eit\_ex14\_AES  
1. 解释对称密码体制。

答：（1）在对称密码系统中，数据发信方将明文和加密[密钥](http://baike.baidu.com/view/934.htm" \t "http://baike.baidu.com/_blank)一起经过特殊加密算法处理后，使其变成复杂的加密密文发送出去。收信方收到密文后，若想解读原文，则需要使用加密用过的同一把[密钥](http://baike.baidu.com/view/934.htm" \t "http://baike.baidu.com/_blank)及相同算法的逆算法对密文进行解密，才能使其恢复成可读明文。

（2）对称密码系统的特点是算法公开、计算量小、加密速度快、加密效率高。

（3）不足之处是，交易双方都使用同样钥匙，安全性得不到保证。此外，每对用户每次使用[对称加密算法](http://baike.baidu.com/view/7591.htm" \t "http://baike.baidu.com/_blank)时，都需要使用其他人不知道的惟一钥匙，这会使得发收信双方所拥有的密钥数量成几何级数增长，[密钥管理](http://baike.baidu.com/view/297229.htm" \t "http://baike.baidu.com/_blank)成为用户的负担。

（4）算法公开，密钥不公开；

2. 以 AES 为例，解释对称密码体制的设计思想。

答：大多数AES计算是在一个特别的有限域完成的。

AES加密过程是在一个4×4的字节矩阵上运作，这个矩阵又称为“体（state）”，其初值就是一个明文区块（矩阵中一个元素大小就是明文区块中的一个Byte）。（Rijndael加密法因支持更大的区块，其矩阵行数可视情况增加）加密时，各轮AES加密循环（除最后一轮外）均包含4个步骤：

AddRoundKey — 矩阵中的每一个字节都与该次回合金钥（round key）做XOR运算；每个子密钥由密钥生成方案产生。

SubBytes — 通过一个非线性的替换函数，用查找表的方式把每个字节替换成对应的字节。 ShiftRows — 将矩阵中的每个横列进行循环式移位。

MixColumns — 为了充分混合矩阵中各个直行的操作。这个步骤使用线性转换来混合每列的四个字节。

最后一个加密循环中省略MixColumns步骤，而以另一个AddRoundKey取代。

在对称密码系统中，数据发信方将明文和加密[密钥](http://baike.baidu.com/view/934.htm" \t "http://baike.baidu.com/_blank)一起经过特殊加密算法处理后，使其变成复杂的加密密文发送出去。收信方收到密文后，若想解读原文，则需要使用加密用过的同一把[密钥](http://baike.baidu.com/view/934.htm" \t "http://baike.baidu.com/_blank)及相同算法的逆算法对密文进行解密，才能使其恢复成可读明文。

3. 从中间状态数据与轮密钥的叠加，能否将 AES 解释为完全保密系统?

答：密码系统称为完全保密，是指对一切,,有 即

而从中间状态数据与轮密钥的叠加, AES算法中的加密、解密过程要经过多次数据变换操作，每一次变换操作会产生一个中间结果，称为状态(State)，算法的执行过程如下：

⑴给定一个明文M，将State初始化为M，并进行AddRoundKey操作，将轮密钥与State异或。

⑵对前Nr-1轮中的每一轮，用S盒进行一次SubBytes代替变换，对State做一次ShiftRows行移位操作，再对State做一次MixColumns列混淆操作，然后进行AddRoundKey操作。

⑶按照顺序分别进行SubBytes、ShiftRows、AddRoundKey操作。

⑷将最后的State中的内容定义为密文C。

AES的解密算法于加密不同，基本运算中除轮密钥加(AddRoundKey)不变之外，其余操作如SubBytes、ShiftRows、MixColumns都要求进行求逆变换。

从SubBytes代替变换开始会改变概率分布，所以从中间状态数据与轮密钥的叠加，能将 AES 解释为完全保密系统。

4. 字节替换是否改变熵?

答：字节的替换会影响p（m），从而可能改变熵。

5. 行移位是否改变熵?

答：，行移不会影响概率分布p（m），从而不改变熵。

6. 列混合是否改变熵?

答：，列混合不会影响概率分布p（m），从而不改变熵。