第一章

1. 2018 年 5 月 25 日，GDPR 在欧盟全面实施。
2. 2021 年，我国颁布实施了两部数据安全方面的基础性法律：《数据安全法》和《个人信息保护法》。
3. 数据是数字经济时代核心的生产要素
4. 隐私保护不仅要保护用户数据的机密性，还要保护用户行为数据的 隐私（或匿名性），也就是对用户元数据的保护。
5. 安全与隐私有什么区别?
   - **关注点不同**：
     - **安全**：侧重于保护资产（数据、系统）的机密性、完整性和可用性，主要防止未授权的访问和破坏（如黑客攻击）。
     - **隐私**：侧重于个人对自己信息的控制权（“被遗忘权”、“数据转移权”等），主要关注数据是否被合法、正当地使用，防止权力的滥用（如“老大哥”监控或企业滥用数据）。
   - **属性维度**：安全通过机密性等技术手段来保障；而隐私则更广泛，不仅需要机密性，还包括匿名性等属性来保护用户身份和行为。
6. 简述个人数据安全与个人隐私的关系。
   - **安全是隐私的基础**：安全手段（如加密）能够保证信息的机密性，是实现隐私保护的技术保障。如果数据缺乏安全保护（如被窃取），隐私必然受损。
   - **隐私泄露会危害安全**：个人隐私信息的泄露（如生日、家庭情况），会使攻击者更容易发起定向攻击（如口令猜测），从而危害整体数据安全。
   - **区别**：安全并不等同于隐私。一个系统可能非常安全（防黑客），但如果拥有者（如服务商或政府）滥用数据监控用户，则侵犯了隐私。
7. 简述技术进步与隐私保护的关系。
   - **带来威胁**：每一次技术变革都衍生出新的侵犯方式（如元数据分析），保护重心从物理空间被迫转向信息空间。
   - **提供手段**：技术同时也发展出加密、隐私计算等防御工具。
   - **挑战升级**：但因数据权属分离和无孔不入的采集，大数据时代的隐私保护难度空前。

第二章

1. 解释完善保密加密的概念。
   完善保密（也称无条件安全）是指密文中不包含明文的任何信息。
   - **核心定义**：即便攻击者拥有无限的计算资源，也无法从密文中恢复出明文，因为对于任意密文，任何明文都有可能生成该密文且概率相同 。
   - **实现条件**：香农证明了要实现完善保密，密钥必须满足以下特征：
     - **完全随机**；
     - **不能重复使用**（一次一密）；
     - **密钥长度至少与明文一样长** 。

2. 设计密码算法的启发式方法有哪些？
   提到的朴素的**启发式设计方法**主要有三种思路：
   - **代换**：将明文的每一个字符替换成密文的一个字符（如凯撒密码）。
   - **置换**：保持明文字母不变，但打乱其顺序（也称换位密码）。
   - **噪声**：在加密时加入噪声（如随机序列），让明文信号淹没在噪声里（流密码的设计思路）。
3. 解释香农的乘积密码设计思想。我们使用的密码算法中有哪些是基于这种设计思想的？
   - **设计思想**：香农在 1949 年提出，通过"乘积"（组合）多种简单的密码体制来构建更安全的系统。具体来说，就是将**代换** 和 **置换** 结合使用（即 SP 结构）。代换用于混淆，置换用于扩散。
   - **典型算法**：我们常用的分组密码如 **DES** 和 **AES** 都是基于这种设计思路设计的
4. 选择分组密码的工作模式时应遵循的基本原则有哪些？
   选择工作模式应遵循以下原则：
   - **避免使用 ECB 模式**：ECB 也是确定性的，无法隐藏明文的统计规律（相同的明文块生成相同的密文块），不具备语义安全（IND-CPA）。
   - **实现语义安全 (IND-CPA)**：应选择引入了随机性（如初始化向量 IV）的模式，如 **CBC** 或 **CTR** 模式。必须确保 **IV 是随机的且唯一的**（不可重复使用），否则会泄露信息。
   - **防止选择密文攻击 (CCA)**：为了防止攻击者篡改密文（可塑性），应使用**认证加密 (AE)** 模式。推荐使用 **GCM**或**Encrypt-then-MAC** 方案，它们能同时保证机密性和完整性。
5. 简述利用生日攻击的方法攻击哈希函数的过程。
   - **攻击目标**：攻击哈希函数的**抗碰撞性**，即寻找两个不同的消息 M_1 和 M_2，使得它们的哈希值相同。
   - **攻击过程**：基于**鸽巢原理**（生日悖论），攻击者不需要遍历所有可能的哈希值。对于输出长度为 m 比特的哈希函数，攻击者只需生成大约 $2^{m/2}$ 个随机消息（而不是 2^m），就有很高的概率（大于 50%）发现至少一对消息具有相同的哈希值。这使得寻找碰撞比寻找原像要快得多。

