不同层次利用局部信息对决策进行分布式计算，以实现个体最优
    - 综合起来，这些局部算法实现了全局最优
  - 可演进性 P46
    - 1. 与已有协议兼容：SAVA体系结构建立在当前互联网体系结构基础上，整体的技术依附于现有体系结构实现， 因此必须要求技术对应协议与现有体系结构协议兼容
    - 2. 自身部署可演进：SAVA的部署是一个持续性的过程，所以会出现部分区域已经部署SAVA，而部分区域尚 未部署SAVA的情况，需要考虑在发展部署的过渡阶段SAVA自身的兼容性
    - 3. 运营商之间可演进：考虑到网络中不同运营商的存在，SAVA体系结构还应允许运营商可以采用各自不同的实 现，SAVA系统各部分相互独立，且功能彼此不依赖
  - 安全性 P47
    - 1. 可信标识风险：标识应当具备唯一性、可追溯性
    - 2. 数据转发风险：需保证携带标签的数据包转发中不被篡改，即使篡改也应被及时并准确地识别
    - 3. 单点信任风险：标签的验证应当不依赖于中心化的网络基础设施，以免引入网络基础设施的信任风险
- SAVA体系结构与关键技术 P50
  - 部署与地址管理范围的失配问题 P50
  - 面向地址域的新型源地址验证SAVA体系结构 P51~52
    - 以一个校园网为例，地址域可以是某一个院系下的某个所某个组，也可以是某个所、某个院系，甚至可以是整个校园网
    - "地址域" 显著提升体系结构的灵活性，实现了部署结构灵活的源地址真实性验证体系结构
    - 接入网、地址域内和地址域间三层结构，具有松耦合、多重防御、支持增量部署等优点
    - 1. 接入层面提供主机粒度的源地址验证能力，以保证地址使用的可追溯性
    - 2. 在地址域内层面提供前缀级别的保护能力，以保护核心设备不被攻击
    - 3. 在地址域间层面提供地址域级别的联盟内可验证能力以及保护自身不被依造的能力
  - SAVI实现源地址验证的“三步曲" P54
    - 1. 获取合法地址：监听控制类报文（如ND、DHCPv6），即CPS分组，获取地址分配信息以识别主机合法IP源地址
    - 2. 建立绑定关系：将合法的IP地址与主机网络附属的链路层属性绑定（"绑定锚"）（可验证的，且比主机的IP源地址本身更难欺骗）
    - 3. 匹配验证：对数据包中的IP源地址与其绑定锚进行匹配（**部署位置越靠近主机，有效性越高**）
  - 接入网异构多样性 P55
    - 1. 终端多样性：终端包括手机、电脑、服务器、嵌入式设备等各种类型设备，即使同一种设备，其上运行的系统也可能不同，比如手机可能是安卓系统，也可能是IOS系统
    - 2. IP分配方式多样性：包括DHCP协议分配、SLAAC协议分配、静态配置等
    - 3. 接入方式多样性：包括有线、无线等，不同接入模式可用的邦定锚可以不同
  - SAVI 验证方式 P56
    - 所有相关网络设备在同一个网络管理机构管理控制下
    - 解决方案与接入子网地址管理分配和控制策略密切相关
    - 解决方案与端系统的接入方式密切相关
  - DHCP P57~59
    - 1. 允许主机在加入网络时从网络服务器动态获取IP地址
    - 2. 在使用中的地址可以更新租期
    - 3. 允许地址重用(仅在连接on时保留地址)
    - 4. 支持想要加入网络的移动用户
  - 无线局域网 WLAN P60~61
  - 无线 SAVI P62
  - 802.1x 认证 P63
  - 动态自适应的地址分配分组监听 P64~65
  - SAVI 技术方案概要 P66
    - 1. 监听地址分配协议的控制报文，确定合法的IP源地址
    - 2. 将合法的IP地址绑定到主机的链路层属性 (绑定锚)
    - 3. 对数据包中的IP源地址与它们所绑定的绑定针是否匹配进行检验
    - 1. 附着在同一接入设备下的主机不能伪造本接入设备下 其余主机的IP地址
    - 2. 附着在某一接入设备下的主机不能伪造其他接入设备 下主机的IP地址
  - 域内真实源地址验证 P67
    - 1. 如果一个地址域与接入网相连，需保证从接入网流入地址域的流量，其源地址不会假冒该接入网之外的地址
    - 2. 如果接入网部署了SAVI，进行二次验证；如果接入网没有部署SAVI，缩小源地址假冒的范围（接入网级别）
    - 3. 保证从地址域内产生并流出地址域的流量，其源地址不会假冒地址域之外的地址
  - 原地址验证表 SAVA-P P68
  - P70 **源地址验证表生成方法**
    - 1. 正确性：解决路由不对称问题
    - 2. **低开销**：通信开销和计算开销
    - 3. **激励性**：网络通过主动部署受益
  - 基本思路和基本框架 P71~73
  - DPP 报文的生成与处理 P74~85
  - 方案总结 P85
  - 域间真实源地址验证 SAVA-X P86
  - P93 层次结构
  - 基于 IPV6 的可信身份识别 P97~98
    - 网络的分布式管理与终端标识的跨域有效性间的矛盾
    - 要求真实性标识基于统一结构
    - IPv6地址空间承载终端标识保障跨域验证的有效性
    - 1. 在地址真实性的基础上，设计嵌入可信设备标识符的IPv6 地址生成算法，兼顾多种IPv6地址分 配机制和组网环境, 将端设备信息携带于数据包中，实现端网协同
    - 2. 地址标识采用动态更新机制，达到对终端设备或终端用户的身份动态标识和隐私保护的目的
    - 3. 构建可信设备标识符的认证、管理、追溯和审计机制
  - P99 数据包防篡改机制 P99~100

