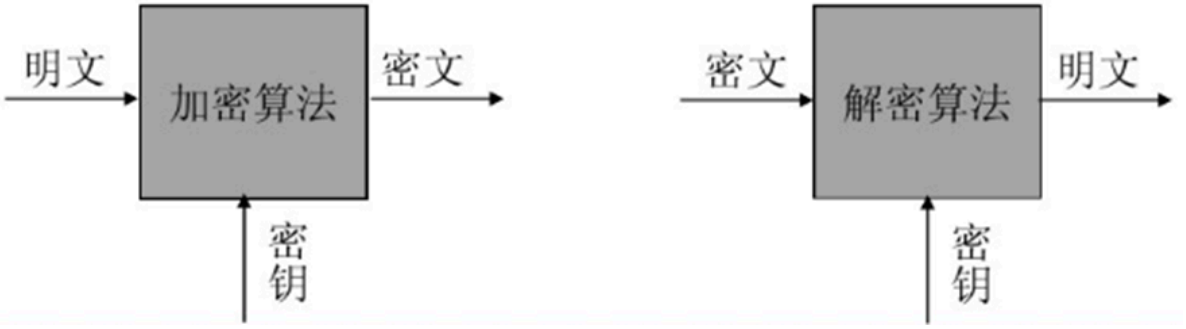
# 第七章 云安全机制

## 7.1 密码学基础(可能会考，再看看)

1. 剩余类定义：给定正整数m，全体整数可按照模m是否同余分为若干两两不相交的集合，使得每一个集合中的任意两个正整数对模m一定同余，而属于不同集合的任意两个整数对模m不同余，每一个这样的集合称为模m的剩余类。
2. **欧几里德算法**又称辗转相除法：用于计算两个整数a，b的最大公约数。
    1. 拓展：对于给定整数a,b，拓展的欧几里得算法不仅可以计算出最大公因子d, 而且可以得到两个整数x,y，并满足：`ax + by = d = gcd (a, b)`
3. 在模运算中，如果一个数a和模数m互质，那么a在模m下存在乘法逆元，表示为a的逆元，记作`a^-1`，满足以下条件：`a * a^-1 ≡ 1 (mod m)`

## 7.2 密码学发展及基本概念

1. 古典密码阶段
    1. 代替（Substitution）：明文中的字符被替换成密文中另一个字符
    2. 置换（Permutation）：明文字母不变，但顺序被打乱。
    3. 特点：密码学还不是科学，而是艺术。出现密码算法设计的基本手段（代替法&置换法）
    4. 保密性：数据的保密**基于加密算法**的保密
    5. 里程碑事件：1883年科克霍夫第一次明确出密码编码原则
2. 现代密码 I 阶段
    1. 时间：1949-1976
    2. 特点：密码学由艺术成为科学
    3. 里程碑事件：1949年Shannon发表“The Communication Theory of Secret Systems”
    4. 香农的贡献：
        1. 定义了理论安全性，提出扩散和混淆原则
        2. 奠定了密码学的理论基础
    5. 扩散：将每一位明文尽可能地散布到多个输出密文中去，以更隐蔽明文数字的统计特性
    6. 混淆：使密文的统计特性与明文密钥之间的关系尽量复杂化
    7. 新特点：数据的安全**基于密钥而不是算法的保密**
3. 现代密码 II 阶段
    1. 时间：1976-1994
    2. 特点：公钥开始出现
    3. 里程碑事件：
        1. 1976年Diffie&Hellman的“New Directions in Cryptography”提出了公钥密码的概念

    2. 1977年Rivest, Shmir&Adleman提出了RSA公钥算法
4. 未来发展--后量子密码:
    1. 在传统计算机上运行，为了抵御量子计算对密码系统的威胁而产生的算法
    2. 目前的研究主要包括：
        1. 基于编码的公钥密码
        2. 基于格的公钥密码
        3. 基于HASH的公钥密码
        4. 多变量公钥密码
4. 安全攻击分类：（以获取密钥和加解密算法为目的），攻击难度从下到上逐渐增加
    1. 唯密文攻击（Cipher-only attack）: 攻击者有**一个或多个密文**，攻击需要统计分析；
    2. 已知明文攻击（Known-plaintext attack）: 攻击者有**一份密文和对应的明文**，进行**算法和密钥推导**；
    3. 选择明文攻击（Chosen-plaintext attack）: 攻击者有机会使用**加密机**，因此可以**选择任何明文并产生对应的密文**，攻击概率更大；
    4. 选择密文攻击（Chosen-cipher attack）: 攻击者有机会使用**解密机**，因此可以选择一些密文并产生对应的明文。
5. 安全性：密钥长度**越长**，加密数据**越不容易**被非法解密
6. 相关概念：
    1. 数据按照一种可读的格式进行编码，这种格式称为**明文**(plaintext)。
    2. 数据在网络上传输时，为避免未授权的恶意访问，可以使用称为**加密部件**(cipher)的标准化算法，通过**密钥**把原始的明文数据转换为**密文**(ciphertext)，这个过程称为**加密**(encryption)。
    3. **解密** Decryption ： 从密文恢复出明文的过程
    4. 

7. 加密机制可以对抗**流量窃听、恶意媒介、授权不足和信任边界重叠**。

# 7.3 古典密码学

1. 对称加密：**加密与解密时使用同一个密钥**，这两个过程都是授权的各方用共享的密钥执行的。
    1. 由于用一个密钥加密的消息只能用同一个密钥解密，因此拥有密钥的被授权方才能创建消息，因此**保证了数据的保密性**。
    2. **对称加密不具备不可否认性**，当有多方使用同一个密钥时，无法判断是哪一方执行的加密和解密。
    3. 优点：算法公开、计算量小、加密速度快、加密效率高。

4. 缺点：事先协商密钥；多用户时密钥量巨大

2. 移位加密（凯撒加密）
    1. 算法：
        1. 由于英文字符有26个字母，可以建立英文字母和模26的剩余之间的对应关系：A = 0  B = 1  C = 2  …  Y = 24  Z = 25
        2. 加密过程：y = x + k  (mod 26)
        3. 解密过程：x = y - k  (mod 26)
        4. 其中，k就是加密密钥。凯撒用k=3进行加密。
    2. 攻击方法：
        1. 已知明文攻击：x,y都已知，那么k = y - x  (mod 26)
        2. 选择明文攻击：能够自己选择x。比如选择 x='a'=0 ，则为 y =  k  (mod 26)
        3. 选择密文攻击：自己选择密文y。例如选择 y='a'=0，那么 x =  - k  (mod 26)
    3. 凯撒加密的安全性
        1. 暴力穷举：把26种可能都试一遍
        2. 词频统计：
            1. 统计密文中字母出现的频率
            2. 与标准的语言字母出现的频率进行比对
            3. 确定密钥k的最可能值

3. 仿射加密
    1. 加密：给定密钥(α,β) (α, β∈Z 26)： `y=αx+β ( mod 26)`
    2. 解密：1/α不是α的倒数，而是乘法逆元。
        1. $$x = \frac{1}{\alpha}(y-\beta)\ (\mathrm{mod}\ 26)$$
    3. 逆元的简单定义：α的逆元是满足αγ=1 (mod 26)的γ的值
        1. 例如，α=7,n=26 则我们寻找a的逆元：得出 `7 * 15 = 1(mod 26)`，即15是7模26的逆元。
    4. 例：给定密钥(7，3)，对明文China进行加密和解密
        1. 加密函数： `y = f (x) = 7x + 3(mod26)`
        2. 解密函数： `x = f-1(y) = 7-1 (y-3)(mod26) = 15y-19(mod26)`
        3. 将China 转换为数字：2，7，8，13，0