第三章

1. 填空题

1. TLS 协议可分为 ***握手协议*** 和 ***记录协议*** 。
2. TLS 协议使用 **DHE（或ECDHE）** __ 来保证前向安全性。
3. HMAC 的工作原理是 **MAC=hash(key+hash(key+message))，将密钥与消息混合后进行两次哈希运算）** __ 。
4. 采用 HSTS 策略的网站将保证浏览器始终连接到该网站的 ***HTTPS*** 加密版本。
5. DTLS 提供了与 SSL 近似的加密保障，但保留了底层高效的 **UDP（或数据报）** __ 传输机制。

2. 选择题

1. TLS 握手协议的任务包括（A、B、C）。
   - A. 协商密钥规格
   - B. 利用公钥证书来认证服务器的身份
   - C. 生成会话密钥
   - D. 用会话密钥加密传输的数据
2. 以下 HTTPS 的部署模式中，（ D）是最安全的。
   - A. HTTP 和 HTTPS 并存
   - B. HTTP 默认跳转到 HTTPS
   - C. 持久跳转 HTTPS (HSTS)
   - D. HSTS + HSTS Preload
3. 以下（A、B）技术是用来解决仿冒证书问题的。
   - A. CT（证书透明度）
   - B. HPKP（公钉孔）
   - C. CDN
   - D. ACME 协议

3. 术语解释

1. HSTS。
   - 即HTTP严格传输安全协议。它是一个安全功能，告诉浏览器只能通过HTTPS协议访问当前网站，禁止使用HTTP方式，从而减少会话劫持风险。
2. CDN。
   - 即内容分发网络。在HTTPS场景下，CDN作为“中间人”，代理用户和源站之间的通信。它引入了复杂的认证问题，因为CDN节点需要代表源站与用户建立HTTPS连接，通常涉及证书的共享或Keyless SSL技术。

4. 简答题

1. 简述 TLS 1.3 协议与以前版本的 TLS 协议在握手流程上的区别。
   - **握手时延降低**：TLS 1.3 将握手减少到 **1-RTT**（单次往返），而TLS 1.2通常需要2-RTT。它还支持 **0-RTT** 模式，允许在会话恢复时立即发送加密数据。
   - **算法精简**：TLS 1.3 删除了安全性较弱的算法（如RSA密钥交换、RC4、MD5等），仅保留安全的AEAD算法和ECDHE/DHE密钥交换。
   - **ServerHello 简化**：TLS 1.3 的 ServerHello 消息确定了密钥交换参数，使得客户端可以直接计算密钥，减少了往返次数
2. 简述 HSTS Preload List 的作用与工作原理。
   - **作用**：解决 HSTS 机制在用户**首次**访问网站时（浏览器尚未收到HSTS头）可能面临的 SSL 剥离或劫持风险。
   - **工作原理**：浏览器厂商内置一份“HSTS 预加载列表”（Preload List）。对于列表中的域名，浏览器在安装时就已经知道必须使用 HTTPS，因此即使用户是第一次访问，浏览器也会强制使用 HTTPS 连接，而不会尝试 HTTP。
3. 简述 Web PKI 的工作原理。

- **核心机制**：基于数字证书和信任链。CA（证书颁发机构）为网站颁发包含公钥和身份信息的数字证书，并用CA的私钥进行签名。
- **验证流程**：浏览器内置了受信任的根证书列表。当访问HTTPS网站时，浏览器接收服务器证书，并沿着信任链（从站点证书 -> 中间证书 -> 根证书）逐级验证签名，直到追溯到受信任的根证书，从而确认服务器身份合法。

