写在前面的一些话

对于资料

抄袭造成的后果（无论是学习上的，还是道德品质上的），一率和本人无关。

个人认为工大密码学的作业中有的题没有参考资料是很难完成和理解的，尤其是对于初学者来说，因此贡献出了此资料。

这个资料最好的利用方法就是写作业没思路了看看，然后立马合上资料自己去思考写完这个题并且之后都不再翻看这个资料。这样就会感叹：资料上的这个思路太一般了一般我有更好的解法，或者是这个解法错了我要改正它。

我**非常非常**不希望这个写的很一般的作业被用来抄袭减少你自己的思考，因为一方面自己实力有限，做的很多题都有更好的解法等着去发现，或者很可能写的时候有漏洞导致就是错的，另一方面这门课不自己思考查资料的话是没法学下去的，我很希望大家多思考多找资料来学习。

作业里有错的部分和参考资料：

目前我觉得自己做的作业五的第题的证明方法有问题，输入之后敌手因该自己产生一个新的随机串用来加密，当b为0时用输入的串，加密1时用自己产生的的随机串，自己的做法是错误的。别的大家来发现吧。

参考资料：

1. 必不可少的就是课本英文第二版的习题解答，如果有时间的话可以看看课本的英文第二版，它改正了第一版的一些错误（中文版就是第一版翻译过来的）。

2. 接着就是你直接可以谷歌出来的别的国外大学的作业题讲解例如：https://danielslamanig.info/lectures/ModernCrypto19Homework11_solutions.pdf，和Stack Overflow上的讨论。

3. **张宇老师推荐的Coursera的Dan的ppt和视频**，b站有大佬加了中文字幕的视频。

HomeWork1

Problem 1：

1. 对于移位密码来说，我认为只需要知道**一个**任意的字母与其对应的密文即可得到对应的密钥。因此选取的明文串长度为 1。假设选取的字母为 m 对应的密文为 c 选取的密钥为 k。

由于$c = Enc_k(m) = (m + k) mod 26$ 因此我们可以很容易得到$k = (c - m) mod 26$ 。从而得到密钥 k。

2. 对于单字母替换密码来说其将字母表中的 26 个英文字母映射到密文中的 26 个英文字母，我们要想知道全部的 26 个字母的映射关系那么密文-明文对中明文串至少应该含有 25 个不同的英文字母（剩下的那个明文字母与剩下的密文字母相对应）。这样才能确定对应的映射关系，而这样的话我们的明文串可以是从 25 开始的任意长度的字符串（但是必须含有 25 个不同的英文字母）。

解密的话我们只要中找到明文串中的每个字母对应的密文串的字母即可得到密钥 K。

3. Vigenère 密码通过重复密码字得到长度与明文串长度相同的字符串，让每个明文字符“加”对应字符串的字符得到对应的密文。因此对于 Vigenère 密码来说解密可以分为两种情况：

如果我们事先知道其密码字长度 x 的话，我们只需要知道长为 x 的明文-密文对即可破解。例如我们假设明文串为 M 对应的密码字为 K 加密后的密文串为 C，它们对应的第 n 个元素用下标 n 表示（n<=x）。那么有：

$$(M_n + K_n) mod 26 = C_n \quad \Rightarrow \quad (C_n - M_n) mod 26 = K_n$$

则利用与移位密码相同的方法可以解出密码字的内容即可得到密钥 K。

如果我们事先并未知道其密码字长度 x 的话，我们所需要的明文长度是不确定的。例如我们想要得到密钥，先假设从明文长度为 1 开始不断尝试利用上面的方法通过明文密文对来获得密码字。我们会发现即使我们得到了形式如 XX 的密码字（X 为某一字符串），我们也没法确定字符串 X 就是最终的密钥，还需要再次进行实验。因为我们没法保证密钥的形式不为 XXY（Y 为与 X 不同的字符串）。

因此我认为我们进行实验时只能用足够长的明文进行测试后得到某个字符串 X 多次重复组成的密码字，接着假设字符串 X 就是最终的密钥（因为此时再加入与 X 不同的字符串

其加密效率较低而且密钥较长难以传输）。而这个较长的明文长度我们无法确定。

Problem 2：

这个问题与问题 1 类似。除了在这个问题中我们可以选择明文串的内容与长度来得到加密后的密文因此比较简便。

1. 对于移位密码来说其需要的长度与问题 1 相同即 1 个字母，只不过我们可以使明文为字母 Z 来简化计算。与问题 1 类似，其结果为$k=(c-m)mod26$，只不过这里 m=Z。

2. 对于字母替换密码来说根据我们问题 1 中的论述可以构造一个最简单的含有 25 个英文字母的字符串$ABCDEFGHIJKLMNOPQRSTUVWXY$来进行推断，根据得到的密文以及问题 1 第 2 问的方法我们就可以得到密钥 K 了。

3. 对于 Vigenère 密码来说我们只能通过选定明文内容来简化计算，但是需要的明文长度无法改变。

如果我们已知其密钥长度 x 那么与第 1 问类似我们可以构造一个字符串长度为 x 其内容全为字母 Z，用问题 1 中的以下公式，我们可以很容易的得到密钥 K。

$$(M_n+K_n)mod26=C_n \quad \Rightarrow \quad (C_n-M_n)mod26=K_n$$

如果我们未知其密钥长度，我们仍然无法知道需要多长的明文。但就像问题 1 中的表达一样我们可以利用全为字母 Z 的明文串来简化计算。我们需要构造足够长的明文，例如我们如果构造以下明文得到密文：

$$ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ$$
$$ABCABCDABCABCDABCABCDABCABCDABCABCDAB$$

那么我们可以相信 ABCABCD 就是其密钥，因为如果不是的话加密效率会过低且密钥太难传送。（当然这种假设是可能出错的，只有不断地实验才能检验其正确性）

Problem 3：

这一论述是错误的，证明如下：

证：假设这一论断是正确的，由于加密方法是完美保密的，我们根据完美保密的定义 1 可以得出：

$$Pr[M=m|C=c]=Pr[M=m]$$

又因为给出了条件：

$$\forall m,m' \in M, \forall c \in C\ , Pr[M=m|C=c] = Pr[M=m'|C=c]$$

我们可以推断出：

$$\forall m,m' \in M, Pr[M=m] = Pr[M=m']$$

因此我们可以得到随机变量 M 的分布函数是确定的，假设|M|=X，则有：

$$\forall m \in M, Pr[M=m] = \frac{1}{|X|}$$

但是题目中给的是 M 的分布函数是任意的，因此有矛盾所以这一论述是错误的。

Problem 4:

(a) 证明：

首先证明移位密码的密钥空间大小为 26，由于移位密码中的密文字母 c 与明文字母 m 以及密钥 k 满足：

$$c = Enc_k(m) = (m+k) mod 26$$

因此我们可以知道当$k \notin [0,25]$时我们可以通过求余运算的性质将其变为$k' \in [0,25]$。因此我们可以认为 K 的密钥空间大小为 26

而且变换后的密文 C 与变换前的明文 M 其空间大小也为 26。并且移位密码的密钥是在其密钥空间上等概率选取的，因此$\forall k \in K, Pr[K=k] = \frac{1}{|K|}$。

并且$\forall m \in M, \forall c \in C\ \ k = (m-c) mod 26$ 由于 m 与 c 是确定的，并且由于 k 的取值范围限定在了 0 到 25 之间，因此 k 也是确定且唯一的。