4. $$7\times\begin{bmatrix}2\\7\\8\\13\\0\end{bmatrix}+\begin{bmatrix}3\\3\\3\\3\\3\end{bmatrix}=\begin{bmatrix}17\\52\\59\\94\\3\end{bmatrix}\bmod 26=\begin{bmatrix}17\\0\\7\\16\\3\end{bmatrix}=\begin{bmatrix}R\\A\\H\\Q\\D\end{bmatrix}$$
5. $$15\times\begin{bmatrix}17\\0\\7\\16\\3\end{bmatrix}-\begin{bmatrix}19\\19\\19\\19\\19\end{bmatrix}=\begin{bmatrix}236\\-19\\86\\221\\26\end{bmatrix}\bmod 26=\begin{bmatrix}2\\7\\8\\13\\0\end{bmatrix}=\begin{bmatrix}C\\H\\I\\N\\A\end{bmatrix}$$

5. 安全性
    1. 定理：对于a，如果相对模数n存在逆元，则需满足`gcd(a,n)=1`.
        1. 为了得到逆元，需要使得 `gcd(a,26)=1`，也就是α与26必须互质，因此α的取值范围为：{1，3，5，7，9，11，15，17，19，21，23，25}只有12个值，此外β取值范围是26，所以对于仿射加密，所有可能只有`12*26=312`种!
    2. 在当前计算机的计算能力下是非常不安全的。
6. 破解
    1. 仿射加密是**线性映射**，所以明文和对应密文出现的**频率是一致**的。
    2. 统计密文中字母的频率，假如统计结果如下：V: 119，W: 67，T: 54….
    3. 与标准字母频率比较，找到两个明文密文映射

| 分类 | 使用频率分类字母集 | 每个字母约占百分数 |
|---|---|---|
| I | 极高使用频率字母集:e | 12% |
| II | 次高使用频率字母集:t,a,o,i,n,s,h,r | 6%~9% |
| III | 中使用频率字母集:d,l | 4% |
| IV | 低使用频率字母集:c, u,m,w,f,g,y,p,b | 1.5%~2.3% |
| V | 次低使用频率字母集:v,k,j,x,q,z | 1% |

    4. 确定映射关系：`e->v; t->w`

5. 由于：E: 4, T: 19，V: 21, W: 22
6. 所以我们可得两个线性方程组：
    1. `21 = 4α +β(mod26)  -> 26m + 21 = 4α +β`
    2. `22 = 19α+β(mod26) ->  26m + 22 = 19α +β`
7. 其中，α 满足0<α<n的正整数，α 要和26互质，因此，α可能取得的值为：{1，3，5，7，9，11，15，17，19，21，23，25}
8. 当m = 1时，方程组解：（7，19）
9. 实际破解时，假定的密钥(a,b)可能不止一组，但是对于当前计算机的计算能力，即使是使用穷举法也很快就能破解仿射加密

4. 维吉尼亚密码
    1. 加密方式：
        1. 列出明文并按照密钥长度分组
        2. 用密钥对每个组内字母进行移位加密

$$C_i \equiv P_i + K_i \pmod{26}$$

$$P_i \equiv C_i - K_i \pmod{26}$$

        3. 加解密公式：
        4. 特点：
            1. 维吉尼亚密码实际上是移位密码的一种扩展
            2. 能够消除字母的频率特征
    2. 破解维吉尼亚密码
        1. 找到密钥长度
        2. 找出密钥
    3. 找出密钥长度后破解密钥就很容易（Why?）
        1. 把密文按照密钥长度L，选出密文中的第1个、第L+1、第2L+1个……字母进行词频统计。
    4. 找到密钥长度的方法
        1. Kasiski实验
            1. 利用英文单词的规律，统计重复间隔。
            2. 如“`YC`”：间隔12。则公约数：1，2，3，4，6，12都可能是密钥长度。
        2. Friedman测试
            1. IC, index of coincidence 重合指数： `IC=P(A)^2 + P(B)^2+……+P(Z)^2 = 0.065`
            2. P(i)由字母频率分析得到
            3. 利用重合指数推测密钥长度的原理在于，对于一个由移位加密的序列，**由于所有字母的位移程度相同，所以密文的重合指数应等于原文语言的重合指数**。
            4. 例：
                1. 密文：`ABCDEABCDEABCDEABC`
                2. 假设密钥长度为1， P = 0.065
                3. 假设密钥长度为2

1. 组1： `ACEBDACEB` （凯撒密码）
2. 组2： `BDACEBDAC` （凯撒密码）
3. 如果组1的重合指数接近0.065，组2的重合指数也接近0.065，那么基本可以断定密钥长度为2。

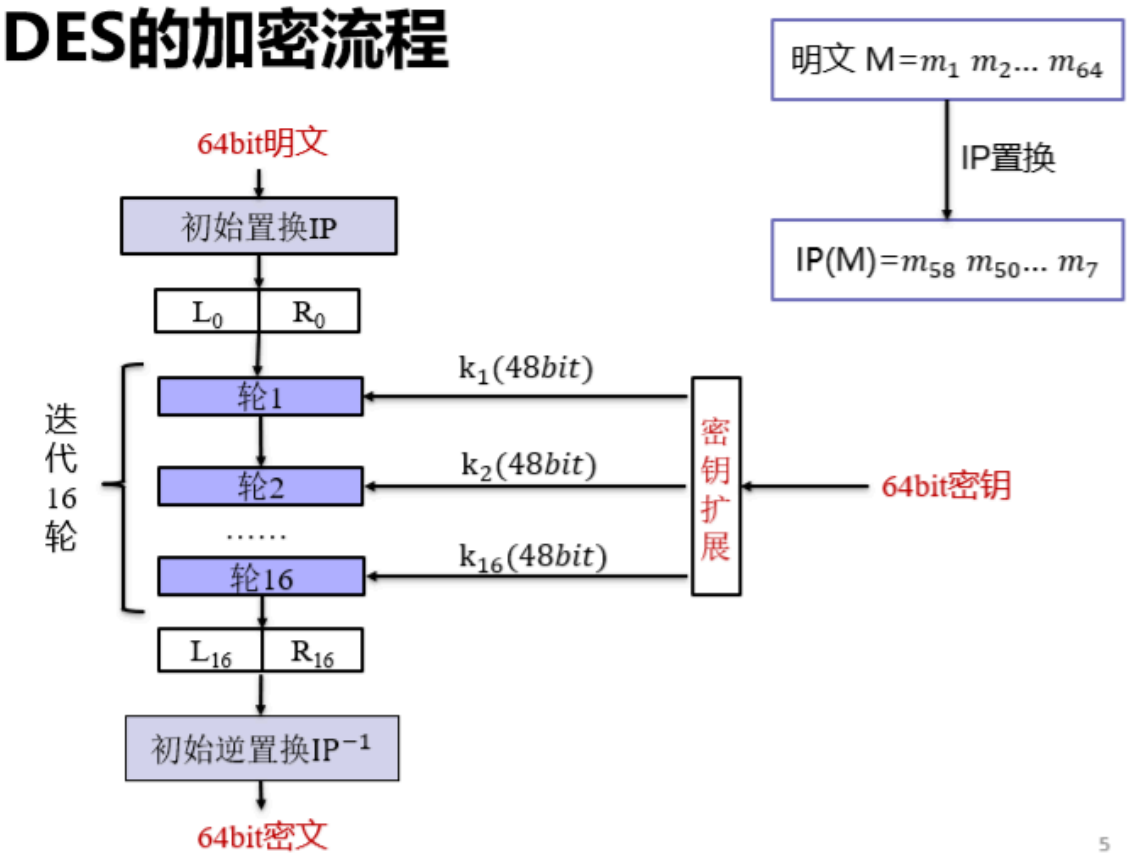
5. 一次性密码本

- 双方共同维护一个本子，使用和明文长度相等的密钥进行加密。每次加密完，换新的密钥。密钥的生成是完全随机的。
- 比如，相当于移位加密中，每个字母的移位都不同
- 一次性密码本是一种**理论上绝对安全**的加密方式
- 适用于二进制加密，流程如下：

$$\begin{array}{ll} (\text{message}) & 00101001 \\ (\text{key}) & \oplus\ 10101100 \\ \hline (\text{ciphertxt}) & 10000101 \end{array}$$