4. 简述 ACME 协议的工作原理。
   - **定义**：自动化证书管理环境，用于自动化处理 CA 与 Web 服务器之间的交互。
   - **流程**：
     - **注册与验证**：客户端向 CA 注册并申请证书。
     - **挑战（Challenge）**：CA 发送一个随机 Token，要求客户端将其部署在 Web 服务器的特定路径（HTTP Challenge）或 DNS 记录中。
     - **验证与签发**：CA 远程访问该 Token 验证域名所有权，验证通过后自动签发证书。
5. 简述 CT 和 PKP 的工作原理。
   - **CT**：
     - **原理**：CA 将签发的所有证书提交到公开的日志服务器（Log Server）。日志服务器返回"已签名证书时间戳"（SCT）。浏览器在握手时会检查证书是否包含 SCT，且监控者会审计日志以发现非法签发的证书。其目的是让证书签发过程公开透明，便于发现伪造证书。
   - **PKP**：
     - **原理**：网站通过 HTTP 响应头（`Public-Key-Pins`）告诉浏览器该网站使用的证书公钥指纹（或中间证书/根证书的指纹）。浏览器将此指纹"钉扎"在本地，未来访问该网站时，强制验证证书链中是否包含该指纹。如果不匹配（即使证书由合法CA签发），浏览器也会断开连接，从而防范中间人攻击。
6. 简述 Noise 协议框架与 TLS 的不同。
   - **TLS**：是**协商驱动**的通用协议。它支持多种加密套件和向后兼容，非常灵活，但也因此**复杂、沉重**且代码量大，主要适用于Web浏览器等通用场景。
   - **Noise**：是一个**协议构建框架**而非单一协议。它允许开发者组合基础原语构建定制协议，通常**不支持动态协商**（双方预设参数），因此比TLS更**轻量、简洁**且攻击面小，常用于VPN（如WireGuard）、嵌入式设备及即时通讯。

第四章

1. 填空题

1. 身份认证中的四个基本原语分别是 ***所知***、***所有***、***特征***、***行为***。
2. FIDO 协议主要包含 ***UAF*** 和 ***U2F*** 两套协议。
3. Kerberos 协议的主要概念中，票据 (Ticket) 是用于向服务器认证身份的凭证。主要有两类，分别是 ***TGT（票据授予票据）*** 和 ***ST（服务票据）***。
4. OAuth 2.0 主要涉及四种角色，分别是：***资源拥有者***、***客户端***、***授权服务器*** 和 ***资源服务器***。
5. OPAQUE 协议可以抵抗 ***预计算（或离线字典攻击）*** 攻击。

2. 选择题

1. 如果是 68 个可用字符，需要长度为（C）的口令才能达到 128 位安全长度。
   - A. 8
   - B. 16
   - C. 21
   - D. 32
2. 存储口令时，更加安全的口令存储方案是：故意增加密码计算所需资源和时间，使得任何人都不可获得足够的资源建立暴力破解所需的彩虹表。这类方案的常用算法有三种，分别是（A、B、C）。
   - A. PBKDF2
   - B. BCRYPT
   - C. SCRYPT
   - D. BeyondCorp
3. 用来实现单点登录的协议包括（A、B、C）。
   - A. Kerberos 协议
   - B. OAuth 协议
   - C. OpenID Connect 协议
   - D. aPAKE 协议

3. 术语解释

1. MFA
   - 多因素身份认证。一种安全系统，要求用户在登录过程中提供两个或多个不同类别的验证因素（如密码、手机验证码、指纹等）来验证身份，以提高账户安全性。
2. Cookie。
   - Cookie 是一小段由 Web 服务器发送并存储在用户浏览器上的数据（包含名称、值、过期时间、路径等属性）。当用户后续访问同一网站时，浏览器会根据匹配规则将 Cookie 发回服务器。它主要用于**会话管理**（如保持登录状态 Auth-cookies）、**个性化设置**以及**在线追踪。**

4. 简答题

1. 简述口令认证的优点和缺点。
   - **优点**：
     - **成本低廉且易于实施**：不需要额外的硬件设备（如令牌），系统部署和维护成本较低。
     - **通用性强**：几乎所有的操作系统和应用程序都原生支持，用户熟悉度高，无需专门培训。
     - **用户可控**：用户可以自主创建、修改和重置密码，拥有对自己凭证的控制权。
   - **缺点**：
     - **安全性较弱**：容易受到暴力破解、字典攻击、撞库（密码复用）以及网络钓鱼等攻击。