所以由香农定理可知在该条件下移位密码是完美保密的。

(b)我认为其为$26!$

首先由于要使单字母替换为完美保密我们由定理 5 及完美保密的局限性知道|M|<=|K|所以明文空间 M 的大小最大为 K 的密钥空间大小及 26!。下证当 M 取到 26！时单字母替换为完美保密。

首先由于其映射函数的一对一特性密文空间与明文空间大小一定相同所以|M|=|C|=|K|，此外其密钥选取满足$\forall k \in K, Pr[K=k] = \frac{1}{|K|}$，并且由于其映射的唯一性给定任意明文与密文其对应的密钥是唯一的。所以由香农定理当|M|=26！单字母替换是完美保密的。

由上面的论述要想单字母替换为完美保密其明文空间 M 的大小最大为 26!。

(c)密钥长度应该和明文长度相同为 t 并且满足$\forall k \in K, Pr[K=k]=\frac{1}{|K|}$。

下证其满足完美保密的要求。

证明:

首先我们很容易知道密文长度为 t，密文空间大小与明文空间与密钥空间相同，并且由于对于明文 M 与密文 C 的位置为 n 的字母均满足以下公式:

$$(M_n + K_n) mod 26 = C_n \quad \Rightarrow \quad (C_n - M_n) mod 26 = K_n$$

因此我们很容易由取余运算的性质知道密钥 K 在明文与密文确定的情况下是唯一的。因此由香农定理我们可以知道此时的 Vigenère 密码满足完美保密。

Problem 5：

（a）我认为首先从完美保密来说这种改动是不应该的，经过改变后其已经不是完美保密的加密方式了，因为密钥空间的大小小于明文空间的大小。这同时也意味着会暴露关于原文的信息，例如攻击者可以知道密文里绝对不会出现明文的任何信息，那么如果在密文中出现了某个地址，那么明文中肯定不是对应的地址，这就给了攻击者明文中的信息的提示。

其次由于密钥为全零的概率与别的任何一个密钥的概率一样是很小的，我们没有理由删除它。我们举一个例子，我们对 one 进行加密，我们知道其加密为 one 与加密为 two 的可能性是相同的因为其对应的密钥都是一个可能性均为 $\frac{1}{|K|}$。所以即使对方收到了对应的明文也没法假设它就是明文。反而如果删除了它受到 two 后可以肯定原文不是 two 会对保密性造成影响。

（b）首先从公式上 OTP 是符合完美保密的定义的，因此我们可以知道在知道密文的情况下攻击者无法获得关于明文的任何信息，即使他获得了明文他也会认为其原文很可能不是这样，因为 ASCII 码长度与密文相同的任何明文加密为该明文的概率都是一样的。

我们现在的问题是为什么即使明文可能传给对方的情况下 OTP 仍然是完美保密的。举个例子，我发送了一句话 I like apple，其可能会被发给对方，但是如果对方收到了这句话他是否能相信这句话呢?

我们假设这个攻击者是完全理性的，从他的角度来看，我的原句子可能是 I like apple，也可能是 I hate apple，也可能是 I hate fruit。这三者由明文加密为密文的概率是相同的，此外还有很多种可能性，只要密文对应的 ASCII 码长度与明文一样长则均可能加密出来且可能性与 I like apple 完全相同均为 $\frac{1}{|K|}$，所以一个理智的攻击者会把这句话直接当作明文的可能性基

本不存在。

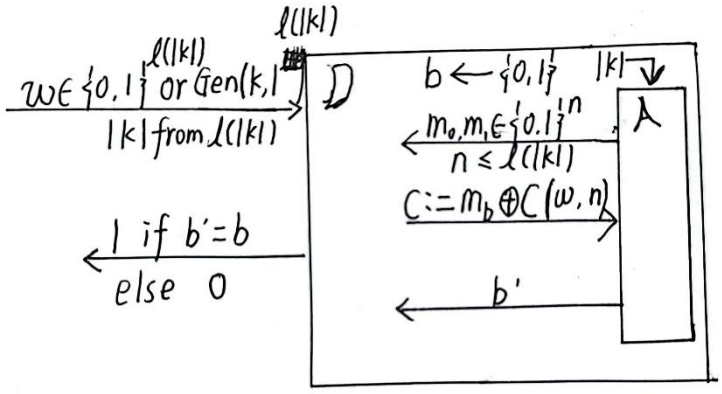
HomeWork2

Problem 1：

假设给定加密方案$\tilde{\Pi}=(Gen,Enc,Dec)$，$Gen$：密钥长度为$n$的变长伪随机生成器,$Enc$：$c:=m\oplus G(k,1^{|m|}),Dec$：$m:=c\oplus G(k,1^{|c|})$。并且加密的明文的长度最大为$\ell(|k|)$，其具有不可区分加密。

证明：

我们构造攻击者$\mathcal{A}$来攻击加密方案$\tilde{\Pi}$。然后用攻击者$\mathcal{A}$来构造区分器$\mathcal{D}$来区分生成器 G。这样当$\mathcal{A}$可以区分$\tilde{\Pi}$时，$\mathcal{D}$可以区分生成器 G。但既然伪随机生成器存在，那么显然可以得到$\mathcal{A}$无法区分$\tilde{\Pi}$，否则假设不成立。具体构造如下：



图中算法 C 的作用是取字符串ω的前 n 位与明文异或生成密文 c，其余符号在上文已经做了解释。

接下来对为何输入 D 中的随机字符串的长度是$\ell(|k|)$进行解释：由于变长随机生成器具有性质$G(s,1^{\ell})$是$G(s,1^{\ell'})$的前缀，$l<l'$，所以只要输入长度为$\ell(|k|)$的伪随机字符串，当使用时取对应的前 n 位即可，与变长随机生成器直接生成效果是相同的。因此“当$\mathcal{A}$可以区分$\tilde{\Pi}$时，$\mathcal{D}$可以区分生成器 Gen”这句论断仍然成立。

1.当输入的字符串ω为 u.a.r 选取的，那么此时的加密方案即为 OTP，那么我们可以得到：

$$Pr(\mathcal{D}(\omega)=1)=Pr\left(PrivK^{eva}_{\mathcal{A},\tilde{\Pi}}(|k|)=1\right)=\frac{1}{2} \tag{1.1}$$

2.当输入的字符串为 G 生成伪随机字符串时，那么此时的加密方案即为$\tilde{\Pi}$，我们对实验成功的概率作出假设后可以得到：

$$Pr\left(\mathcal{D}\left(G\left(k,1^{|k|}\right)\right)=1\right)=Pr\left(PrivK_{\mathcal{A},\tilde{\Pi}}^{eva}(|k|)=1\right)=\frac{1}{2}+x(n) \tag{1.2}$$

其中 x(n)为未知的与 n 有关的函数。

由于题目中假设变长伪随机生成器存在显然我们有：

$$|Pr\left(\mathcal{D}(\omega)=1\right)-Pr\left(\mathcal{D}\left(G\left(k,1^{|k|}\right)=1\right)\right)|\leq negl(n) \tag{1.3}$$

我们将 1.1，1.2，1.3 结合可以知道：

$$|Pr\left(PrivK_{\mathcal{A},\tilde{\Pi}}^{eva}\right)-\frac{1}{2}|=|x(n)|\leq negl(n) \tag{1.4}$$

所以由定义我们可以知道加密方案$\tilde{\Pi}$为在窃听者存在下的不可区分加密。

Problem 2：

1. 当$g(s)=f(s)\oplus f'(s)$时我觉得$g(s)$不一定是 PRG。

因为当这两者有关系时，例如：$f'(s)$是$f(s)'$每位取反的时候，此时两者单独来看仍然满足题目中所说的都是 PRG，但是$g(s)$的输出是一个定值即为长度为 n 的每位为 1 的字符串，显然它不是 PRG。