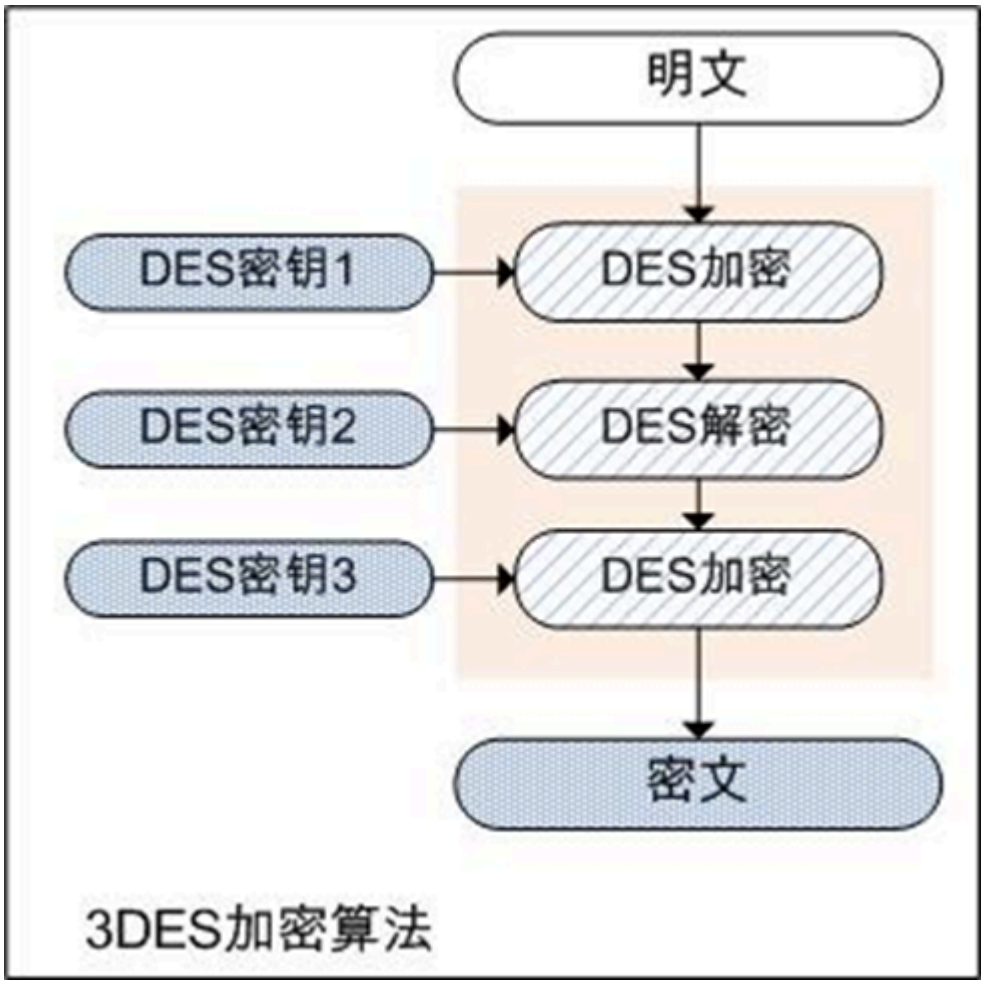
## 7.4 现代密码学-对称密钥

1. DES对称加密
    1. 定义：DES （Data Encryption Standard）是一种使用56位密钥对64位长分组进行加密的密码，是一种迭代算法。DES是第一个公开的分组加密算法。
    2. 特点：
        1. 对称加密算法：同一个密钥用于加密和解密，密钥必须保密。
        2. 块加密：对固定大小（64位）的数据块进行加密，每个数据块独立处理。
        3. Feistel结构：DES由16轮相似的操作组成，每轮使用一个子密钥进行替换和置换操作。
        4. 56位密钥：实际使用的密钥长度为56位，尽管**输入的密钥长度为64位**（其中8位用于校验）。




3. 5

4. DES的安全性
    1. 密钥长度：由于密钥长度较短，容易被暴力破解。1997年，DES的密钥首次被公开破解。
    2. 其它攻击方法：差分密码分析、线性密码分析等方法可以有效攻击DES。
    3. 替代方案：为了增强安全性，Triple DES（3DES）应运而生，使用三个DES操作（加密-解密-加密）来增加密钥长度和复杂性。
5. 3DES：由三个连续的DES加密组成，也称之为三重DES。
    1. 三个密码组件既可以是一个加密函数又可以是一个解密函数。

2. 

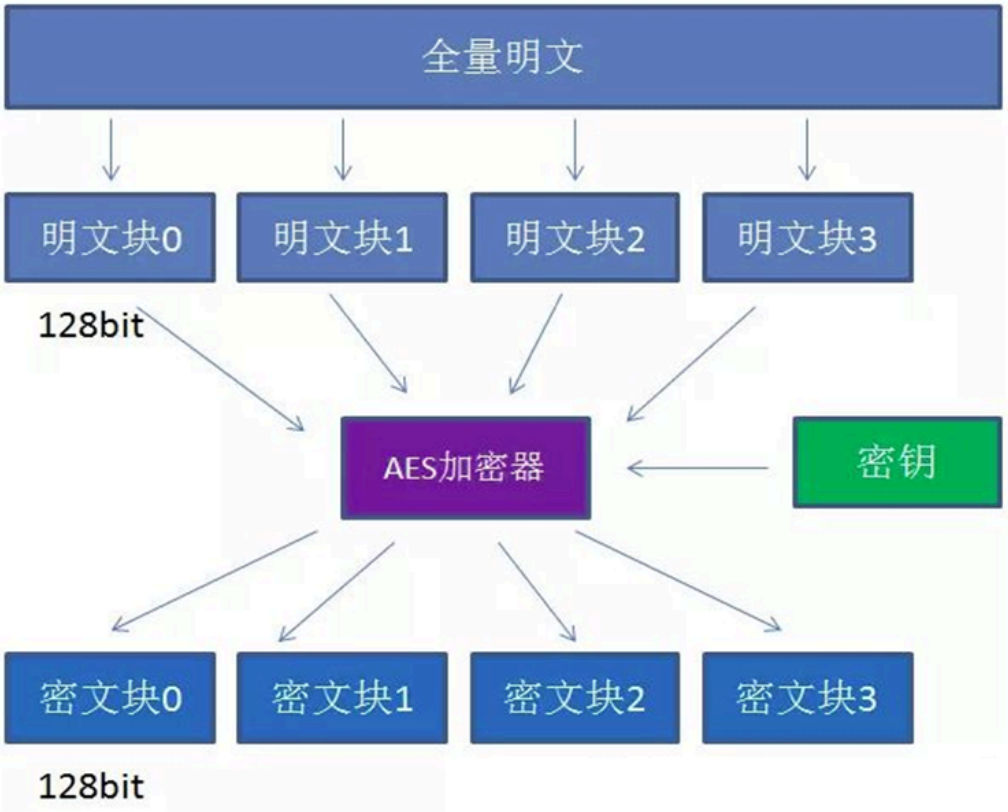
3. 安全性：
    1. n若k_1 、k_2 、 k_3互不相等，密钥长度 `56*3=168` ，每秒尝试10亿个秘钥，暴力破解需要10^31年！
    2. 3DES目前仍有足够的安全性
4. **优点：**
    - **密钥长度增加到112位或168位，克服了DES面临的穷举攻击。**
    - **相对于DES，增强了抗差分分析和线性分析等的能力。**
    - **由于DES已经大规模使用，升级到3DES比更新新算法成本小得多。**
    - **DES比其它任何加密算法受到的分析时间都长的多，相应地，3DES抗分析能力更强。**