    - **管理成本高**：用户容易忘记密码，导致频繁的密码重置请求，增加了IT支持的负担。
    - **用户体验问题**：复杂的密码策略可能导致“密码疲劳”，促使用户记录密码或使用弱密码。
2. 简述基于硬件安全的数字证书的优点和缺点。
    - **优点**：
        - **极高的安全性**：私钥存储在硬件安全模块（如 USB Key、智能卡）中，无法被导出或复制，有效防止私钥泄露。
        - **抗网络钓鱼**：基于硬件的认证通常需要物理接触（如触摸），远程攻击者难以通过网络钓鱼窃取凭证。
        - **完整性保障**：常用于代码签名或高等级身份认证，确保数据来源真实且未被篡改。
    - **缺点**：
        - **成本较高**：每个用户都需要购买物理设备，大规模部署费用昂贵。
        - **便携性与易用性差**：用户必须随身携带硬件，一旦丢失或损坏将导致无法登录（拒绝服务），且需要重新分发设备。
        - **兼容性问题**：需要特定的接口（如 USB-A/C）或驱动程序支持，在移动设备上使用可能不便。
3. 简述 FIDO2 与 FIDO UAF 和 FIDO U2F 之间的区别和联系。
    - 区别
        - **FIDO U2F**：主要用于**第二验证因子**。用户先输入密码，再使用硬件密钥（如 YubiKey）进行物理验证，侧重于抗钓鱼。
            - **FIDO UAF**：主要用于**无密码认证**。利用本地生物识别（指纹、FaceID）代替密码，侧重于移动端的便捷登录。
            - **FIDO2**：是 FIDO 技术的最新演进，包含了 **WebAuthn**（Web API）和 **CTAP**（设备协议）。它不仅向后兼容 U2F，还支持完整的**无密码登录**（Passkeys），适用于 Web 和桌面环境。
    - 联系
        - FIDO2 是 U2F 和 UAF 理念的集大成者和继任者。现在的 FIDO2 认证器通常同时支持 U2F 协议（作为 2FA）和 FIDO2 协议（作为无密码登录）。
4. 简述 OAuth 2.0 与 OpenID Connect 在工作流程上的区别。
    - **OAuth 2.0 (授权)**：
        - **目标**：授权第三方应用访问受保护的资源（如“读取你的通讯录”）。
        - **核心凭证**：返回 **Access Token**（访问令牌）。
        - **流程**：客户端请求授权 -> 用户同意 -> 获取授权码 -> 换取 Access Token -> 访问 API。它不直接关心“用户是谁”，只关心“用户准许了什么”。
    - **OpenID Connect (OIDC, 认证)**：
        - **目标**：验证用户的身份（如“使用 Google 登录”）。
        - **核心凭证**：在 OAuth 2.0 的基础上，多返回一个 **ID Token**（通常是 JWT 格式），并提供 **UserInfo Endpoint**。

- **区别**：OIDC 是构建在 OAuth 2.0 之上的身份层。在工作流程中，OIDC 的授权请求中会包含 `openid` 这个 Scope，且最终响应中除了 Access Token 外，必定包含包含用户身份信息的 ID Token。

5. 简述 Cookie 劫持攻击的基本原理。
   - **被动攻击**：攻击者在公共网络监听未加密的 HTTP 流量，直接捕获明文 Cookie，进而重放 Cookie 冒充用户登录 。
   - **主动攻击**：利用中间人或 XSS 注入恶意内容，强制浏览器发起 HTTP 请求，从而暴露未设置 Secure 属性的 Cookie 。
6. 简述 OPAQUE 协议的基本原理。
   OPAQUE 是一种**非对称 PAKE 协议**，无需在网络上传输密码。
   - **核心机制**：客户端利用密码与服务器执行 **OPRF（不经意伪随机函数）** 生成密钥（服务器无法看到密码），用该密钥解密服务器存储的"加密包络"以恢复私钥，最后完成标准密钥交换。
   - **优势**：服务器仅存加密包络而非密码哈希。即使数据库泄露，攻击者因缺失服务器私钥，**彻底无法进行离线字典攻击。**