2. g(s)不一定为 PRG。

首先我们认为当一个长度较短的串 s 与长度较长的串$f(s)$进行异或时会先在 s 前面补 0 致两者长度相同，然后异或。我们假设有一个伪随机生成器ω(s)其可以输入长度为|s|-1 的随机串，可以生成长度为|f(s)|-1 的伪随机串。假设随机串 s 除去最低位得到的串为$D(s)$，我们构造 f(s)如下：

$$f(s)=\omega(D(s))\ ||\ LSB(s) \tag{1.5}$$

那么 f(s)是一个 PRG。 我们可以利用生成器是否是为伪随机的与“生成的串能否通过前 k 位（k 大于等于 1 小于 n）的值预测 k+1 位的值”等价这一定理（Yao‘82）来说明这一事实：

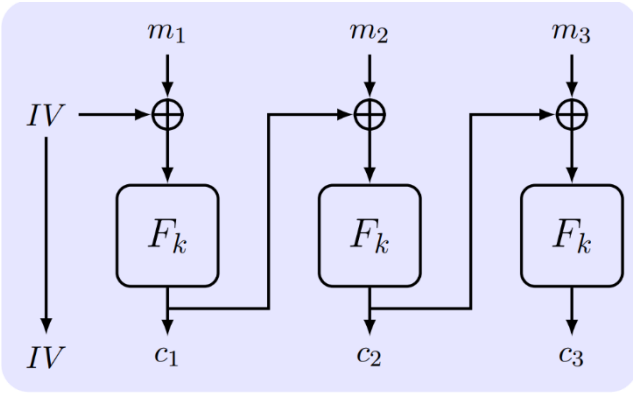
前|f(s)|-1 位一定符合该条件因为其是伪随机生成器的输出。后一位因为其与前面生成的串无关，且本身是随机生成的串的最后一位所以无法用前几位推出。因此 f(s)是为随机生

成器。

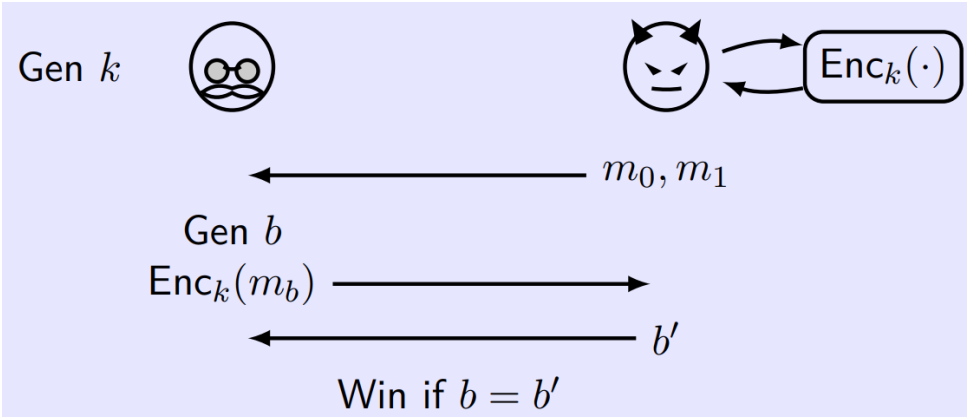
所以 g(s)的最后一位可以被构造为 0。这样 g(s)不符合随机生成器的定义，所以 g(s)不一定为 PRG。

Problem 3：

我们构造 CBC 加密方案的变体，我们令加密时的 IV 每次加密消息时加 1，这个方案是非 CPA 安全的但是具有不可区分的多次加密安全性。



首先我们证明其是非 CPA 安全的，假设消息长度为 n，敌手 A 在发送消息之前先向数据库询问消息m_0，并且得到$<IV, F_k(IV \oplus m_0)>$我们把返回的密文记作c_0。我们接着构造新的消息$m_1 = m_0 \oplus (IV+1) \oplus (IV), m_2 = \{0\}^n$，那么如果选择加密$m_1$就会得到$c_0$，我们此时返回 1 否则返回 0，假设密钥长度为 k，如果 F 为伪随机置换， 那么如果我们最初的$IV \oplus m_0$与 IV+1 不相同，那么我们成功，否则成功概率为 1/2。所以在大部分选择正确的情况下，



$$|PrivK^{cpa}_{\mathcal{A},\Pi}(n) = 1| > \frac{1}{2} + negl(n) \tag{1.6}$$

所以该加密方案不为 CPA 安全的。

当面对多消息加密安全实验时，由于其不具有访问数据库的能力，并且与别的确定方案不同这里相同的明文产生相同的密文的概率极小，因为即使相同的明文，不同部分输入伪随机置换函数的串也是随机的。因此并不存在可以通过多次加密相同的明文来得到关于输入串信息的情况。因此我认为其是具有不可区分的多次加密安全性的。

Problem 4:

我们可以构造如下的伪随机生成器：

$$G_k(l)=F_k(1)||F_k(2)\cdots||F_k(l) \tag{1.7}$$

我们得到了一个扩展因子为 l*n 的伪随机生成器（其中 n 为定长伪随机函数的输出长度），因此对于我们题目中要求的变长伪随机生成器我们可以这么构造，当消息长度 m 小于 n*n 时：

$$B(k,1^l)=\begin{cases} G_k(l)\ \ (l\%n=0) \\ pre_l(G_k(l'))\ (l'\text{满足：大于}l\text{，距离}l\text{最近，}l'\%n=0) \end{cases} \tag{1.8}$$

显然我们只要已知长度为 x 的伪随机生成器通过取前缀一定可以构造出长度小于 x 的伪随机生成器，因为 A 只要可以区分伪随机串的前缀和随机串，那么其一定可以区分整个生成的伪随机串与随机串，而这与存在长度伪 x 的伪随机生成器矛盾。

接下来我们只要证明 G 满足其为扩展因子为 l*n 的伪随机生成器即可。

我们构造 D 来区分 G 与真随机串，我们可以得到

$$\varepsilon(n)=|Pr_{r\leftarrow\{0,1\}^{l*n}}[D(r)=1]-Pr_{s\leftarrow\{0,1\}^n}[D(G_s(l))]| \tag{1.9}$$

接着我们构造 A 来攻击伪随机函数F_k，然后我们将串$O(0)||O(1)\cdots||O(l)$输入给它，其中 O 为 f 与 F 中的一个，显然

$$|Pr[A^{f(\cdot)}(1^n)=1]-Pr[A^{F_k(\cdot)}(1^n)=1]|=\varepsilon(n) \tag{1.10}$$

并且由于 F 是为随机函数所以$\varepsilon(n)$是可以忽略的。

Problem 5：

与第三问的构造类似，只不过之前是每次加密后可以预测到加一，这次只是可以预测，假设预测下一轮的 IV 值为$P(IV)$，那么与之前类似，这次只叙述公式的变化：

$$m_0=<IV,F_k(IV\oplus m_0)> \tag{1.11}$$

$$m_1 = IV \oplus m_0 \oplus P(IV) \tag{1.12}$$

m_2 为除了 m_1 以外的任何值均可，此时 $|PrivK^{cpa}_{\mathcal{A},\Pi}(n)=1| = 1 > \frac{1}{2} + negl(n)$ 所以其不具有 CPA 安全。