    **不足：**
    - **3DES处理速度较慢。**
    - **虽然密钥长度增加了，但明文分组长度没变，与密钥长度的增长不匹配。**

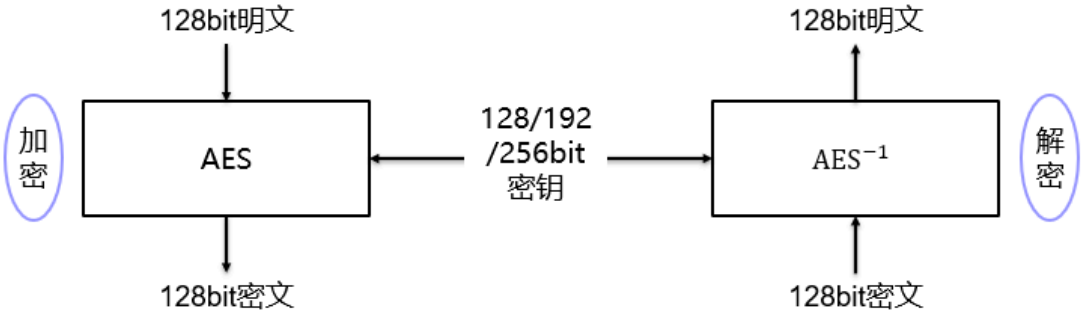
2. AES对称密码
    1. 背景：
        1. DES算法由于其密钥较短，难以抵抗现有的攻击，因此不再作为加密标准。
        2. 1997年1月，美国NIST向全世界密码学界发出征集21世纪高级加密标准（AES—Advanced Encryption Standard）算法的公告，并成立了AES标准工作研究

室，1997年4月15日的例会制定了对AES的评估标准。

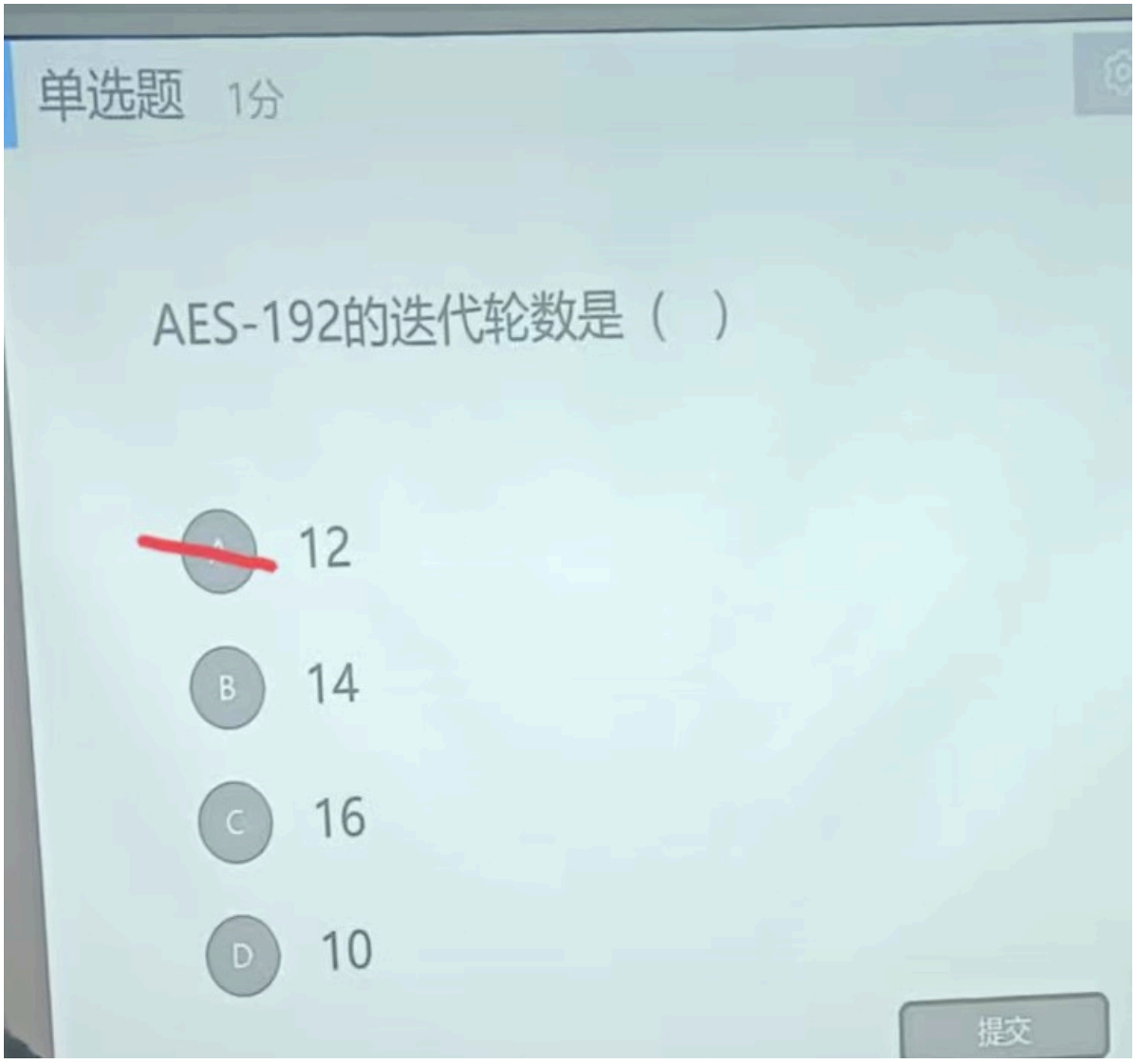
2. AES算法征集的要求：
    1. AES是公开的；
    2. AES为对称密钥分组密码体制；
    3. AES的密钥长度可变，可按需要增大；
    4. AES适于用软件和硬件实现；
    5. AES可以自由地使用，或按符合美国国家标准（ANST）策略的条件使用。
3. 算法衡量条件
    1. 安全性
    2. 计算效率
    3. 内存要求
    4. 使用简便性
    5. 灵活性
4. 简介
    1. 高级加密标准：分组为128位、支持3种密钥长度（128、192、256）且软硬件实现都很高效。是当前应用最广泛的对称算法。



2. 参数




| | AES-128 | AES-192 | AES-256 |
|---|---|---|---|
| 密钥长度 | 128 | 192 | 256 |
| **迭代轮数** | 10 | 12 | 14 |

5. 安全性，已知攻击方法：
    1. 差分密码分析：攻击者通过分析输入和输出差分来推测密钥。
    2. 线性密码分析：利用线性逼近的方法对密钥进行推测。
6. 


3. 对称密钥体质缺陷
    1. 对称密码体制(例如DES, AES) 需要两个用户利用提前共享的秘钥来建立“**安全信道**”，然而通信双方**共享秘钥并不容易**。

2. 考虑一个具有N个用户的团体，如果用户两两之间都需要进行安全通信，采用对称密码体制来保护用户之间的通信: 每个用户需要与其余的N -1个用户共享私钥，整个系统**需要管理N(N -1)/2个密钥**。
3. 无法保证可追溯性（accountability）

# 7.5 公钥密码（非对称密钥）

1. 基本原理：
    1. 公钥与私钥
        1. 公钥可以公开分发，任何人都可以使用公钥加密消息。
        2. 私钥必须保密，仅持有私钥的一方可以解密消息或生成签名。
    2. 加密与解密过程
        1. 加密：使用接收方的公钥加密消息，只有接收方能用其私钥解密。
        2. 解密：接收方用私钥解密加密的消息。
    3. 数字签名与验证
        1. 签名：发送方用自己的私钥对消息进行签名，生成数字签名。
        2. 验证：接收方用发送方的公钥验证签名，确认消息的真实性和完整性。
2. RSA-经典公钥密码算法
    1. 欧拉定理：
        1. 剩余类互素/互质定义：在模m的一个剩余类当中，如果有一个数与m互素，则该剩余类中所有的数均与m互素，这时称该剩余类与m互素。
        2. 欧拉函数定义：与m互素的剩余类的个数称为欧拉函数，记为φ(m)。$\varphi(m)$ 等于Z_m当中与m互素的数的个数。对于任意一个素数m，φ(m)=m-1。
        3. 欧拉定理：设m是正整数，rϵZ_m, 若 `gcd(r,m)=1` ，则r^(φ(m))=1(mod m).
            1. Z_m：对于m来说，0,1,2,…,m-1即为最小非负完全剩余系。
            2. `gcd(r,m)=1` ：r,m的最大公约数为1，即两者互素。
            3. r^(φ(m))=1(mod m)：即r^(φ(m))除以m的余数为1.