第五章

1. 填空题

1. 云计算的三种主要服务模式包括：***SaaS (软件即服务)***、***PaaS (平台即服务)***、***IaaS (基础设施即服务)***。
2. 容器技术的三个核心概念包括：***镜像***、***容器***、***仓库***。
3. 如果一个容器可以访问到外面的资源，甚至获得宿主主机的权限，这就叫做 "***容器逃逸（Docker逃逸）***"。
4. 英特尔 CPU 中存在的漏洞具有两种攻击模式，分别是**Meltdown (熔断)**、**Spectre (幽灵)**___。
5. 在谷歌的大数据处理系统中，最广为人知的三项云计算技术是__GFS (Google File System)、MapReduce、BigTable___。

2. 选择题

1. CAP 定理是指不可能同时实现以下（A、B、C ）三种属性。
   - A. 一致性
   - B. 可用性
   - C. 分区容错性
   - D. 持久性
2. ACID 事务提供以下保证：（A\B、C、D ）。
   - A. 原子性
   - B. 一致性
   - C. 隔离性
   - D. 持久性
3. BASE 模型中的 BASE 是指（A、B、C）。

- A. 基本可用
- B. 软状态
- C. 最终一致性
- D. 隔离性

3. 术语解释

1. 沙箱容器。
   - 一种通过创建更强信任边界（如独立内核或轻量级虚拟机）来解决传统容器隔离性不足的技术。它将容器与宿主机操作系统隔离开来，在提供接近虚拟机强隔离性的同时，保留了容器的高效率。常见方案包括 gVisor 和 Kata Containers。
2. PIR。
   - 隐私信息检索。一种加密协议，允许用户从数据库中检索特定数据项，同时不向服务器泄露用户具体检索的是哪一项数据。它是保护用户查询隐私的关键技术。

   4. 简答题
3. 简述虚拟机与容器的区别。
   - **操作系统支持**：虚拟机拥有独立的客户机操作系统 (Guest OS)，运行在 Hypervisor 之上；容器与宿主机共享操作系统内核，更为轻量。
   - **体积与性能**：虚拟机镜像庞大 (GB级)，占用更多 CPU 和内存资源；容器镜像小 (MB级)，几乎没有额外的性能损失。
   - **启动速度**：虚拟机启动较慢 (秒级甚至分钟级)；容器启动极快 (毫秒级或秒级)。
   - **隔离性**：虚拟机提供硬件级隔离，安全性更高；容器提供进程级/OS级隔离，安全性相对较弱。
4. 简述造成容器逃逸的两类原因与应对措施。
   - **原因**：
     1. **软件漏洞**：主要是宿主机内核漏洞（如 Dirty COW）或容器运行时软件的设计缺陷（如 runc 漏洞 CVE-2019-5736），导致攻击者通过漏洞获取宿主机权限。
     2. **错误配置**：例如部署了特权容器（Privileged Container），或挂载了关键目录（如 `/var/run/docker.sock` 或 `/proc`），使攻击者能直接操作宿主机。
   - **应对措施**： 采用**沙箱容器**技术（如 gVisor, Kata Containers, Firecracker），构建真正的强信任边界，将容器与宿主机的内核隔离开来。
5. 简述 Meltdown 和 Spectre 的区别。
   - **Meltdown (熔断)**：打破了**用户应用程序与操作系统**之间的隔离。它允许用户态的程序访问内核态的内存数据，从而窃取操作系统或其他程序的秘密。主要影响 Intel CPU，较易通过软件补丁修复（如 KPTI）。
   - **Spectre (幽灵)**：打破了**不同应用程序**之间的隔离。它利用投机执行诱骗合法的程序泄露其自身的秘密数据。Spectre 比 Meltdown 更难利用，但也更难修复，通常需要重编译软件或修改硬件设计。
6. 简述 POR 技术的基本原理。
   - **POR (可检索性证明)** 用于验证云端数据的完整性并确保数据可恢复。

- **原理**：用户预先在文件中植入随机的“岗哨”块。验证时，通过挑战服务器返回特定岗哨内容来探测数据是否损坏。该技术通常结合纠删码（如 RS 码）以实现数据恢复。