Problem 6：

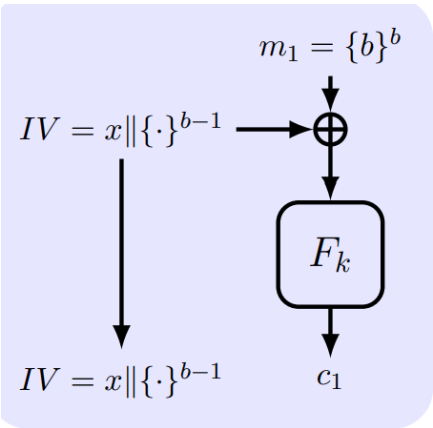
我们可以构造一个攻击者，为了简单假设其只加密两块长的消息 $m_0 m_1$，假设：$m_0 = 0x^{n-1}0y^{n-1}$，$m_1 = 1x^{n-1}1y^{n-1}$，我们假设攻击者受到密文后将其第一位反转后输入解密数据库，，根据定义这是可行的，我们接下来对其进行分类讨论与破解：

对于 CBC 模式我们会发现若收到的解密值的第 n+1 位为 1 则说明其为 m_0 我们输出 0，否则输出 1，这样就可以破解了，这是由于其加密性质的缘故。

对于 OFB 与 CTR 模式，我们直接观察收到的解密值的第一位，若其为 1 则说明其为 m_0 我们输出 0，否则输出 1，这样就可以破解了。

Problem 7：

经查阅资料，我发现 PKCS 5 中块的长度是 8 字节。那么我们构造如下的攻击方式确定其信息长度是否为 1bytes。我们假设信息长度为 1 比特那么信息之后会附上 7 个大小为 7 的 1 比特的信息，用 16 进制即为 07 07 07 07 07 07 07。

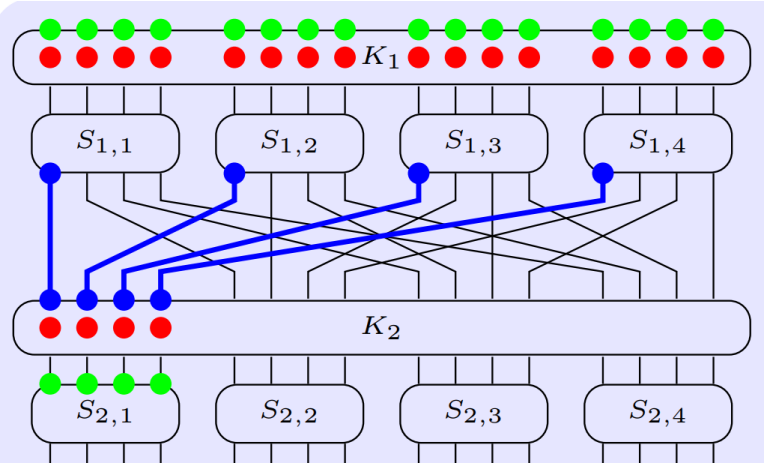


以下是攻击过程：我们把收到的加密后的信息中的 IV 提取出来，假设其信息为 1A 07 05 02 01 A7 37 那么我们修改第二比特的信息，将其变为 08，然后我们将修改后的 IV' 放入原信息中，接着我们把该信息送往 CAPTCHA 服务器，如果确实原来的信息就是 1byte 的话其在解密时会发现附加原来的信息变为了 $07 \oplus 08 \oplus M_2$ 其中 M_2 表示原文第二比特，要是 padding 是 7 个字节的话其现在后七个字节已经变成了 08 07 07 07 07 07 07，出现错误，反之要是不是 7 个字节的话，更改不会造成任何影响，比如是 6 个字节后七个字节变为 08 06 06 06 06 06 06 不会对 padding 格式造成影响，也就不会返回错误。

总结下来就是对第二比特的信息做修改。要是返回错误，则原消息是 1 比特，否则则不是。

HomeWork3

Problem 1：



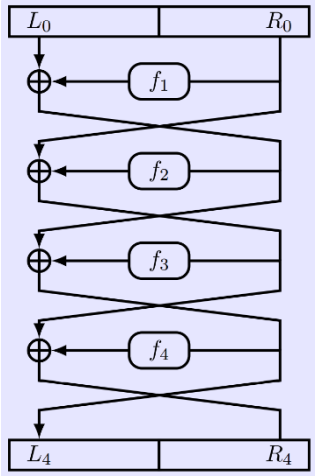
1. 此时我们先猜测 8+8*8=72 位的密钥，其中 8 位属于K_2另外的 64 位属于K_1。如果我们猜测错误的话，由于输出是随机的，因此测试通过的概率为2^{-8}。并且由于我们要遍历所有的密钥，所以遍历的次数为2^{72}。因此我们选择使用 16 对输入输出对（其实大于 9 对即可，**但是**一般使用 2 的幂，所以选择了 16 对输入输出对。）来使得随机遍历时出错的期望下降到小于 1（$2^{72-16\times 8}=1.38778\times 10^{-17}$）。此时我们的复杂度为$16 * 2^{72} * 8 = 2^{79}$。
2. 同样的我们先猜测 72 位的密钥，其中 8 位属于K_2另外的 64 位属于K_1，接着选择的 IO 对的数目与第 1 问相同，唯一不同的是复杂度为$16 * 2^{72} * 16 = 2^{80}$。
3. 从上面看我们发现复杂度其实与分块长度和 Sbox 的长度均相关。下面我们进行分析:

   我们假设分块长度为 B，Sbox 的长度为 S（S<B，且$B\%S=0$，且$S^2 \le B$）。首先我们要猜测 S+S*S 位的密钥，然后我们用的输入输出对的数目为 k，由于出错的期望：$2^{(S+1)*S-k*S}<1$所以我们假设 k 选择 S+2（先抛去 2 的幂次这一习惯）。因此我们的复杂度为$(S+2) * 2^{S*(S+1)} * \frac{B}{S}$，我们发现当任意一者增加时都会使复杂度增加。复杂度的上界为$2B * 2^{S^2+S}$（假设 S 大于等于 2）。

   我们以复杂度的上界为准，假设 S 增加百分之 a，B 不变其变化率为：$2^{a^2S^2+3aS}$，S

不变，B 增加百分之 a 变化率为：$1+a$。考虑到二者的实际值，我认为在某一固定的个 S，B 点，两者增加相同比率，S 增加产生的变化率大于 B 增加产生的变化率。

Problem 2：



证明:

我们先对如果输入 x 与密钥 k 取反进行简单的分析，由于 DES 会对 k 以及输入进行扩展我们需要分析经过取反 DES 变化之后的 x 与 k 与原 x 与 k 的关系。

先分析 k，DES 需要的 16 个子密钥是由主密钥先经过置换移位选择得到的，这些操作都不会改变密钥本身的数值，而且对密钥进行选择操作得到的位数也与密钥本身无关，所以我们认为得到的第 i 轮的密钥K_i'是原来的第 i 轮密钥K_i的按位取反即$K_i'=\bar{K}_i$。

由于明文输入进行处理之前先要进行 IP 置换，假设其产生的新明文叫$IP(x')$，我们接下来的分析就由它进行，由于其进行的是置换其与之前的输入满足$IP(x')=\overline{IP(x)}$。

接下来，我们对 sp 网络中用到的扩展函数进行分析，由于其只会复制输入 x 一半的比特因此其不会对 x 的数值位产生变化。我们假设每一轮用到的 sp 网络为函数f_i，观察其构造我们容易得出：$f_i(k,x)=f_i\left(\bar{k},\bar{x}\right)$。

