作业06.

1. (1) $y \equiv g^x \equiv 4^{16} \equiv 77 \bmod 83$ 公钥 y 为 77.

(2) 签名：$r \equiv (g^k \equiv 4^{23} \equiv 51 \bmod 83) \equiv 10 \bmod 41$

$s = (h(m) + r \cdot x) \cdot k^{-1} \bmod q$.

$k \cdot k^{-1} \equiv 1 \bmod q$ 解得 $k^{-1} = 25$.

则 $s = 29$. 结果签名为 $(r, s) = (10, 29)$.

验证：$s^{-1} = 17 \pmod q$

$h(m) \cdot s^{-1} \equiv 16 \times 17 \equiv 9 \bmod 41$.

$g^{h(m) \cdot s^{-1}} \equiv 4^9 \equiv 30 \bmod 83$.

$r \cdot s^{-1} \equiv 10 \times 17 \equiv 6 \bmod 41$

$y^{r \cdot s^{-1}} \equiv 77^6 \equiv 10 \bmod 83$

$g^{h(m) \cdot s^{-1}} \cdot y^{r s^{-1}} \equiv 300 \equiv 51 \bmod 83$

$51 \equiv 10 \equiv r \bmod 41$. 验证成功.

2. (1) 参数设置+密钥生成：选取大素数 p, q. 计算 $n = p \times q$. 选取素数 e 保证 $\gcd(e, \varphi(n)) = 1$. 计算 $ed \equiv 1 \bmod \varphi(n)$.

公钥：(n, e)，私钥 (n, d).

同时选取一个安全哈希函数 H.

签名：对明文 m，先算 $H(m) = h$. 然后用私钥签名，即计算 $S \equiv h^d \bmod n$. 签名为 S.

验签：计算 $h' = H(m)$. $m' \equiv s^e \bmod n$.

比较 h' 和 $H(m')$，相等则签名有效，否则无效.

/ /

(2) 抗原像：给定哈希值 h，要找 m 使 $H(m)=h$ 在计算上不可行。

↳ 复杂度：2^{128}.

抗第二原像：给定 m_1，找 m_2 使 $H(m_2)=H(m_1)$ … 不可行.

↳ 复杂度：2^{128}

抗碰撞：找 (m_1, m_2) 使 $H(m_1)=H(m_2)$ … 不可行.

↳ 复杂度：2^{64}.

(3) 唯密钥：基于抗原像性，攻击者若想伪造签名 s' 需要找到 m 使 $H(m)=s'$ 使之有效.

已知消息～：抗第二原像性，攻击者基于 $(m_i, s_i)_{(i\in n)}$ 伪造 $(m', s_{i\neq})$.

即找到 m' 使 $H(m')=H(m_i)$ $(i\in n)$ 使之匹配上.

选择消息～：抗碰撞性，攻击者找 m_1, m_2 使得 $H(m_1)=H(m_2)$

其中 m_1 无害，m_2 有害使得签署者签署 m_1 的签名 S 可用于 m_2 造成危害.

3. (1) 参数设置：大素数 p. 选素数 q 使 $q|(p-1)$. g 为 $\mathbb{Z}_p^*$ 中 q 阶元素. h 为安全的哈希函数. ⇒ 公开.

密钥生成：随机 $x\in(0,q)$，算 $y\equiv g^x \bmod p$.

公钥：y ；私钥：x.

签名：对明文 m，随机 k. 计算 $(r\equiv g^k \bmod p) \bmod q$.

$s\equiv (h(m)+xr)\cdot k^{-1} \bmod q$.

* 签名为 (r,s).

验证：若 $r=(g^{ms^{-1} \bmod q} \cdot y^{rs^{-1} \bmod q} \bmod p) \bmod q$.

则有效，否则无效.

(2) k泄露：将签名方程变为 $k \cdot S \equiv h(m) + r \cdot x \pmod p$，其中 S，$h(m)$，r，k 均已知，则易求 x（私钥），进行导致系统安全性破坏。

k重复使用：$\begin{cases} kS_1 \equiv h(m_1) + r \cdot x \pmod p \\ kS_2 \equiv h(m_2) + r \cdot x \pmod p \end{cases}$

$\Rightarrow k(S_1 - S_2) \equiv h(m_1) - h(m_2) \pmod p$，只有k未知，易求。

即造成k泄露从而 x 私钥泄露。

4. ① 降低复杂性：只需信任和管理少量的主密钥即可；

② 可伸缩性好：使系统可支持大量实体不崩溃。

③ 保证安全一致性：集中控制整个体系并强制执行统一安全策略。

④ 容错性好：不会因一部分密钥系统被攻破而直接崩溃。