7. 简述加密去重技术的基本原理。
   - **背景矛盾**：传统加密会使相同的数据生成不同的密文（随机性），导致无法去重；而去重需要识别重复数据。
   - 原理 (基于收敛加密 CE 或 MLE)：加密密钥直接来源于数据本身（例如 $Key = Hash(Message)$）。因为相同的明文 M 必定生成相同的密钥 K，进而生成相同的密文 C。这样，云服务器可以通过比较密文的哈希值来识别重复数据，既实现了数据的加密保护，又保留了去重的能力。

第六章

**1. 填空题

1. 安全多方计算模型可以分为 ***半诚实模型*** 和恶意模型。
2. SGX 是 Intel 架构新的扩展，在原有架构上增加了 ***一组新指令集*** 和 ***内存访问控制机制*** 。
3. 可信操作系统启动包括两种情况： **安全启动**__ 和 **可信启动（或度量启动）**__ 。

2. 选择题

1. SGX 与 TrustZone 技术相比，其优势为（B）。
   - A. 支持虚拟化技术、容器技术
   - B. 将系统的可信计算基缩小到 CPU
   - C. SGX 可抵御侧信道攻击
   - D. 通过内存泄露攻击获取关键信息的难度更大

3. 术语解释

2. 同态加密。

- 同态加密是一种特殊的加密体制，允许在不解密的情况下对密文进行特定运算（如加法、乘法）。其结果解密后，与直接对明文进行相应运算的结果一致。这使得云服务器可以在不知晓数据内容（明文）的情况下对数据进行处理。

3. 安全多方计算。

- 指在无可信第三方的情况下，多个参与方（m个）利用各自的私有输入共同计算一个约定函数 $f(x_1, \ldots, x_m)$ 的过程。该技术保证了计算的正确性，同时确保每个参与方除了计算结果外，无法获知其他方的私有输入信息。

4. 零知识证明。

- 指证明者 (Prover) 能够在不向验证者 (Verifier) 透露任何有用信息（除了“该命题为真”这一事实外）的情况下，使验证者相信某个论断是正确的。

5. 混淆电路。

- 是安全多方计算的基础协议之一（通常用于两方）。加密方（Garbler）将计算逻辑编译成布尔电路，并对电路中的每个门（真值表）进行加密和打乱（混淆）。计算方（Evaluator）通过不经意传输 (OT) 获取输入对应的密钥，对混淆电路进行解密求值，从而得到计算结果而不泄露输入。

4. 简答题

1. 简述 Rich OS、TEE 与 SE 的异同。
   - **Rich OS (REE):** 普通操作系统（如 Android, iOS），功能最丰富，性能最强，但安全性最低，TCB（可信计算基）很大，容易受攻击。
   - **TEE:** 可信执行环境（如 TrustZone, SGX），提供隔离的执行环境，安全性高于 Rich OS，TCB 较小，能兼顾一定的性能和灵活性。
   - **SE:** 安全单元（如智能卡芯片），安全性最高，具有物理防篡改能力，但计算能力和存储空间非常有限，通常用于存储密钥。
2. 在 Paillier 加密算法的正确性与安全性方面：
   - ① 为什么 Paillier 加密算法是语义安全的?
     - Paillier 是概率加密算法。在加密过程中引入了随机数 r，使得同一个明文在不同次加密时会生成不同的密文。其语义安全性（IND-CPA）基于**判定合数剩余类假设 (DCRA)**，即区分 n^2 模下的 n 次剩余是困难的。
   - ② 为什么 Paillier 加密算法密文长度是其明文长度的 2 倍?
     - 因为 Paillier 的明文 m 是在 $\mathbb{Z}_n$ 空间（长度约为 $|n|$），而密文 c 是在 $\mathbb{Z}_{n^2}^*$ 空间计算的（模 n^2）。由于 n^2 的比特长度是 n 的两倍，所以密文长度是明文的 2 倍。
   - ③ 对于同一个明文 m，为什么每次 Paillier 密码算法加密出来的密文不同，却都能够正确解密出明文 m? 请予以证明。
     - **加密公式：** $c = g^m \cdot r^n \mod n^2$。
     - **原因：** 尽管随机数 r 使密文不同，但在解密过程中，通过私钥 λ 运算 $c^\lambda \mod n^2$，根据公式推导会得到 $c^\lambda \equiv (g^m r^n)^\lambda \equiv g^{m\lambda} r^{n\lambda} \mod n^2$。
     - 由于 $r^{n\lambda} \equiv 1 \mod n^2$（基于相关数论定理），$r$ 的影响被消除，仅保留了含 m 的项，从而能恢复出唯一的 m。
3. 关于同态加密：
   - ① 简述同态加密的定义。
     - 记加密操作为 E，解密为 D。如果针对明文的某种操作 f，存在针对密文的操作 F，使得 $D(F(E(m_1), E(m_2), \dots)) = f(m_1, m_2, \dots)$，则称该加密算法对操作 f 具有同态性。
   - ② 证明 RSA 算法是乘法同态的。
     - RSA 加密 $E(m) = m^e \mod n$。
     - 设有两个密文 $c_1 = m_1^e \mod n, c_2 = m_2^e \mod n$。
     - 计算 $c_1 \cdot c_2 = m_1^e \cdot m_2^e = (m_1 \cdot m_2)^e \mod n$。
     - 这也是 $m_1 \cdot m_2$ 的加密结果，即 $E(m_1) \cdot E(m_2) = E(m_1 \cdot m_2)$，故满足乘法同态。
   - ③ 证明 Paillier 算法是加法同态的。
     - Paillier 加密 $E(m, r) = g^m \cdot r^n \mod n^2$。
     - 设有两个密文 $c_1 = g^{m_1} r_1^n \mod n^2, c_2 = g^{m_2} r_2^n \mod n^2$。
     - 计算 $c_1 \cdot c_2 = (g^{m_1} r_1^n) \cdot (g^{m_2} r_2^n) = g^{m_1+m_2} \cdot (r_1 r_2)^n \mod n^2$。