ch11 公钥基础设施

- PKI 体系结构的组成 P9
  - 保证公钥真实性 P19
  - PKI 作用 P20~22
  - CA P23~25
  - RA P28
  - 证书库 P29
  - 安全服务器 P30
  - 密钥备份和恢复系统 P31
  - 证书撤销列表 P32
  - PKI 应用接口系统 P33
  - CA 的信任关系 P34
  - PKI 信任模式 P35~36
  - CA 的层次结构 P37
- CA的信任模型 P38
  - 1. 单CA信任模型：基本信任模型，也是目前许多组织或单位在Intranet中普遍使用的一种模型
  - 2. 层次信任模型：层次信任模型也称为分级信任模型，它是一个以主、从CA关系建立的分级PKI结构
  - 3. 分布式信任模型：分布式信任模型也称为网状信任模型，在这种模型中CA间存在着交叉认证
  - 4. 桥CA的信任模型：桥CA信任模型也称为中心辐射式信任模型，被设计成用来克服层次信任模型和分布式信任模型的缺点，并连接不同的PKI系统。
  - 5. WEB信任模型：Web信任模型构建在Web浏览器的基础上，浏览器厂商在浏览器中内置了多个根CA，每个根CA相互间是平行的，浏览器用户同时信任多个根CA并把这些根CA作为自己的信任锚。Web信任模型通过与相关域进行互连而不是扩大现有的主体群来使用户实体成为在浏览器中所给出的所有域的依托方。
  - 严格层次结构模型 P40~42
  - 层次模型的优缺点 P45
  - CA
  - 分布式信任模式 P46
  - 交叉验证 P47~48
  - 桥 CA 信任模型 P49
  - Web CA 信任模式 P50 & P54
  - 以用户为中心的信任模式 PGP P56
- PKI 安全问题 P59
  - 1. 数字证书颁发过程中的安全问题：证书的合法性难以保证，误发证书，恶意颁发证书，钓鱼网站证书
  - 2. 数字证书维护过程中的安全问题：证书的有效性难以验证，证书过期，证书撤销
  - PKI 证书撤销 P63
    - 私钥泄露、密钥更换、用户变化时，需要 撤销证书
    - CA 维 护 CRL (Certificate Revocation List) ，未到期但已撤销证书列表
    - 检查CRL的URL内嵌在用户证书，浏览器 安全访问URL查询证书状态, 确定是否被 CA 撤销
    - 在线证书状态协议 OCSP (online certificate status protocol)是CRL之外维护PKI证书安全的另一个协议
  - 中心化监督带来的问题 P64
- PKI安全问题的解决思路 P66
  - **建立监督机制** P67~68
  - 建立证书状态日志 P69
  - 证书撤销列表 P70
  - 在线证书状态 P71
  - 证书区块链 P72
- PKI 主要应用 P76
  - 数字证书使用

