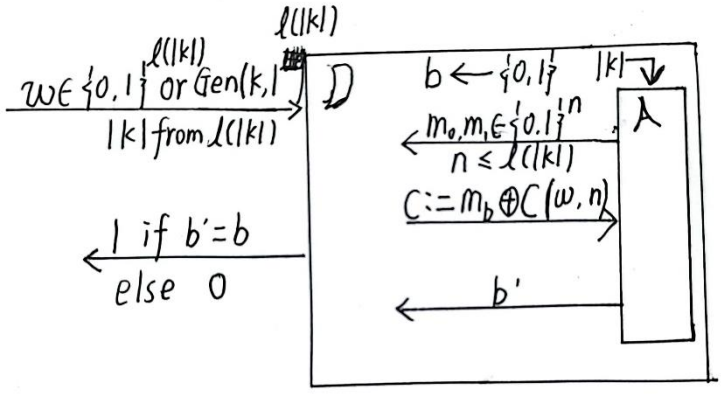
HomeWork2

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：

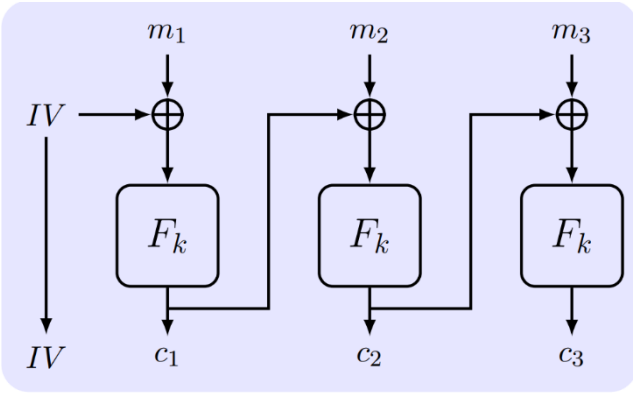
前|f(s)|-1 位一定符合该条件因为其是伪随机生成器的输出。后一位因为其与前面生成的串无关，且本身是随机生成的串的最后一位所以无法用前几位推出。因此 f(s)是为随机生

成器。

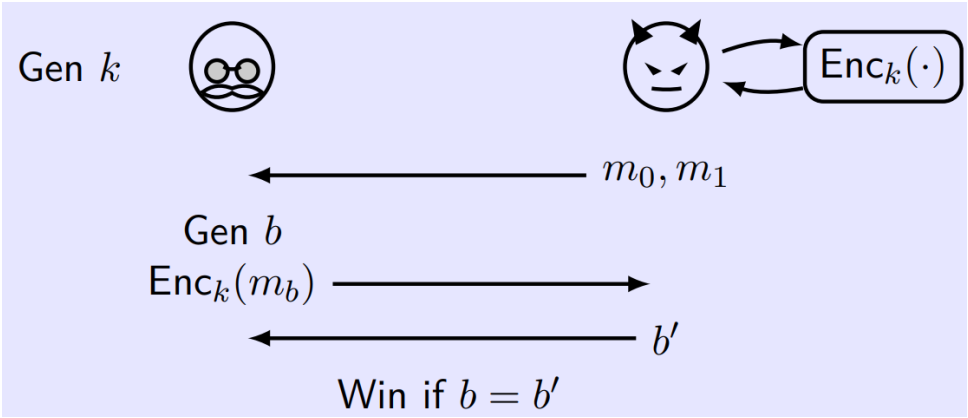
所以 g(s)的最后一位可以被构造为 0。这样 g(s)不符合随机生成器的定义，所以 g(s)不一定为 PRG。

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：

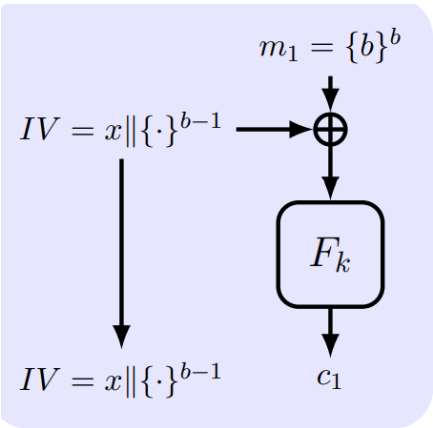
我们可以构造一个攻击者，为了简单假设其只加密两块长的消息 $m_0 m_1$，假设：$m_0 = 0x^{n-1}0y^{n-1}$，$m_1 = 1x^{n-1}1y^{n-1}$，我们假设攻击者受到密文后将其第一位反转后输入解密数据库，，根据定义这是可行的，我们接下来对其进行分类讨论与破解：

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。