2. 算法

**Key Generation**

| | |
|---|---|
| Select $p, q$ | $p$ and $q$ both prime |
| Calculate $n$ | $n = p \times q$ |
| Select integer $d$ | $gcd(\phi(n), d) = 1; 1 < d < \phi(n)$ |
| Calculate $e$ | $e = d^{-1} \bmod \phi(n)$ |
| Public Key | $KU = \{e, n\}$ |
| Private Key | $KR = \{d, n\}$ |

**Encryption**

Plaintext: $M < n$

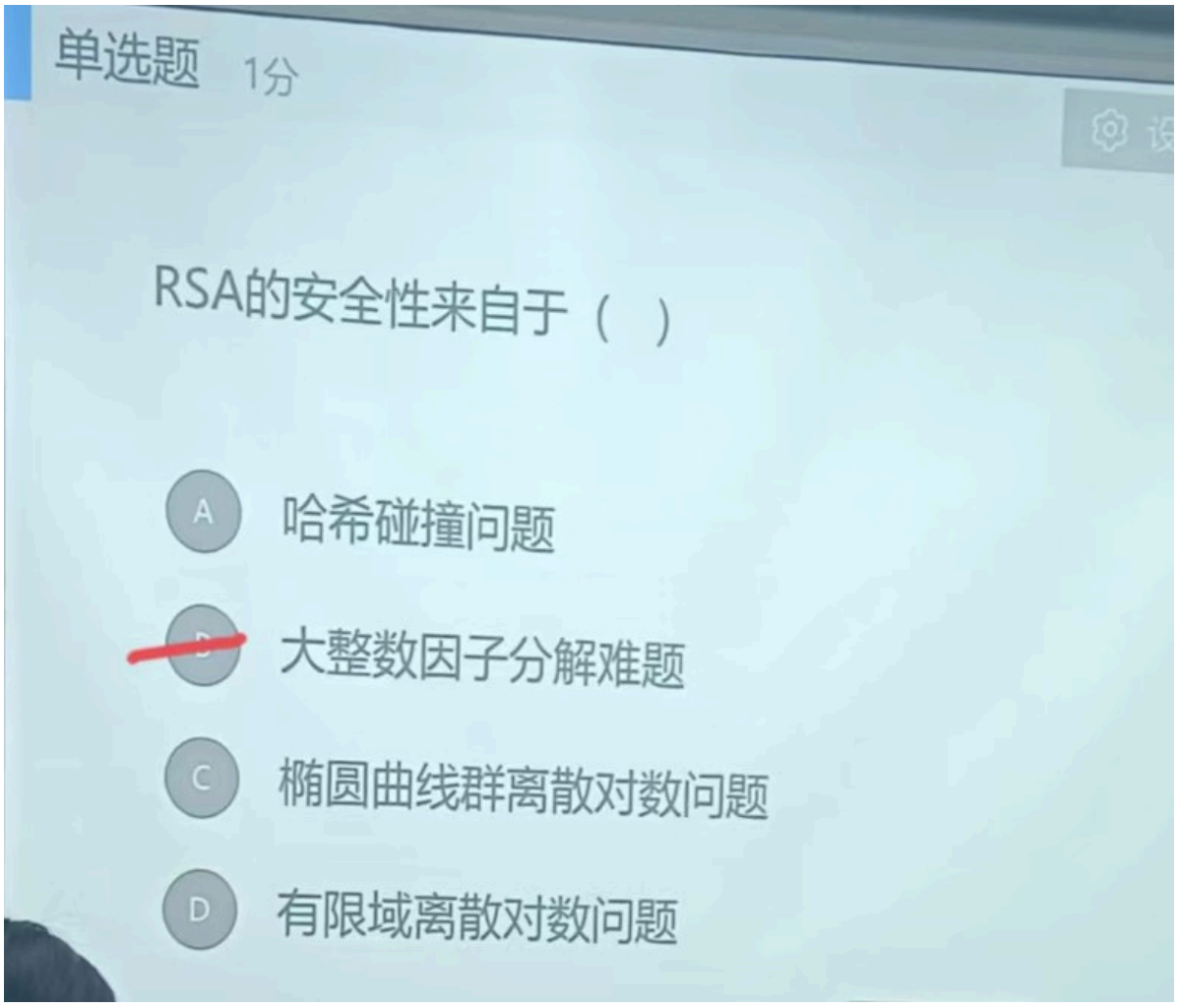
Ciphertext: $C = M^e \pmod n$

**Decryption**

Ciphertext: C

Plaintext: $M = C^d \pmod n$

3. 安全性：
    1. 对于攻击者来说，已知e、n，如果无法求出d就是安全的。
    2. 问题本质：已知n，求n=p * q，即数的素分解问题。一般情况下，破解RSA密钥成为计算上的不可解问题。
4. 优点：密钥空间大
5. 缺点：
    1. 产生密钥很麻烦，受到素数产生技术的影响，因此很难做到一次一密。
    2. 分组长度太长。
    3. 速度太慢，RSA最快的情况也比DES慢100倍。

6. 


3. ECC（椭圆曲线密码学）
    1. 发明者：Victor Miller和Neal Koblitz
    2. 基本原理：基于椭圆曲线离散对数问题
    3. 安全性：提供相同安全性的前提下，使用较短的密钥长度
        1. 例如：ECC的160位密钥提供的安全性相当于RSA的1024位密钥。
    4. 过程：密钥生成、加密、解密、签名、验证
    5. 应用：移动设备、智能卡、SSL/TLS等领域
4. 哈希函数
    1. 哈希函数是一种将任意长度的输入数据转换为固定长度的输出数据的函数。
    2. 哈希值通常用于数据完整性校验、数字签名、密码存储等领域。
    3. 常见的哈希函数
        1. MD5 (Message Digest Algorithm 5)
            1. 输出长度128位（16字节）；
            2. 计算速度快，但已被证明不够安全，容易受到碰撞攻击。
        2. SHA (Secure Hash Algorithm)家族算法
            1. SHA-1：
                1. 输出长度160位（20字节）；
                2. 曾广泛用于数字签名、证书验证，但已被证明存在安全漏洞，容易受到碰撞攻击。

