1) 在 DSS数字签名标准中，参数选取 p=83 q=41 g=4 mod 83，若 Alice 的私钥 x 为 16

Alice 的公钥 $y = g^x \mod p = 77$

签名过程

$r = (g^k \mod p) \mod q = 10$

$k^{-1} = 25$

$s = (H(m) + xr)k^{-1} \pmod q = 29$

签名为 (m, r, s)

验证过程

$s^{-1} = 17$

$(g^{H(m)s^{-1} \mod q} y^{rs^{-1} \mod q} \mod p) \mod q = (4^{56 \cdot 17 \mod 41} 77^{10 \cdot 17 \mod 41} \mod 83) \mod 41 = (4^9 77^6 \mod 83) \mod 41 = 51 \mod 41 = 10 = r$

故签名有效

2) 叙述基于 hash 的 RSA 签名算法的过程；阐述 hash 函数的三个基本性质；分别说明基

于 hash 的 RSA 签名算法如何 1. 抵抗唯密钥攻击 2. 抗已知消息攻击 3. 抗选择消息攻击的

- **基于 hash 的 RSA 签名算法**

  选取素数 p, q **构造** $n = pq, \varphi(n) = (p-1)(q-1)$

  选取 e **使得** $(e, \varphi(n)) = 1$

  计算 d **使得** $ed \equiv 1 \pmod{\varphi(n)}$

  密钥为 (n, e)

  签名 $s = m^d \pmod{n}$

  验签 $m' = s^e \pmod{n}$

  若 $m' = m$ **则签名有效**

- **hash 函数的三个基本性质**

  抗原象：给定 $H(m)$ **难以通过计算找到** m

  抗第二原象：给定 $x, H(x)$ **难以通过计算找到** $H(x) = H(y)$

  抗碰撞：难以通过计算找到 x, y **使得** $H(x) = H(y)$

  抗原象攻击复杂度 $O(2^{128})$

  抗第二原象攻击复杂度 $O(2^{128})$

抗碰撞攻击复杂度 $O(2^{64})$

- 分别说明基于 hash 的 RSA 签名算法如何 1. 抵抗唯密钥攻击 2. 抗已知消息攻击 3. 抗选择消息攻击的
  - 抵抗唯密钥攻击

    攻击者已知密钥 e，挑选一个随机数作为签名 $s = k$，构造消息 $m = k^e$

    基于 hash 的 RSA 则需要构造 $H(m) = k^e$

    hash 函数具有抗原象性，难以构造 m
  - 抗已知消息攻击

    攻击者已知密钥 e 和部分消息签名对 $(m_1, s_1), (m_2, s_2)$

    由于 $s_1 = m_1^d, s_2 = m_2^d, (s_1 s_2)^e = s_1^e s_2^e = m_1 m_2$

    令签名 $s = s_1 s_2$ 即可构造消息 $m = m_1 m_2$ 的签名

    基于 hash 的 RSA 则需要构造消息 $H(m) =$

$H(m_1)H(m_2)$ 的签名

hash 函数具有抗原象性，难以构造 m

- 抗选择消息攻击

  攻击者已知密钥 e 和任意消息签名对 $(m_1, s_1), ..., (m_k, s_k)$

  由于 $s_i = m_i^d, (s_1...s_k)^e = s_1^e...s_k^e = m_1...m_k$

  令签名 $s = s_1...s_k$ 即可构造消息 $m = m_1...m_k$ 的签

  基于 hash 的 RSA 则需要构造消息 $H(m) = H(m_1)...H(m_2)$ 的签名

  hash 函数具有抗原象性，难以构造 m

3) 简述 DSA 签名体制的过程；说明签名者随机选取的 k 被泄露，或者 k 值重复使用的危

害性

选取随机数 $k < q$

计算 $r = (g^k \mod p) \mod q$

计算 $s = (H(m) + xr)k^{-1} \mod q$

签名 (m, r, s)

验签 $(y^{rs^{-1} \mod q} g^{H(m)s^{-1} \mod q} \mod p) \mod q = r$

k 被泄露时

由 $s = (H(m) + xr)k^{-1} \mod q$

可计算出私钥 $x = (sk - H(m))r^{-1} \mod q$

k 重复使用时

由 $s_1 = (H(m_1) + xr)k^{-1} \mod q, s_2 = (H(m_2) + xr)k^{-1} \mod q$

可计算出 $k = (H(m_1) - H(m_2))(s_1 - s_2)^{-1} \mod q$

4) 简述密钥管理采用层次化的结构的好处

1. 安全性增强
   - 密钥隔离：不同层级的密钥职责分离（如主密钥、密钥加密密钥、数据加密密钥），降低单一密钥泄露的影响范围。
   - 减少暴露风险：高层级密钥（如主密钥）不直接参与数据加密，仅用于派生或保护下层密钥，暴露机会减少。

2. 灵活性与可扩展性
   - 动态更新：下层密钥（如会话密钥）可频繁更换，而上层密钥无需频繁变动，适应大规模系统需求。
   - 分级管理：支持多级权限控制，不同层级由不同管理员负责，适合分布式或跨部门场景。
3. 效率优化
   - 减少主密钥使用频率：通过中间层密钥（KEK）保护大量数据密钥（DEK），避免主密钥频繁调用，提升性能。
   - 简化密钥分发：仅需安全分发高层级密钥，下层密钥可通过加密通道传输，降低开销。
4. 审计与合规性
   - 职责明确：层级划分便于跟踪密钥使用和访问权限，满足合规要求（如GDPR、FIPS）。
   - 故障隔离：某一层密钥问题不会直接影响其他层级，便于问题定位和恢复。
5. 生命周期管理
   - 差异化策略：不同层级可设置不同生命周期（如主密钥长期保存，会话密钥短期使用），优化管理成本。