测试点 2-1

(1)密码学是研究编制密码和破译密码的技术科学。研究工作方向：研究密码变化的客观规律，应用于编制密码以保守通信秘密的，称为编码学；应用于破译密码以获取通信情报的，称为破译学，总称密码学。

(2)古代加密方法：源自对自然的直接感受、加密思想简单，直观、技巧性强、安全性依赖算法本身保密性、依赖于实物，不便传输、仍有一定安全性。

古典密码：基于变换的思想（能有针对性地设计算法、安全性依赖变换参数（密钥）的保密）、安全性有所提高（可以跟换变换参数（密钥）、比古代加密方法安全性高）。

近代密码：数学开始主导密码学、对密码安全有了更深刻的理解。

现代密码：基于计算机科学的发展（不再依赖技巧，而以数学理论为基础、安全性完全依赖于严密的数学证明、允许引入大量复杂运算、破解代价极高，且极度依赖数学的发展）。

(3)明文及密文空间都是 26（字母表大小）；加密方法：明文中的所有字母都在字母表上向后（或向前）按照一个固定数目进行偏移后被替换成密文；解密方法：将密文中所有字母都在字母表上向后（或向前）按照加密的固定数目进行偏移；密钥空间：25.

(4)对称加密：n(n-1)个，公钥加密：2n 个。

(5)
```
#include <stdio.h>
#include <string.h>

int main()
{
    char passwd[100],encrypted[100];
    int i,j,k,t,move;
    while(1)
    {
        printf("Enter message to be encrypted:");
        gets(passwd);
        printf("Enter shift amount(1-25):");
        scanf("%d%*c",&move);
        for(i=0; i<strlen(passwd); i++)
        {
            if(passwd[i] >= 'A' && passwd[i] <= 'Z')
            {
                passwd[i] = ((passwd[i]-'A')+move)%26+'A';
            }
            else if(passwd[i] >= 'a' && passwd[i] <= 'z')
            {
                passwd[i] = ((passwd[i]-'a')+move)%26+'a';
            }
        }
        printf("%s",passwd);
        printf("\n");
    }
    return 0;
}
```

测试点 2-2

(1)物理安全是保护计算机设备、设施（网络及通信线路）免遭地震、恐袭、火灾等环境事故和人为操作失误或错误及各种计算机犯罪行为破坏的措施和过程。解决途径：提供防护措施、提高可靠性、隔绝危险。

(2)防止设备被盗或被毁。在机房安排执勤人员 24 小时监控。

(3)通常可以通过增加冗余（备份）系统的方式来实现，这也是物理安全中常见的技术手段。

容错：应对系统中普遍存在的一般性故障（包括设备故障，人为操作失误等）采取的技术手段，如双机热备系统，分布式处理系统。

容灾：针对突发性灾难事故（包括地震、飓风等严重自然灾害，恐怖袭击等人为灾难）的预防和恢复采取的技术手段，如建立异地的灾难备份中心。

强调“异地”是为了防止备份一起损坏在此次灾难中。

(4)增加系统容量：随着业务量越来越大，而要能够应对越来越大的业务量时，一台机器的性能已经无法满足业务的需求了，我们需要多台机器才能应对大规模的应用场景。所以，我们需要垂直或水平拆分业务系统，让其变成一个分布式的架构。

加强系统可用性：由于业务变得越来越关键，需要提高整个系统架构的可用性，这就意味着架构中不能存在单点故障。这样整个系统不会因为一台机器出现故障导致整体不可用。所以需要通过分布式架构来冗余系统以消除单点故障，从而提高系统的可用性。

C(T,N)(1-p)^T+C(T+1,N)(1-p)^(T+1)+…+(1-p)^N

测试点 2-3

(1)静态认证：登录密码、动态认证：短信验证码、生物特征认证：Touch ID、多因子认证：银行 U 盾、认证协议。

(2)根据认证条件（凭据）的数目、根据认证条件（凭据）的状态。手机网银属于后者，需要提供银行预留手机号、登录密码、验证码。

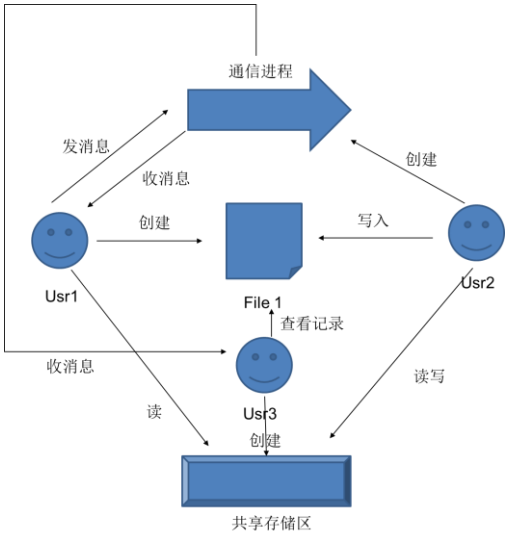
(3)不正确。因为即使是采用了 3D 建模方式进行的验证也有可能被人脸倒模硅胶面具所骗过。

测试点 2-4

(1)ACL：优点：权限回收容易；缺点：权限传递困难；ACCL：如用于各个 LAN 间的接口，过滤 LAN 流量。

ACCL：优点：权限传递简单；缺点：权限回收容易；如分布式系统。

(2)



(3)不能。RBAC 的一种实现模型：
定义：RBAC0 模型由以下描述确定：
U、R、P、S 分别表示用户集合、角色集合、许可权集合和会话集合。
PA P×R 表示许可权与角色之间多对多的指派关系。
UA U×R 表示用户与角色之间多对多的指派关系。
用户：S→U 每个会话 si 到单个用户 user(si)的映射函数（常量代表会话的声明周期）。
角色：S→R 每个会话 si 到角色子集 roles(si) {r|user(si, r')∈UA}（能随时间改变）的映射函数，会话 si 有许可权 Ur∈roles(si){p|(p,r')∈PA}。
在使用 RBAC0 模型时，应该要求每个许可权和每个用户至少应该被分配给一个角色。两个角色被分配的许可权完全一样是可能的，但仍是两个完全独立的角色，用户也有类似情况。角色可以适当的被看做是一种语义结构，是访问控制策略形式化的基础。
RBAC0 把许可权处理为非解释符号，因为其精确含义只能由实现确定且与系统有关。RBAC0 中的许可权只能应用于数据和资源类客体，但不能应用于模型本身的组件。修改集合 U、R、P 和关系 PA 和 UA 的权限称为管理权限，后面将介绍 RBAC 的管理模型。因此，在 RBAC0 中假定只有安全管理员才能修改这些组件。
会话是由单个用户控制的，在模型中，用户可以创建会话，并有选择的激活用户角色的某些子集。在一个会话中的角色的激活是由用户来决断的，会话的终止也是由用户初始化的。RBAC0 不允许由一个会话去创建另一个会话，会话只能由用户创建。
(4)是 ACL。因为在文件的安全属性一栏，是以客体为中心建立访问权限表的。