3. 案例：

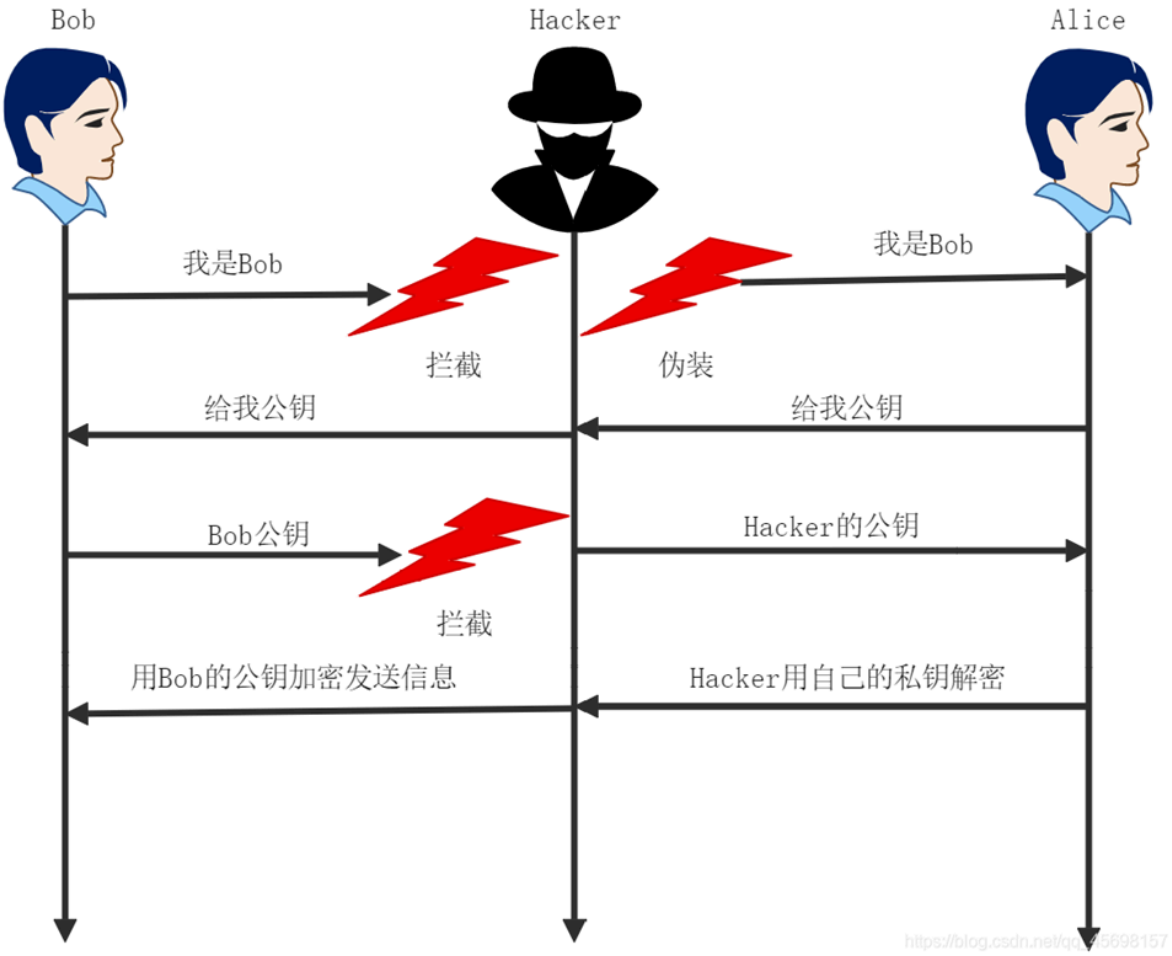
```
In [1]: import hashlib

        # 两段HEX字节串，注意它们有细微差别
        a = bytearray.fromhex("0e306561559aa787d00bc6f70bbdfe3404cf03659e704f8534c00ffb659c4c8740c

        b = bytearray.fromhex("0e306561559aa787d00bc6f70bbdfe3404cf03659e744f8534c00ffb659c4c8740c

        # 输出MD5，它们的结果一致
        print(hashlib.md5(a).hexdigest())
        print(hashlib.md5(b).hexdigest())

cee9a457e790cf20d4bdaa6d69f01e41
cee9a457e790cf20d4bdaa6d69f01e41
```

2. SHA-2：
    1. 包含SHA-224、SHA-256、SHA-384和SHA-512等变种。
    2. 输出长度：224位、256位、384位、512位
    3. 特点：安全性较高，目前广泛应用于各类安全应用。
    4. 使用场景：数字签名、SSL/TLS协议、证书验证、数据完整性校验。
3. SHA-3: SHA-3算法正式发布于2015年

4. 哈希函数需满足的特性：
    1. 输入长度可变：对任何长度的输入x都适用
    2. 输出长度固定：对任何长度的输入x，输出z都是固定长度
    3. 效率：计算复杂度低
    4. 抗原像攻击（单向性）：对于给定输出y，不可能找到对应的输入x
    5. 抗第二原像攻击（抗弱碰撞性）：对于给定的x_1，找到满足h(x_1)=h(x_2)的x_2是不可能的
    6. 抗碰撞攻击（抗强碰撞性）：找到满足h(x_1)=h(x_2 )的偶对x_1=x_2是不可行的。
5. 应用：
    1. 数据完整性检验
    2. 用于数字签名
    3. 密钥推导
    4. 密码存储

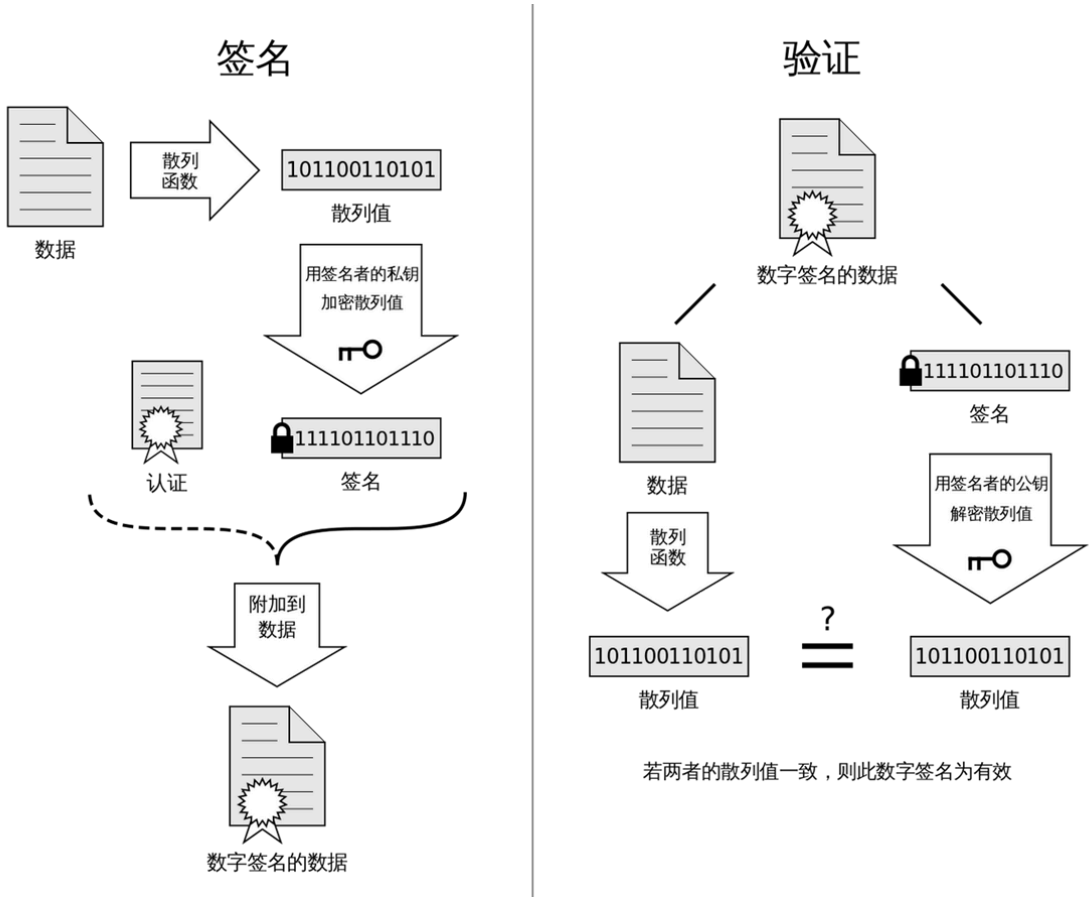
5. 数字签名

1. 为什么




2. **数字签名**是一种通过身份验证和不可否认性来提供数据真实性和完整性的手段。
    1. 发送消息时，附加一个数字签名，如果消息被篡改，数字签名就会变得非法。
3. 数字签名可以**通过非对称加密（用私钥加密）和哈希实现**。
    1. 用私钥对消息摘要进行加密，附加到消息摘要后面。
    2. 接受者收到后对消息进行哈希得到一个**消息摘要1**，用公钥对加密后的消息摘要进行解密得到**消息摘要2**，如果一致则保证了消息的完整性。