首先我们有：$R_0'=\overline{R_0}$，$L_0'=\overline{L_0}$。接下来我们分析当 x 与 k 反转后得到的L_1',R_1'与原来的有何区别:

$$L_1'=R_0'=\bar{L}_1 \tag{1.1}$$

$$R_1'=f_1(K_1',R_0')\oplus L_0'=L_0'\oplus f_1(K_1,R_0) \tag{1.2}$$

并且我们可以由异或的性质得到以下的公式：

$$x \oplus \overline{y} = \overline{x \oplus y} \tag{1.3}$$

所以有

$$R_1' = \overline{R_1} \tag{1.4}$$

假设对于第 i 轮我们的假设 $R_i' = \overline{R_i}$，$L_i' = \overline{L_i}$ 成立，我们对第 i+1 轮进行推理：

$$L_{i+1}' = R_i' = \overline{R_i} = \overline{L_{i+1}} \tag{1.5}$$

$$R_{i+1}' = L_i' \oplus f_{i+1}(K_{i+1}', R_i') = L_i' \oplus f_{i+1}(K_{i+1}, R_i) = \overline{R_{i+1}} \tag{1.6}$$

发现其仍然符合此规律。

所以我们不难知道 16 轮以后我们的规律仍然存在，因此经过 Feistel 网络之后的 $Feistel(IP(x')) = \overline{Feistel(IP(x)}$，再经过一次逆 IP 置换我们可以知道这不会对这一关系产生影响，所以我们有：

$$DES_k(x) = \overline{DES_{\overline{k}}(\overline{x})} \tag{1.7}$$

■

Problem 3：

我认为这个函数不是单向函数，因为加法给定一个 $f(x,y)$ 可以找到多个符合要求的（x，y）对，这样的话就可以轻易的构造攻击者 A。例如我有一个攻击者 A 他返回以下的值：

$$\begin{cases} (0,0)\,(f(x,y)=0) \\ (f(x,y)-1,1)\,(f(x,y)>0) \end{cases}$$

很容易验证这个攻击者成功的概率为 1，这与我们的定义不符合。

Problem 4:

我认为不一定是单向函数：

证明：

下面说明存在反例使得 f 不为单向函数。假设 f_1, f_2 均由单向函数 f_3 构造而来，且均定义如下：$f_1, f_2 : \{0,1\}^{256} \to \{0,1\}^{256}$。构造如下的函数 f:

$$f_1(x||x_1) = f_3(x)||x_1(|x| = |x_1|, |x_1| > 128)$$
$$f_2(x||x_1) = f_3(x_1)||x(|x| = |x_1|, |x| > 128)$$
$$f(x_1||x_2) = f_1(x_1, x_2)||f_2(x_1, x_2) = f_3(x_1)||x_2||f(x_2)||x_1$$

其中f_1, f_2均为 ppt 上的例子，均是可逆函数。我们可以很轻松的为 f 构造攻击者 A:

把输入的值的 129 到 256 位作为x_2第 385 到 512 位作为x_1然后返回(x_1, x_2)。实验每次都会成功，所以以f_1, f_2作为参数的函数 f 不一定是可逆的。

■

Problem 5：

第一问我认为$f(f(x))$不一定是单向函数，我们可以构造一个函数 f 来证明。

证明:

假设存在单向函数 h, 有: $h: \{0,1\}^n \to \{0,1\}^n, f: \{0,1\}^{2n} \to \{0,1\}^{2n}$。且有$x \in \{0,1\}^n$，$x_1 \in \{0,1\}^n$。

构造 f 如下:

$$f(x||x_1) = \begin{cases} \{0\}^{2n} (x = \{0\}^n) \\ 0^n||h(x) \end{cases} \tag{1.8}$$

我们很容易知道$f(f(x))$的值恒为为长度为 2n 的 0 串，因此不是单向函数。

下证 f 是单向函数:

我们假设$Pr[Invert_{\mathcal{A},f}(1^{2n}) = 1] = c(n)$，下面我们构造$\mathcal{A}'$攻击函数 h，函数 h 向$\mathcal{A}'$输入$(1^n, h(x))$、然后$\mathcal{A}'$将$(1^{2n}, 0^n||h(x))$输入给$\mathcal{A}$，并把$\mathcal{A}$的输出的前 n 位直接当作$\mathcal{A}'$的输出。我们可以得到:

$$Pr[Invert_{\mathcal{A}',h}[1^n] = 1] \geq Pr[Invert_{\mathcal{A},f}[1^{2n}] = 1 \,|\, x \neq \{0\}^n] \tag{1.9}$$

由于我们有:

$$\begin{aligned} Pr[Invert_{\mathcal{A},f}[1^{2n}] = 1] = {} & Pr[Invert_{\mathcal{A},f}(1^{2n}) = 1 \,|\, x \neq \{0\}^n] \cdot Pr[x \neq \{0\}^n] \\ & + Pr[Invert_{\mathcal{A},f}(1^{2n}) = 1 \,|\, x = \{0\}^n] \cdot Pr[x = \{0\}^n] \end{aligned} \tag{1.10}$$

化简得:

$$\begin{aligned} Pr[Invert_{\mathcal{A},f}[1^{2n}] = 1] &\leq Pr[Invert_{\mathcal{A},f}(1^{2n}) = 1 \,|\, x \neq \{0\}^n] + Pr[x = \{0\}^n] \\ &\leq Pr[Invert_{\mathcal{A}',h}(1^n) = 1] + Pr[x = \{0\}^n] \end{aligned} \tag{1.11}$$

及:

$$Pr[Invert_{\mathcal{A}',h}(1^n)=1]\geq c(n)-\frac{1}{2^n} \tag{1.12}$$

由于 h 是单向函数，且 $\frac{1}{2^n}$ 可忽略，所以 c(n)可忽略，所以 f 是单向函数。

■

第二问我认为 g（x）是单向函数，

证明：

我们构造$g(x)=f(x)||f(f(x))$，其中：$f:\{0,1\}^n\to\{0,1\}^{2n}$，$x\in\{0,1\}^n$。假设有一个对其的攻击者$\mathcal{A}'$，我们有一个对单向函数 f 的攻击者$\mathcal{A}$利用$\mathcal{A}'$进行攻击。

首先$\mathcal{A}$收到 f(x)后计算 f(f(x))然后将$(1^n,f(x)||f(f(x)))$传给$\mathcal{A}'$，然后将$\mathcal{A}'$的输出直接输出，那么我们可以得到：

$$Pr[Invert_{\mathcal{A}',g}(1^{2n})=1]=Pr[Invert_{\mathcal{A},f}(1^n)=1]=\varepsilon(n)$$

又因为 f 是单向函数所以我们得到 g 也是单向函数。

■

HomeWork4

Problem 1：

由题意我们可以知道F_k:$\{0,1\}^n \to \{0,1\}^n$，及函数 F 的输入与输出位数相同。我们构造如下的攻击者$\mathcal{A}$来攻击此 MAC:

攻击者先询问数据库长度为 2n 的串$m_1||m_1$, 其中m_1是随机选取的, 所以我们得到 tag: $\langle F_k(m_1), F_k(F_k(m_1))\rangle$。接着伪造输入为$F_k(m_1)||m_1$的 tag 为$\langle F_k(F_k(m_1)), F_k(F_k(m_1))\rangle$并将其输出给老头。那么其成功的概率我们不难计算出为：$Pr[Macforge_{\mathcal{A},\Pi}(2n)=1 \,|m_1 \neq F_k(m_1)] \cdot X + Pr[Macforge_{\mathcal{A},\Pi}(2n)=1 \,|m_1 = F_k(m_1)] \cdot \bar{X}$ 其中$X = Pr[m_1 \neq F_k(m_1)]$。由于后一项我们无法知道其概率大小但是其一定大于等于 0, 所以我们取前面的一项，成功的概率肯定大于它。