- 这正是明文和 m_1+m_2 (随机数为 r_1r_2) 的加密形式，即 $D(c_1\cdot c_2)=m_1+m_2$，故满足加法同态。

5. 代码分析题

题目： 找出下列 Paillier 同态加密实现中的错误。

```
def Encryption(m):
    n,g = pk
    r = random.randint(0,n)
    c = pow(g,m,n**2)* pow(r,n,n**2)% (n ** 2)
    return c
```

代码中 **随机数 `r` 的生成方式错误。**

1. **错误详情**：`r = random.randint(0,n)` 可能会生成 0 或者与 n 不互素的数（即 $\gcd(r,n)\neq 1$）。
2. **正确要求**：根据 Paillier 算法定义，随机数 r 必须满足 $0<r<n$ 且 $r\in\mathbb{Z}_n^*$，即必须保证 $\gcd(r,n)=1$。
3. **后果**：如果 r 与 n 不互素（例如 $r=0$ 或 r 是 p,q 的倍数），将导致解密失败或安全性丧失（如PPT示例中提到的 $r=3,n=15$ 导致解密错误的情况）。

第7章

1. 填空题

1. 在具体应用中，***隐私*** 即数据所有者不愿意披露的敏感信息，包括敏感数据以及数据所表征的特性。
2. 若 s 表示敏感数据，事件 S_k 表示"攻击者在背景知识 K 的帮助下揭露敏感数据 s"，则披露风险 $r(s,K)$ 表示为 $Pr(S_k)$。
3. 对于任意一对相邻数据库（定义为差别最多有一个记录的两个数据库）D_1 和 D_2，任意一个可能的带噪中间件 S，一个提供 ε-差分隐私保护的算法 M 必须满足不等式 $Pr[M(D_1)=S]\leq e^{\varepsilon}\cdot Pr[M(D_2)=S]$。
4. 本地化差分隐私的定义为：给定 n 个用户，每个用户拥有一条记录，如果算法 M 对任意两条记录 t 和 t'，得到相同输出结果 t^* 的可能性满足不等式 __，则说明其满足 ε-本地化差分隐私。
5. **RAPPOR**__ 算法是谷歌提出的基于随机响应的一个通用差分隐私算法，主要用于收集用户的数据并保证用户的隐私。