4. 数字签名的过程

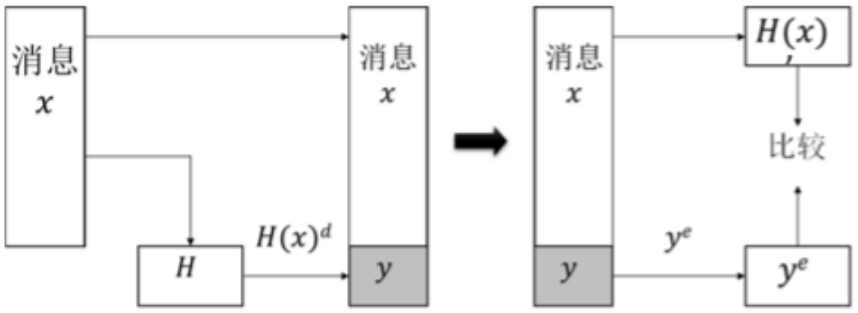



5. 数字签名实现方案分类
    1. 利用特殊的公钥加密算法实现。
        1. RSA数字签名算法
            1. RSA数字签名算法是一种基于RSA公钥加密算法的数字签名方法。它不仅可以用于加密，还可以用于生成和验证数字签名
            2. 安全性：基于大整数因子分解问题，具有很高的安全性。
            3. 应用场景
                1. SSL/TLS证书：用于保护互联网通信的安全，确保网站身份的真实性
                2. 电子邮件签名：通过数字签名验证电子邮件的发送者身份和邮件内容的完整性。
                3. 软件签名：验证软件分发和更新的真实性和完整性，防止恶意软件的传播。
                4. 数字证书：在公共密钥基础设施（PKI）中，用于数字证书的签名和验证。

4. 实现过程

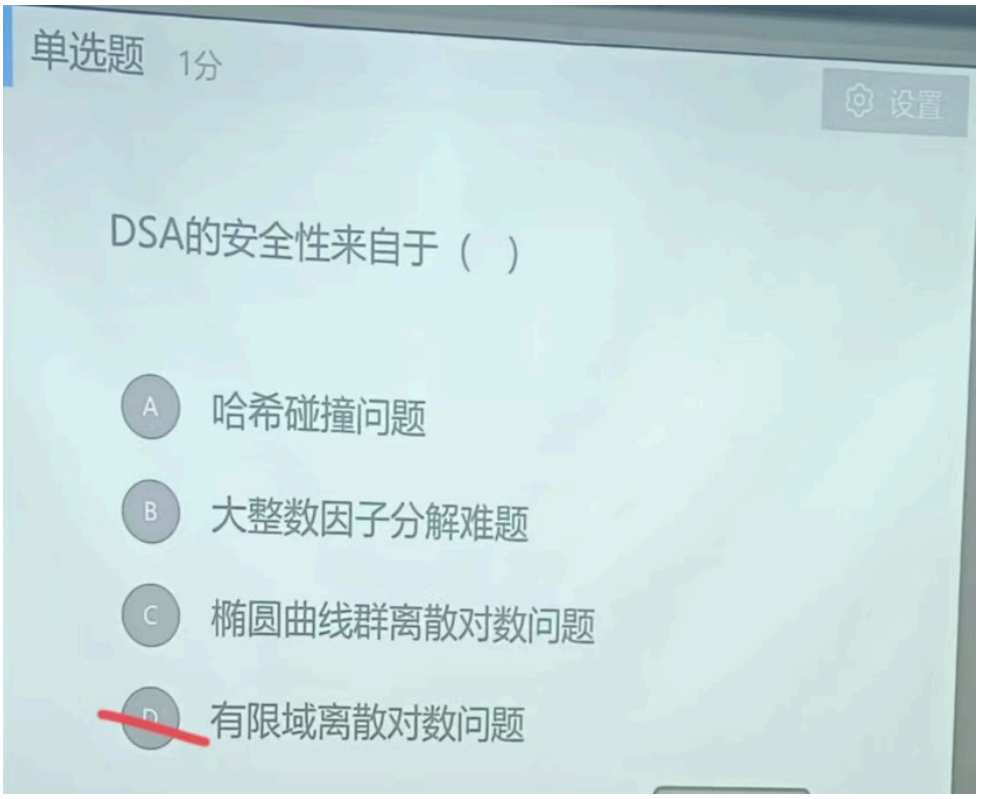
$$签名：Sig = y = H(x)^d mod\ n$$

$$验证：H'(x) \overset{?}{=} H(x) = y^e mod\ n$$




2. ECDSA（Elliptic Curve Digital Signature Algorithm）
    1. 一种基于椭圆曲线密码学的数字签名方法。
    2. 安全性：基于椭圆曲线离散对数问题的难解性，提供与RSA相同级别的安全性，但使用的密钥长度更短。
    3. 应用场景
        1. SSL/TLS证书：用于保护互联网通信的安全，确保网站身份的真实性。
        2. 移动支付：在移动支付系统中，ECC的高效性和较小的密钥长度使其成为理想选择。
        3. 区块链和加密货币：许多区块链系统（如比特币和以太坊）使用ECC进行交易签名，以确保交易的真实性和完整性。
        4. 物联网设备：在资源受限的物联网设备上，ECC的高效性和低资源消耗非常适合。

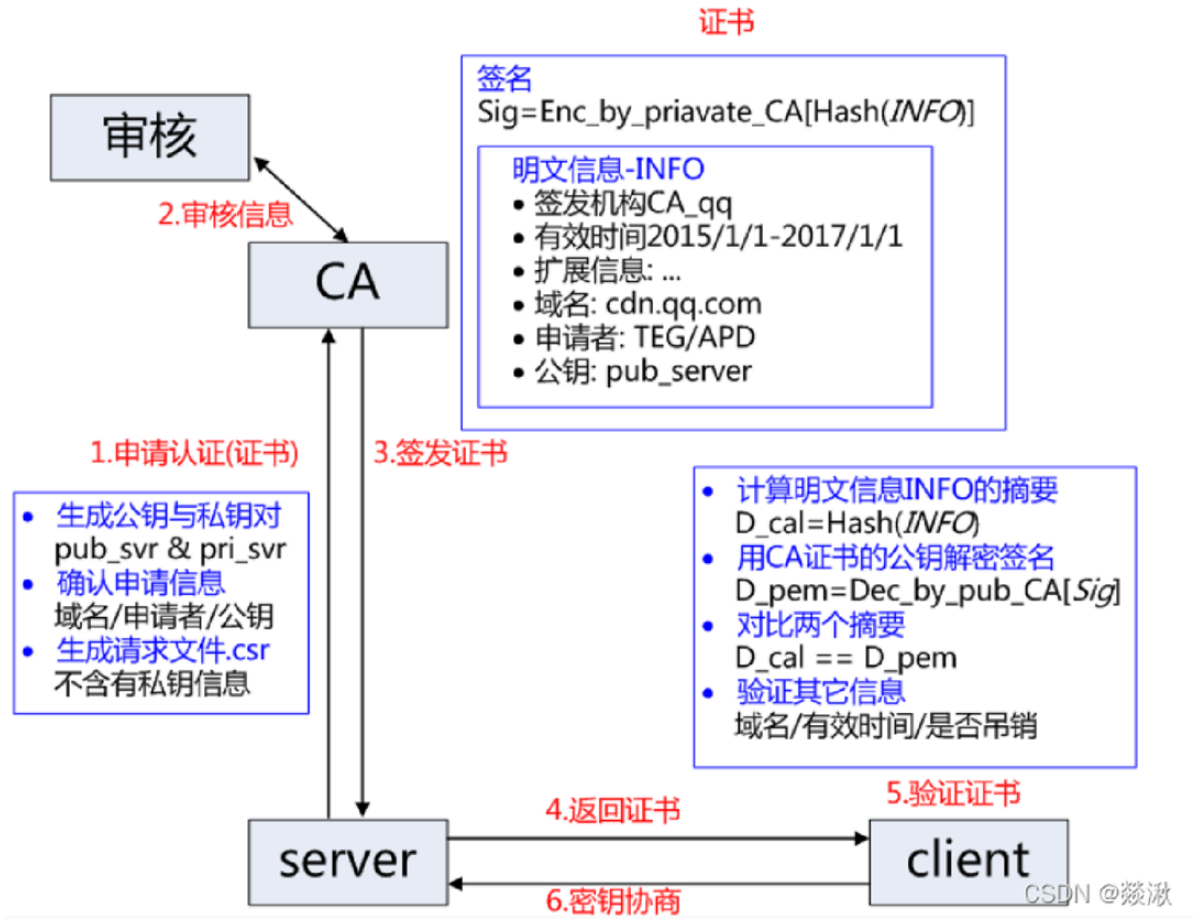
2. 利用专门设计的数字签名算法实现。
    1. DSA数字签名算法（Digital Signature Algorithm ）
        1. 是由美国国家安全局（NSA）为美国国家标准与技术研究院（NIST）设计的一种数字签名标准。它仅用于签名，不用于加密。
        2. 安全性：基于离散对数问题的难解性，具有很高的安全性。
        3. 应用场景
            1. 软件分发：确保软件安装包的完整性和真实性。
            2. 数字证书：用于认证数字证书的签名，确保证书的可信性。
            3. 电子文件签名：对电子文档进行签名，防止篡改和伪造。