由于当$m_1 \neq F_k(m_1)$时伪造成功的概率为 1。所以伪造成功的概率大于：$1 \cdot Pr[m_1 \neq F_k(m_1)] = 1 - negl(n)$, 所以$Pr[Macforge_{\mathcal{A},\Pi}(2n)=1] > negl(n)$。所以此 MAC 不具有在适应性 CMA 下的存在性不可伪造性，是不安全的。

Problem 2：

证明:

我们对其进行规约, 假设我们有对$\hat{H}$的攻击者$\mathcal{A}$, 我们利用$\mathcal{A}$构造攻击者D来攻击$\mathcal{H}$, 则D能成功破解$\mathcal{H}$当且当$\mathcal{A}$能成功实现对$\hat{H}$的攻击。

首先D收到生成的哈希函数的 key s，接着将其输入到攻击者$\mathcal{A}$中，$\mathcal{A}$会输出两个不同的输出满足：$H^s(H^s(x_1)) = H^s(H^s(x_2)), x_1 \neq x_2$。由于给定 key 的哈希函数是确定，所以函数给定相同的输入时输出相同。我们构造D的输出如下：

1. 若$H^s(x_1) = H^s(x_2)$那么我们直接输出x_1, x_2即可满足。
2. 若$H^s(x_1) \neq H^s(x_2)$那么我们输出$H^s(x_1), H^s(x_2)$即可满足。

假设$\mathcal{H}$是防碰撞的哈希函数我们可以得到$Pr[Hashcoll_{D,\Pi}(n)=1] \leq negl(n)$，所以我

们的对$\hat{H}$攻击满足：$Pr\left[Hashcoll_{\mathcal{A},\hat{\Pi}}(n)=1\right]\leq negl(n)$，所以我们可以得到$\hat{H}$也是防碰撞的。

■

Problem 3：

1. 不是防碰撞的。例如：假设h^s的输入长度为 n，我们构造长为 n+1 的串$x||0$，以及一个长为 n+2 的串$x||00$。由于我们构造的新 transform 无需使用长度我们得到二者的结果均为：$h^s(h^s(IV||x)||0^n)$，所以其不是防碰撞的哈希函数。
2. 是防碰撞的，证明如下：

   由于其最后一个块的输出为$z_B||L$，所以如果两个输入的长度不同最终得到的结果肯定不同，无法发生碰撞。

   所以我们只需要证明输入长度相同下的不同输入的碰撞即可，此时碰撞只可能发生在前 B 个块，我们假设这种情况出现：

   那么从后向前分析：

   1. $x_B\neq x'_B\ or\ z_{B-1}\neq z'_{B-1}$，此时第 B 个块发生碰撞。
   2. $x_B=x'_B\ and\ z_{B-1}=z'_{B-1}$此时碰撞发生于前 B-1 个块。

   由于$IV=IV'\ and\ x\neq x'$所以按照这样的推导，肯定会在某个前 B 个块发生碰撞，而我们根据 h 是防碰撞的得出这不可能，所以存在矛盾，即假设错误，碰撞不会发生。

   ■

3. 是防碰撞的，与上面的证明类似，只简述不同：

   证明：

   首先这种情况下如果长度不同，我们可以得到两个输入的最后一块 h 的输入$z_B||L$是不同的，所以此时最后一块发生了碰撞，但是这在 h 是防碰撞时是不可能的。

   当长度相同时，与上面的证明相同，只不过若一直推到碰撞发生在第一个块时我们可以得到$x_1\neq x'_1$（因为之前的块中满足$x_i=x'_i\ and\ z_{i-1}=z'_{i-1}$）且$z_1=z'_1$而我们知道这是不可能的因为$z_1=x_1$，所以碰撞不可能发生。

因此在这种 transform 下碰撞不可能发生。

■

4. 不是防碰撞的，假设我们构造如下的单向函数，假设$g:\{0,1\}^{2n}\to\{0,1\}^{n-1}$且 g 是防碰撞的单向函数。根据提示的思想构造一个新的单向函数$h:\{0,1\}^{2n}\to\{0,1\}^{n}$:

$$h(x)=\begin{cases} l & x=(2n)_2||x_1 \\ g(x)||0 & else \end{cases} \qquad x\in\{0,1\}^{2n} \tag{1.1}$$

其中l为表示 n 这个数的二进制串，其且长度为 n（需要注意这里的 l 最后一位为 1）（不满 n 在前面补零），上面的条件是当 x 等于表示 2n 的数的二进制串（其长度为 n，构造方法同上）与长为 n 的一个串x_1连接。

此时我们构造两个输入使其碰撞：$x_1||x_2,x_2$。其中$x_1,x_2\in\{0,1\}^n$。当输入$x_1||x_2$时首先第一个块中输入为$(2n)_2||x_1$（前面为附加的 L）所以输出为$z_1=l$，与第二个块的输入x_2综合得出结果为$g(x_2)||0$。同样的我们可以得到输入x_2时结果为：$g(x_2)||0$发生碰撞，所以其不是防碰撞的。

总结一下其构造思想是根据防碰撞哈希函数的性质构造一个新的哈希函数。两个长度相差 1 个块的大小，及$len(m)-len(m')=n$，但是除第一部分外二者组成相同的输入值输入哈希函数，会因为构造的原因导致$z_1=len(m')=z_0'$，因此输出相同，所以其不防碰撞。

■

Problem 4:

（1）我认为其是 CCA 安全的:

证明:

我们证明它通过将其归约到证明强伪随机置换与随机置换不可区分上，假设我们构造攻击者$\mathcal{A}$攻击此加密策略，同时利用它构造攻击者$\mathcal{D}$来攻击强伪随机置换。

开始输入加密与解密函数$\mathcal{O},\mathcal{O}^{-1}$其可能是强伪随机置换也可能是随机置换，我们用它来回答$\mathcal{A}$的询问，我们有:

$$Pr[PrivK^{cca}_{\mathcal{A},\Pi}(n)=1]=Pr[D^{F_k(\cdot),F_k^{-1}(\cdot)}(n)=1]=\varepsilon(n)+\frac{1}{2} \tag{1.2}$$

我们接下来分析当输入的加密函数为随机置换函数时发生的情况：

首先攻击者会向加密数据库与解密数据库询问得到原文与密文对，这里我们假设其采取最优的策略及尽量得到与m_1,m_2相关的明文密文对来便于攻击。、

首先分析向加密数据库分析的情况：在 D 发送密文前后我们都只进行m_1,m_2的询问并将结果的密文与对应的 m 存储于 E 中，容易知道其询问数量必然是多项式次用$q(n)$次表示，最佳情况每次加密用的随机串 r 均不同，一般情况下满足：$|E|\le q(n)$。

接着我们分析对解密数据库的询问，此时因为无论在 D 发送密文前还是后，无论询问什么密文得到的明文都是随机的，所以我们只能随机进行询问期望得到与m_1,m_2相关的信息，同样的此时询问了多项式$x(n)$次。此时由于是伪随机置换，用不同密文得到的明文信息必然是不同的，将得到的明文中与m_1,m_2相关的、对应的明文密文对存储于 H 中，最佳情况询问得到的明文均是与m_1,m_2相关的，所以有$|H|\le x(n)$，