2. 选择题

1. （A、B、C ）是常见的隐私保护模型。
   - A. k 重匿名
   - B. l-多样化
   - C. t 相近

- D. 高斯机制

2. （A ）机制是中心化差分隐私领域的基础机制。
   - A. 拉普拉斯机制
   - B. 高斯机制
   - C. 指数机制
   - D. 随机响应机制
3. 差分隐私使用（ C、D）添加随机扰动的方式，来保证机器学习模型的隐私保护。
   - A. 输入扰动
   - B. 中间参数扰动
   - C. 目标扰动
   - D. 输出扰动

3. 术语解释

1. 抑制。
   - 指在发布数据时，对特定的标识符或属性（如显式标识符）不做发布处理（即删除或隐藏），以防止身份识别。
2. 泛化。
   - 指将数据集中特定的属性值（如准标识符）用更概括、抽象的值来替代具体描述值（例如将具体的出生日期概括为年份），核心思想是用一个不确切但忠于原值的值来代替原值，以实现隐私保护

   4. 简答题
3. 简述k重匿名如何抵御记录链接攻击。
   - k 重匿名要求在发布的数据表中，如果一个记录包含某个准标识符（QID）值序列，那么至少还有 $k-1$ 个其他记录也包含相同的QID值序列。
   - 这样，攻击者即使掌握了受害者的背景知识（QID），也无法将受害者与表中的特定记录唯一对应起来（最多只能定位到一个包含 k 条记录的等价组），从而通过降低识别精度来抵御记录链接攻击。
4. 简述l-多样化如何抵御属性链接攻击。
   - l-多样化是为了解决 k-匿名中同一等价组内敏感属性值可能单一的问题。
   - 它要求每一个等价组（QID相同的组）内，敏感属性至少包含 l 个“表现良好”的（不同的）值。
   - 这样，即使攻击者定位到了受害者所在的组，也无法确定受害者具体对应哪一个敏感属性值（因为有 l 种可能性），从而阻断属性链接攻击。
5. 分别简述拉普拉斯机制、高斯机制和指数机制的基本原理，以及三者之间的区别。
   - **拉普拉斯机制**：针对**数值型**查询结果，通过向真实结果中加入服从拉普拉斯分布的噪声来实现隐私保护。噪声大小取决于敏感度 Δf 和隐私预算 ϵ。
   - **指数机制**：针对**非数值型**（离散型）的查询结果（如“选出最好的那个”），通过定义评分函数，以正比于分数的指数概率来随机选择输出结果。
   - **高斯机制**：针对数值型查询，通过加入高斯分布（正态分布）的噪声来实现，通常满足 (ϵ, δ)-差分隐私（近似差分隐私）。

- **区别**：
  - 拉普拉斯机制适用于数值输出，满足严格 ϵ-差分隐私。
  - 指数机制适用于非数值输出，满足严格 ϵ-差分隐私。
  - 高斯机制适用于数值输出，但通常只满足近似 (ϵ, δ)-差分隐私（允许极小概率违反隐私）。

6. 简述中心化差分隐私和本地化差分隐私的区别。
   - **中心化差分隐私 (CDP)**：依赖可信第三方收集原始数据，统一加噪后发布。优点是噪声少、数据可用性高；缺点是第三方被攻破会导致原始数据泄露。
   - **本地化差分隐私 (LDP)**：假设收集者不可信，用户先在本地加噪再上传。优点是无需可信第三方、更安全；缺点是噪声叠加严重，需更多数据量以保证可用性。
7. 分别简述随机响应机制和一元编码机制的基本原理。
   - **随机响应机制**：用户掷硬币决定回答方式：正面如实回答，反面随机作答。以此引入不确定性，使攻击者无法辨别答案真伪。
   - **一元编码机制**：将数据编码为位向量（如独热码），按设定概率随机翻转向量中的每一位进行扰动，再上传给收集者聚合。
8. 简述 RAPPOR 算法的具体步骤。
   - RAPPOR 是基于一元编码和随机响应的算法，主要包含以下步骤：
     1. **编码**：将用户数据（如字符串）映射或哈希为一个位向量（通常是Bloom Filter或独热编码）。
     2. **扰动**：在本地对编码后的位向量进行随机响应处理（翻转比特位），以满足差分隐私。PPT中提到通过概率 p 和 q 翻转比特位。
     3. **聚合**：收集者接收所有用户的扰动向量，按位求和，并利用统计方法（校正）去除噪声影响，估算出原始数据的分布（如直方图）。