4. 


3. 比较

| 特性 | RSA | ECDSA | DSA |
| --- | --- | --- | --- |
| 基本概念 | 基于RSA公钥加密算法的数字签名方法 | 基于椭圆曲线密码学（ECC）的数字签名方法 | 基于离散对数问题的数字签名算法 |
| 安全性 | 基于大整数因子分解问题，安全性较高 | 基于椭圆曲线离散对数问题，安全性高 | 基于离散对数问题，安全性较高 |
| 密钥长度 | 2048位及以上 | 256位及以上，提供同等安全性时密钥更短 | 2048位及以上 |
| 签名长度 | 较长（与密钥长度成正比） | 较短（相对于同等安全性的RSA） | 中等（与密钥长度成正比） |
| 签名生成速度 | 较慢 | 较快 | 较快 |
| 签名验证速度 | 较慢 | 较快 | 较慢 |
| 计算复杂度 | 较高 | 较低 | 中等 |
| 性能 | 对计算资源需求较高 | 高效，适合资源受限环境 | 依赖高质量随机数，速度适中 |
| 应用场景 | SSL/TLS证书、电子邮件签名、软件签名 | 移动支付、区块链、物联网设备 | 软件分发、数字证书、电子文件签名 |

6. 公钥基础设施
    1. 管理非对称密钥颁发的常用方法是基于**公钥基础设施**（Public Key Infrastruction, PKI）机制的，它是由一个协议、数据格式、规则和实施组成的系统。
    2. 这个系统用来把公钥与对应私钥所有者对应起来（**公钥身份识别**）。
        1. PKI依赖于使用数字证书。
        2. 数字证书是带数字签名的数据结构，它与公钥一起来验证证书所有者身份以及相关信息，例如有效期。
        3. 数字证书通常由第三方证书颁发机构（Certificate Authority, CA）数字签发的。

3. 

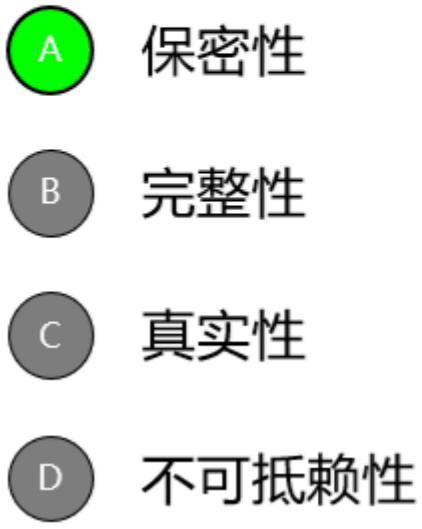

7. 身份与访问管理
    1. **身份与访问管理**（Identity and Access Management, IAM）包括控制和追踪用户身份以及IT资源、环境、系统访问特权的必要组件和策略。
    2. IAM机制由**四个主要部分**组成：
        1. 认证：如用户名和密码的组合，还支持数字签名、数字证书、生物特征识别硬件、把用户账号和注册IP或MAC地址进行绑定等。
        2. 授权：授权组件定义正确的访问控制粒度，监管身份、访问控制权力和IT资源可用性之间的关系。
        3. 用户管理：负责创建新的用户身份和访问组，重设密码、定义密码策略和管理特权。
        4. 证书管理：建立对已定义的用户账号的身份和访问控制的规则。
8. **单点登录**(Single Sign-On, SSO)是为了解决跨越多个云服务为云服务用户传播认证和授权，特别是在需要大量的云服务或IT资源时的困难。
    1. 单点登录机制使得一个云服务用户能够**被一个安全代理认证**。
    2. 这个安全代理建立起一个**安全上下文**，当云用户访问其他云服务或IT资源时，这个上下文被**持久化**。
    3. 否则，云用户需要在后续的每个请求中认证自己。
    4. 这个机制**不直接抵抗任何云威胁**，主要增强基于云的环境的访问并管理分布式IT资源。
    5. 优点：
        1. 简化用户体验
        2. 提高生产力

3. 统一管理

9. 基于云的安全组
    1. **基于云的安全组**(cloud-based security group)是不同用户和组创建各自的物理和逻辑IT环境。
        1. 如组A部署防火墙，用于外部因特网访问；组B是内部网，无防火墙，不能访问因特网。
    2. 通过给虚拟机分配各种不同的物理IT资源，**物理资源分割**可以保护不同组的物理机器。
    3. 网络分割成**逻辑的基于云的安全组**，形成逻辑网络边界。
        1. 逻辑的基于云的安全组会有一些特殊的规则，控制安全组之间的通信。
    4. 能够帮助对抗拒绝服务、授权不足和信任边界重叠等威胁。
10. 强化的虚拟服务器映像
    1. **强化**是将不必要的软件从系统中剥离出来，避免了潜在的漏洞。
        1. 如关闭不必要的端口，不使用的服务、内部根账户和宾客访问。
    2. 强化的服务器映像能够帮助对抗拒绝服务、授权不足和信任边界重叠等威胁。

# 7.6 课后题

1. A发送给B的消息被某个第三方窃听了，这属于破坏了安全通信的（ ）



B接收到A的一条消息，但发现这条消息生成的数字签名与A发送的数字签名不一致，这种对数字签名的比对操作属于保证通信数据的（　）

A 保密性

B 完整性

C 真实性

D 不可抵赖性

2.

对称加密主要用来保证数据的（　）

A 保密性

B 完整性

C 真实性

D 不可抵赖性

3.

私钥加密主要用来保证数据的（　）

A 保密性

B 完整性

C 真实性

D 不可抵赖性

4.

公钥加密主要用来保证数据的（　）

A 保密性

B 完整性

C 真实性

D 不可抵赖性

5.

数字签名主要用来保证数据的（　）

A 保密性

B 完整性

C 真实性

D 不可抵赖性

6.

数字签名可以通过（　）实现。

A 私钥加密

B 公钥加密

C 对称加密

D 哈希

7.

8. 以下在云安全中非常重要，但却不直接抵抗任何云威胁的是（ ）

A 身份与访问管理

B 私钥加密

C 单点登录

D 基于云的安全组

9.