接下来我们分析两个集合组合起来可以得到多少与m_1,m_2相关的明文密文对：假设最不可能的情况我们有两个集合中的元素没有一个相同，那么我们有将两个集合做加法操作后得到的集合 K 有$|K|\le x(n)+q(n)$。我们用集合 K 的信息回答质询c，假设 c 对应的明文为m_n可以得到：

$$\begin{aligned}Pr[PrivK^{cca}_{\mathcal{A},\Pi'}(n)=1]&=Pr[PrivK^{cca}_{\mathcal{A},\Pi'}(n)=1\,|\,(c,m_n)\in K]+Pr[PrivK^{cca}_{\mathcal{A},\Pi'}(n)=1\,|\,(c,m_n)\notin K]\\&\le 1\cdot\frac{x(n)+q(n)}{2^{\frac{n}{2}+1}}+\frac{1}{2}\cdot\left(1-\frac{x(n)+q(n)}{2^{\frac{n}{2}+1}}\right)\\&<\frac{1}{2}+\frac{x(n)+q(n)}{2^{\frac{n}{2}+1}}=\frac{1}{2}+negl(n)\end{aligned}$$

这里当(c,x_n)不在 K 中时由于随机置换的定义我们知道其成功的概率为 1/2。所以我们有：

$$Pr[PrivK^{cca}_{\mathcal{A},\Pi'}(n)=1]=Pr[D^{f_k(\cdot),f_k^{-1}(\cdot)}(n)=1]=\frac{1}{2}+negl(n) \tag{1.3}$$

结合上面的式子（1.2）我们知道：

$$Pr(D^{F_k(\cdot),F_k^{-1}(\cdot)}(n)=1)-Pr[D^{f_k(\cdot),f_k^{-1}(\cdot)}(n)=1]=\varepsilon(n)-negl(n) \tag{1.4}$$

而由于 F 为强伪随机置换所以我们有$Pr[PrivK^{cca}_{\mathcal{A},\Pi}(n)=1]=\frac{1}{2}+\varepsilon(n)=\frac{1}{2}+negl(n)$

因此此加密方案满足 CCA 安全

■

（2）我不认为 CCA 安全就可以得出其实现了认证通信，我们以第一问的加密方案为例，由于其使用了强伪随机置换，所以我们向Dec_k输入的任何密文均会被解密且均有意义，不会输出$\perp$。所以我只需要输出任意的一个密文，其均有有意义的明文与其对应。因此满足：$Pr[Auth_{\mathcal{A},\Pi}(n)=1]=1>negl(n)$。所以其不满足认证通信，但由第一问可知其满足了 CCA 安全。

Problem 5：

我们利用安全的 MAC$\langle Mac,Vrfy\rangle$构造这样的数据传输策略：

$$\begin{cases} EncMac_k(m)=(Mac_k(m),m) \\ Dec_k(c_1,c_2)=\begin{cases} c_2 & Vrfy_k(c_2,c_1) \\ \perp & otherwise \end{cases} \end{cases} \tag{1.5}$$

我们用规约的方法证明其实现了认证通信：

首先我们还是假设攻击者$\mathcal{A}$攻击我们的认证通信，我们利用它构造攻击者$\mathcal{D}$来攻击 MAC，$\mathcal{D}$得到Mac_k的访问权，并且将$\mathcal{A}$的询问数据$(m_1,m_2,m_3,\cdots,m_n)$输入Mac_k得到输出后返回：

$((Mac_k(m_1),m_1),\cdots,(Mac_k(m_n),m_n))$，并且将$\mathcal{A}$的输出$(Mac_k(m_0),m_0)\quad m_0\notin\{m_1,\ldots,m_n\}$中的$Mac_k(m_0)$返回。

我们可以得到：

$$Pr[Auth_{\mathcal{A},\Pi}(n)=1]=Pr[Macforge_{\mathcal{A},\Pi}(n)=1] \tag{1.6}$$

所以我们的方法实现了认证通信，但是其将明文暴露了出来明显其不是 CCA 安全的。

HomeWork5

Problem 1：

我们首先容易得到$N=35=5*7$所以说我们可以计算 N 的 Euler phi 函数值及：$\phi(N)=(5-1)^1*(7-1)^1=24$而我们根据欧拉定理可以得到$\forall x\in(\mathbb{Z}_N)^*, x^{\phi(N)}=1\ in\ \mathbb{Z}_N$我们将$101^{4,800,000,023}=(2*35+31)^{4,800,000,023}$容易得到原式子转换为$31^{4,800,000,023} mod 35$然后利用欧拉定理得到将原式子化简得$31^{23} mod 35=31^{-1} mod 35$。

而求 mod 下的逆元，我们可以用扩展欧几里得算法求$a*31+b*35=1$中的 1 值，计算得 a=26，所以原式子的值为 26。

Problem 2：

我们假设使用欧拉函数得到的值为x_1，及$\phi(N)=p^{1-1}(p-1)*q^{1-1}(q-1)=x_1$，我们有$N=p*q$那么我们可以得到与 p 相关的式子：$p^2-(N+1-x_1)p+N=0$。接着我们只要证明对这个式子的求解得到的 p 是多项式时间的即可，我们利用求根公式得到：

$$p=\frac{(N+1-x_1)\pm\sqrt{(N+1-x_1)^2-4N}}{2}. \tag{1.1}$$

接下来对计算该式子的时间进行分析：首先加减法均是线性时间可以计算出来的，其次乘法与除法是多项式时间可以计算的得到的。对于开方运算，我们知道这个结果中求根所得到的结果是有理数，所以我们可以用直接用二分法对其进行计算，可以得到其时间的上界为：$O(\log(N^2)N^2)$，其也是多项式时间可以计算得到的，实际计算的时间经过改进肯定远小于此时间。

因此我们可以得到由于计算中涉及到的所有计算都是多项式时间内可以完成的，所以求出 p 的时间是多项式的，所以求出 q 的时间也是多项式的，因此得证。

Problem 3：

首先介绍下算法的基本思想，首先由于 N 为两个大质数的乘积，所以 2 这个元素一定属于$\mathbb{Z}_N{}^*$，所以根据扩展欧几里得算法可以在多项式时间求得$y=2^{-1} mod N$。

之后我们可以分析$x^e mod N$的组成。首先如果其最低位为 LSB（x），**假设x去掉最低位后得到的值为x'**，则我们可以得到$x^e mod N=(x'*2+LSB(x))^e mod N$。所以我们

在$x^e modN$的值上乘y^e之后 mod N 可以得到:

$$((x^e modN)\cdot y^e)modN=(x^e\cdot y^e)modN=(x'+LSB(x)\cdot y)^e modN \tag{1.2}$$

我们之后的计算可以在这个的基础上进行。当 LSB（x）为 0 时我们直接在$x^e modN$的值上乘y^e可以得到$(x')^e modN$，接着继续进行即可。要是 LSB(x)不为 0，我们得到的值为：$(x'+y)^e modN$，此时没法直接得到$(x')^e modN$，因此无法直接调用题中的函数，我们采用下面的方法:

x 的最右 l 比特与$2x''$-N 的最右 l 比特相同，其中x''是通过$(x\cdot y)modN$计算得到的最右 l-1 比特，这样就得到了长度有关的递归式，下面证明这个结论是对的:

$$\begin{aligned}((2*(x*y)\ mod\ N)\ mod\ 2^{l-1}-N)mod\ 2^l&=((x*y*2)mod\ N-N)\ mod\ 2^l\\&=(x\ *y*2)\ mod\ 2^l\\&=x\ mod\ 2^l\end{aligned}$$