ch12 分布式系统

- 概述
  - 分布式系统的组成 P6
    - 交互网络 — 实现协作的前提，系统组件基于交互网络实现消息交互
    - 分布式算法 — 协调分布式组建之间的交互，确保系统稳定运行
    - 信任机制 — 解决不同系统组件之间的信任问题，实现可信协作
  - 三个属性 P7~8
    - 一致性：分布式系统中各正确节点的所保存的同一变量状态信息都是一致的
    - 可用性：分布式系统在收到客户端的请 求后，必须给出回应，不能让客户端陷入无限等待过程中
    - 分区容错性：分布式系统容忍其中节点 出现分区，当分区出现时，一个区域中节点发往另一个区域中节点的数据包全 部丢失，即区域间无法进行通信
  - 安全问题 P9
    - 攻击者可以攻击交互网络，监听交互过程中的隐私信息或造成网络分区
    - 协作过程中可能会出现冲突，故障等问题，从而影响一致性和可用性
    - 系统内部可能存在恶意组件，系统和客户端交互时也存在信任问题
- 交互网络 P12
  - 交互网络的组成 P12
  - **交互网络存在的安全隐患** P13~16
  - TCP/TLS 交互信道 P17~18
    - 1. 数据完整性：验证数据是否伪造而来或未遭篡改过
    - 2. 身份认证：获取通信方的身份，为访问控制机制作支撑
    - 3. 数据加密：协商会话密钥用于数据加密，防止第三方窃听传输数据
  - 防止重放攻击 P19
    - 1. TLS 防止重放攻击 P19：TLS交互信道带来的优势——基于TCP协议确保传输层面数据包不会出现重复；通过数据加密确保恶意节点无法通过监听获取应用层消息。以下消息依然可能遭受重放攻击：客户端发起的TLS请求建立信息ClientHello；TLS1.3为了加速握手过程而引入的0-RTT信息。
    - 2. **基于时间戳防御重放攻击** P20：优点：容易实现；缺点：要求交互双方进行精确的时钟同步；存在一个可以被攻击的时间窗口。
    - 3. **基于随机数防御重放攻击** P21：优点：无需时钟同步；不存在可被攻击的时间窗口；缺点：复用之前的数时，可能遭受重放攻击；需要记录大量已用过的随机数。
    - 4. 时间戳和随机数相结合（书 P409），只需要在 δ 时间内随机数不重复即可 书 page409
  - **链路故障** P23~26
  - 弹性覆盖网络RON P27
  - P2P vs C/S p33~36
  - 日蚀攻击 P37~40
- 分布式算法 P42
  - 协作过程中的安全隐患 P42
  - 时钟同步算法 P43~46
  - 并发控制 P48~71
    - 满足 一致性准则 和 有效性准则。写 W 个节点自后，认为写操作成功，在读取数据的时候，要读 R 个节点，由于 W+R > N，所以一定能读到正确的数据
    - 无leader，每个节点提出自己的提案，模拟一群追求快速达成一致的立法者。缺陷在于，追求所有节点相对平等的关系，造成不必要的信息交互。甚至可能会形成**活锁**
    - 增加了领导者的角色，只有领导者能够提案。算法分为 **领导者选举** 和 **日志复制** 两个部分。只有领导者故障的时候进行领导者选举。当用户来了一个请求，给领导者，领导者试图和所有跟随者同步日志，如果同步成功，则提交事物，并给客户端返回，否则进行回滚。
- 信任问题 P73
  - 访问控制 P75
  - 基于角色的访问控制 P76
  - 基于属性的访问控制 P77
  - 基于信任度的访问控制 P78
  - **访问控制的局限性** P79
  - 信任评价模型 P80~86
- 拜占庭将军问题 P87
  - 一致性和正确性条件 P95
  - 简化问题的解决方案 P96
  - 口头协议 P97
  - 签名消息协议 P111
- 实用拜占庭容错 P115
  - 同步异步 BFT 共识的 Quorum 设计 P117~119