所以我们可以通过递归反复的进行$(x^e\cdot y^e)modN$操作来得到我们想要的最后一位。

Algorithm 1

INPUT N，e，c，l（其中 l 需要计算得到的 x 的位数，从后往前）
Output C 对应的明文
If l = 1 **then**

return $\mathcal{A}(N,c)$

Else

$x_1=A(N,c)$，$y=2^{-1}\ mod\ N$

$x'=$ Algorithm(N，e，$(c\cdot y^e)mod\ N$，l-1)

if $x_1=0$ **then**

return $x'||x_1$

else

return $(2x'-N)\ mod\ 2^l$

end
end

可以发现我们把当 c 的最后一位为 1 时无法计算的情况转换为了算$(c\cdot y^e)mod\ N$的 l-1

位，由于递归到 l 为 1 时我们是可以直接计算的，所以可以通过递归计算，递归到递归终止条件后再从后往前计算$(2x'-N)\ mod\ 2^l$即可。

Problem 4:

首先我们说明两人共享相同的密钥，即 Bob 最终收到的密文是什么：

$$w \oplus t = u \oplus r \oplus t = s \oplus r = k$$

因此 Bob 最终收到的密钥与 Alice 的密钥相同。

但是，这个密钥交换方法Π是不安全的，我们用安全密钥交换实验$KE^{eva}_{\mathcal{A},\Pi}$来分析：构造如下的攻击者$\mathcal{A}$，其利用 Alice 与 Bob 的对话记录 trans 来进行攻击。其首先将对话记录中的所有消息异或，得到了：$s \oplus u \oplus w = k$，因此我们通过二者的对话记录就可以直接得到密钥，所以我们的实验成功的概率为：

$$Pr(KE^{eva}_{\mathcal{A},\Pi}(n)=1)=1-\frac{1}{2}Pr\left(\hat{k}=k\right)=1-\frac{1}{2^{n+1}}>\frac{1}{2}+negl(n)$$

所以该密钥交换方案为不安全的。

Problem 5：

（a）其解密方法可以通过计算$c_1{}^x$看它是否与c_2相等，如果相等则输出 0，否则输出 1。由于其解密有一定概率可能出错，我们验证一下其是否符合公钥加密的定义：计算得$Pr(Dec_{sk}(Enc_{pk}(b))=b)=1-\frac{1}{2^n}$所以其符合公钥加密得定义。

（b）我们将加密方法Π的安全性归约到 DDH 难问题上：具体的我们假设攻击者$\mathcal{A}$来攻击加密方案Π，然后我们构造攻击者$\mathcal{A}'$来攻击 DDH 问题。具体的当进行实验时$\mathcal{A}'$收到$(\mathbb{G},q,g,g^x,g^y,g^w)$其中$w$可以是$g^{xy}$（意味着 b=0）也可以是$g^z$（意味着 b=1）其中 y 与 z 满足：$y,z\xleftarrow[random]{}\mathbb{Z}_q$。且$\mathcal{A}'$平常可以自由的向数据库询问任何一种情况，使其攻击对象产生一组新的$(\mathbb{G},q,g,g^x,g^y,g^w)$值，其中 y 与 z 每次重新生成，**均为随机数**。

接着$\mathcal{A}'$利用收到的消息中的(G,q,g,g^x)作为公钥，攻击者$\mathcal{A}$可以向$\mathcal{A}'$询问，当其想加密 1 时$\mathcal{A}'$向数据库询问后输出$<g^y,g^z>$，加密 0 时$\mathcal{A}'$向数据库询问后输出$<g^y,g^{xy}>$。需要注意的是由于每次$\mathcal{A}'$查询时产生的 y 与 z **均为随机的**，因此引入了随机性使得 cpa 攻击

失效。

加密时$\mathcal{A}'$将输入的消息$<g^y, g^w>$发送给攻击者$\mathcal{A}$。并且当$\mathcal{A}$输出 0 时$\mathcal{A}'$输出 0 ，否则$\mathcal{A}'$输出 1。那么我们有：

$$\begin{aligned}|2 \cdot Pr(PubK^{cpa}_{\mathcal{A},\Pi}(n)=1)| &= |Pr(PubK^{cpa}_{\mathcal{A},\Pi}(n=1)=1\,|b=0) + Pr(PubK^{cpa}_{\mathcal{A},\Pi}(n=1)=1\,|b=1)| \\ &= |1 - Pr(\mathcal{A}'(\mathbb{G},q,g,g^x,g^y,g^{xy})=1) + Pr(\mathcal{A}'(\mathbb{G},q,g,g^x,g^y,g^z)=1)| \\ &= 1 + w(n)\end{aligned}$$

由于 DH 问题是难解的所以有$w(n)$为可忽略的，所以我们有：

$$Pr(PubK^{cpa}_{\mathcal{A},\Pi}(n)=1) = \frac{1+w(n)}{2} \leq \frac{1}{2} + negl(n) \tag{1.3}$$

所以原加密方案为 CPA 安全的。

■

Problem 6：

由于 El Gamal 加密方案中我们如果知道g^r以及公钥$(\mathbb{G},q,g,h)$是无法得到h^r的，因为攻击者无法由g^r求出 r 或者由h求出 x，因此无法用乘方的方法直接求h^r或者是用$g^{r\cdot x}$的方法来求。

因此我们可以直接把h^r作为私钥加密方案的**密钥**，类似的我们发送消息：$<g^r, Enc_{h^r}(m)>, r \xleftarrow[random]{} \mathbb{Z}_q$。而攻击者无法得到密钥$h^r$自然无法对其进行有效攻击，当接收方收到后只要用公钥对应的私钥 x 做$g^{r\cdot x}$的操作即可得到私钥加密的密钥h^r。

综上 $Enc^{hy}: r \leftarrow \mathbb{Z}_q, k \leftarrow h^r, c \leftarrow Enc_k(m)$； $Dec^{hy}: k := Dec_{sk}(h^r) = h^{r\cdot x}, m := Dec'_k(c)$。

Problem 7：

(a) 我认为它是不安全的我们可以选择$m' = m^{-1}\ mod\ N$作为我们对 signing 数据库的输入，我们计算可得$\sigma' = (m^{-1})^d = (m)^{-d}\ mod\ N$，那么因此我们可以伪造原消息的 signature：

$$(\sigma')^{-1} mod\ N\ =\ (m)^d\ mod\ N = \sigma$$

所以这个签名方案不是安全的。接下来分析这么做的正确性：首先$m \in \mathbb{Z}_n{}^*$，这样才能用扩展欧几里得算法求逆，而这是 RSA 的前提，是满足的，同样的由于σ'也属于$\mathbb{Z}_N{}^*$，所以σ'

也是可以求逆的。所以这种做法是可行的，因此此加密方案是不安全的。

(b) 我认为它是安全的。如果不是的话我们假设存在一个攻击者，其可以只根据公钥 e 与 m 就可以得出m^d，那么我们可以知道任何的 RSA 加密方案都可以用它来破解，我们只需要输入公钥 e 以及$c = m^e$它会自动得到$(m^e)^d = m$，而这显然是不可能的。因此这个加密方案是安全的。

Problem 8：

非常容易地我们可以想到只要满足$m_0 \oplus m_1 = 1^l$即可，例如$m_0 = 0^l, m_1 = 1^l$，我们因此得到$\sigma_0 = (x_{1,0}, x_{2,0}, \cdots, x_{l,0}), \sigma_1 = (x_{1,1}, x_{2,1}, \cdots, x_{l,1})$我们可以很容易的构造加密矩阵，见老师上课的 ppt 如下：

$$sk = \begin{pmatrix} x_{1,0} & x_{2,0} & \cdots & x_{\ell,0} \\ x_{1,1} & x_{2,1} & \cdots & x_{\ell,1} \end{pmatrix}$$

所以现在我们可以加密伪造任何消息了，攻击完毕。