ch13应用安全

应用安全攻击的共性特征及防御 P60

- 资源有限 DDoS P61
- 资源共享 非法访问 P62
- 系统漏洞 P63
- 防御原理 P64
  - 1. P65 身份认证和访问控制：身份认证是保证信息安全的第一道门户；用户在被确认身份之后在信息系统中根据身份所有的权限享受相应的信息服务。
  - 2. P66 隐私保护：隐私数据的泄露会引起严重的危险后果，通过身份认证和信任管理可以一定程度保护隐私，但是无法从数据本身保护；利用一些隐私保护算法或者技术来对隐私数据进行保护。
  - 3. P67 监控防御，用第三方软件：一定网络应用系统的漏洞是无法避免的；在网络发生被攻击、破坏的情况下，可以通过监控检测，快速识别、恢复网络应用的服务，减少损失；在网络系统各个点上部署安全防御措施，避免出现安全的木桶效应。
- 典例
  - 1. P69 微博被 XSS 攻击
  - 2. P75 微信广告
  - 3. P76 剑桥分析操纵大选

移动应用安全 P57

- P58 防御手段
  - 源代码层面：应用代码调用的组合分析出潜在的恶意行为，从而进行识别
  - 应用分发渠道层面：被篡改、盗版、二次打包、注入、反编译等破坏，需要对应用进行加固保护，构建正版指纹信息库
  - 智能终端安全监测层面：通过样本特征、行为或者缺陷等分析技术，在终端处进行安全监控，检测异常行为，进行安全控制

物联网安全 P53

- 定义 P53
- 结构与挑战 P54 ~ 56
  - 综合应用层：服务各种需求的物联网应用
  - 管理服务层：将大规模数据存储、处理、 分析
  - 网络构建层：使得感知设备接入互联网中
  - 感知识别层：物理世界与信息世界的纽带， 获取现实世界的物理数据

Web 安全 P10

- 定义 P10
- XSS 攻击 P11~12
- 防御方法 P13
- 跨站请求伪造 CSRF P14
- SQL 注入 P16~17
- SQL 注入防御方法 P18
- 点击劫持 P19
- URL 跳转漏洞 P21
- OS 命令注入攻击 P22

CDN 安全 P24

- CDN 定义 P24
- P25 工作流程
- P26 CDN 优势
- P27 ~ 33 RangeAmp 攻击
  - 1. P28~29 HTTP Range 请求
  - 2. P30 CDN 处理 Range 请求
  - 3. P31小字节范围攻击
  - 4. P32 重叠字节范围攻击
  - 5. P33 解决方案：服务器侧，CDN侧，协议侧

社交网络安全 P34

- P35 数据挖掘
- P36 垃圾信息传递
- P37 网络钓鱼
- P38 女巫攻击 Sybil attack
- P40 女巫攻击账号检测

云计算安全 P42

- P42 定义
- P43 服务类型
- P44 虚拟化与虚拟机
- P45 Hypervisor 虚拟机监视器
- P46 OpenStack 云管理平台
- P47 Docker
- P48 K8S Kubernetes
- P49 虚拟机逃逸
- P50 提权攻击
- P51 侧信道攻击
- P52 镜像和快照攻击

ch14人工智能安全

算法局限性 P112

- 数据难以获取 P112
- 数据不完整或偏斜 P114
- 成本局限性 P116
- 算法偏见 P119
- 伦理局限 P122

算法安全 P61

鲁棒性安全 P61

- 人工智能算法的优化原理 P62 ~ 63
- 人工智能算法的可解释性 P64 ~ 69
- 1. 建模前可解释性方法：数据可视化，寻找ProtoTypes 和 Criticisms（典例与特例）——Prototype 是指能够表示大部分数据的数据实例，Criticism 是指不能由Prototype很好表示出的数据实例
- 2. 建立本身具备可解释性的模型：一些具有良好可解释性的模型包括决策树模型、线性模型以及贝叶斯实例模型等
- 3. 使用可解释性方法对模型进行解释：敏感性分析和隐层分析，基于可视化方法的模型解释（结构可视化、训练可视化）
- 人工智能算法的鲁棒性评估 P70

分类维度 P72 ~ 78

- 白盒攻击：攻击者能够获知机器学习所使用的算法，以及算法所使用的参数。攻击者在产生对抗性攻击数据的过程中能够与机器学习的系统有所交互。也就说，攻击者可以将样本送入模型中以获取梯度等信息，然后依据这些信息对输入进行调整
- 黑盒攻击：攻击者对攻击的模型的内部结构，训练参数，防御方法（如果加入了防御手段的话）等等一无所知，只能通过输出输出与模型进行交互。攻击者会使用one-pixel-attack暴力攻击或使用迁移样本进行攻击。这里的迁移样本指的是，对抗样本往往是可迁移特性，即针对机器学习模型A构造的对抗性图像，也会有很大的比例能欺骗机器学习模型B。
- 目标攻击：生成的对抗样本被DNNs误分类为某个指定类别。目标攻击一般发生在多分类问题中。
- 无目标攻击：生成的对抗样本识别结果和原标注无关，即只要攻击成功就好，对抗样本最终属于哪一类不做限制。因为无目标攻击有更多的选择和更大的输出范围，所以比目标攻击更易实现。
- 基于梯度攻击：考虑模型对样本的梯度，根据梯度的方向和大小等对样本进行调整，使损失函数增大
- 基于优化攻击：将输入样本视为可变量，并通过优化算法来最小化或最大化某个特定的目标函数，以找到最优的输入样本，使得模型在该样本上产生误导性的输出结果

- 安全角度审视机器学习系统 P79
- 投毒攻击 P82 ~ 83
- P84 与对抗攻击区别
- P88 ~ 89 投毒攻击的防御
- 对抗样本攻击 P103

人工智能安全应用 P23

- 网络信息安全应用：主要包括网络安全防护应用、信息内容安全审查应用、数据安全管理应用等
- 社会公共安全应用：主要包括智能安防应用、金融风控应用等

框架安全 P32

- Pytorch 相比 TensorFlow 的优势
- **Keras** P36 ~ 37
- Caffe P38~39
- 框架安全漏洞
  - 1. TensorFlow P41 ~ 43：推理/训练中的拒绝服务攻击，分段错误，在数据流图中插入恶意操作后，不影响模型的正常功能，也就是说模型的使用者从黑盒角度是没有感知的
  - Pytorch、Keras 和 Caffe P44 ~ 45：内存泄漏，权重丢失隐患，SQL注入漏洞
- **环境接触带来的漏洞**
  - 1. 第三方基础库漏洞 P47 ~ 50：Numpy 拒绝服务，OpenCV 堆溢出
  - 2. 可移植软件容器漏洞 P51 ~ 52：Kubeflow 挖矿，部署恶意容器