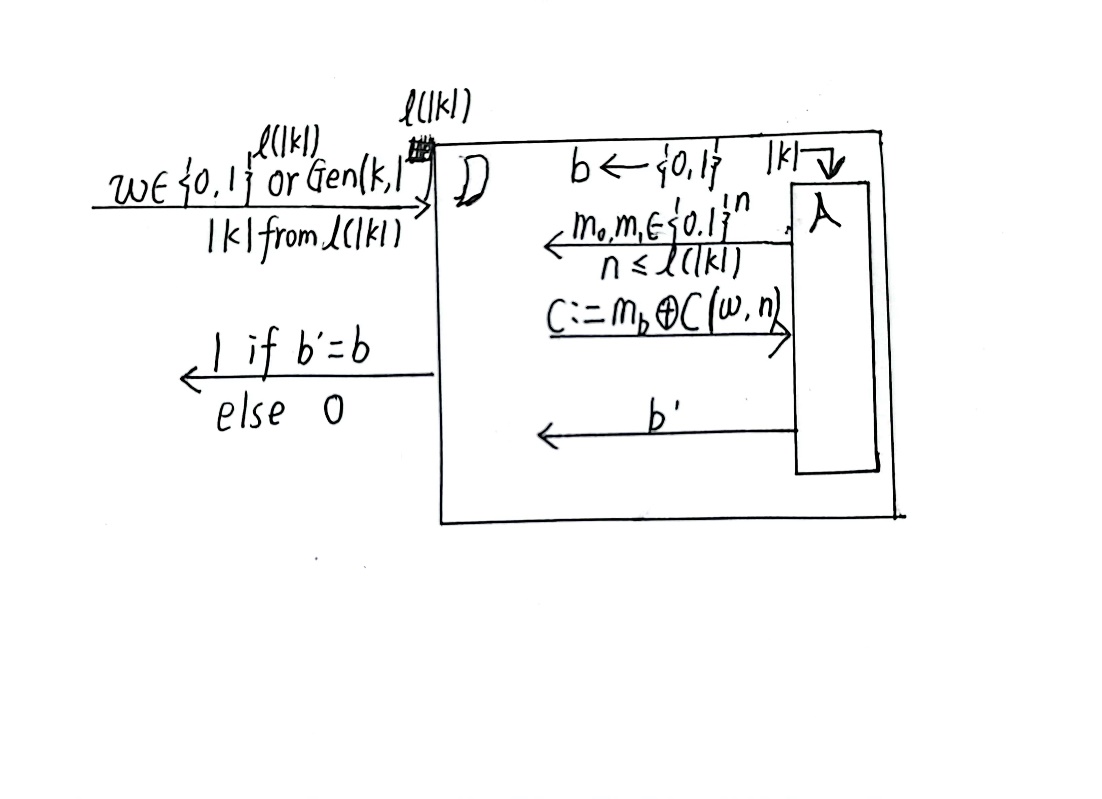
HomeWork2

120L020614-刘昕烨

**Problem 1：**

证明：

我们构造攻击者来攻击加密方案。然后用攻击者来构造区分器来区分生成器G。这样当可以区分时，可以区分生成器G。但既然伪随机生成器存在，那么显然可以得到无法区分，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进行解释：由于变长随机生成器具有性质是的前缀，，所以只要输入长度为的伪随机字符串，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“当可以区分时，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：



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，我们对实验成功的概率作出假设后可以得到：



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：



我们将1.1，1.2，1.3结合可以知道：



所以由定义我们可以知道加密方案为在窃听者存在下的不可区分加密。

**Problem 2：**

1. 当时我觉得不一定是PRG。

因为当这两者有关系时，例如：是每位取反的时候，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。

1. 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(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，我们构造f(s)如下:

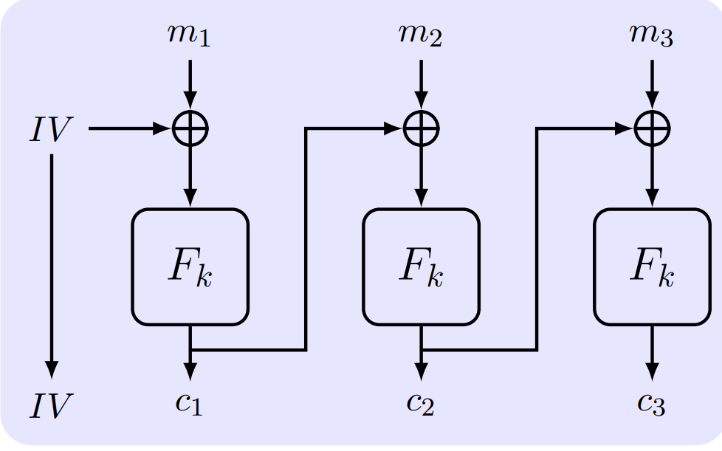


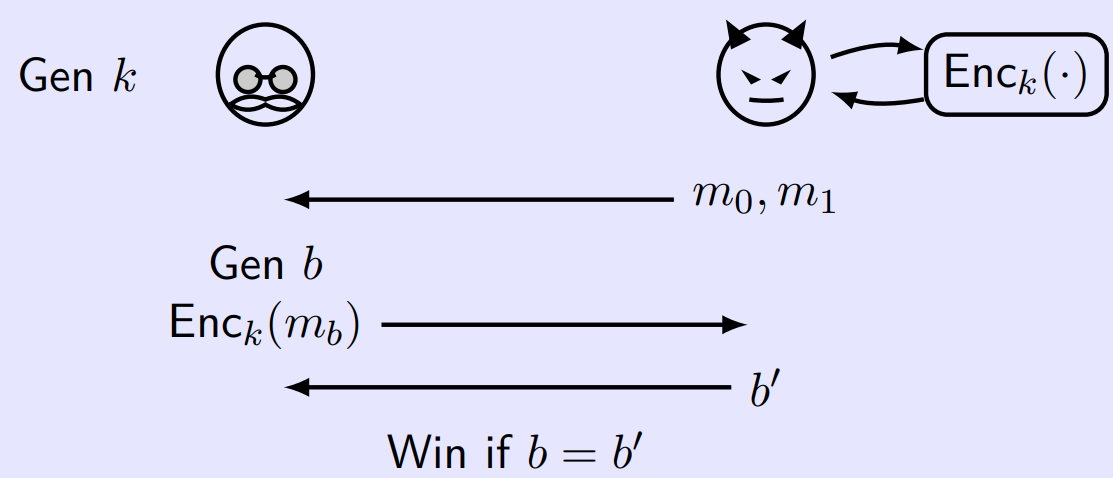
那么f(s)是一个PRG。 我们可以利用生成器是否是为伪随机的与“生成的串能否通过前k位（k大于等于1小于n）的值预测k+1位的值”等价这一定理（Yao‘82）来说明这一事实：

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，并且得到我们把返回的密文记作。我们接着构造新的消息，那么如果选择加密就会得到，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，



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：



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：



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



接着我们构造A来攻击伪随机函数，然后我们将串输入给它，其中O为f与F中的一个，显然



并且由于F是为随机函数所以是可以忽略的。

**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，那么与之前类似，这次只叙述公式的变化：





为除了以外的任何值均可，此时所以其不具有CPA安全。

**Problem 6：**

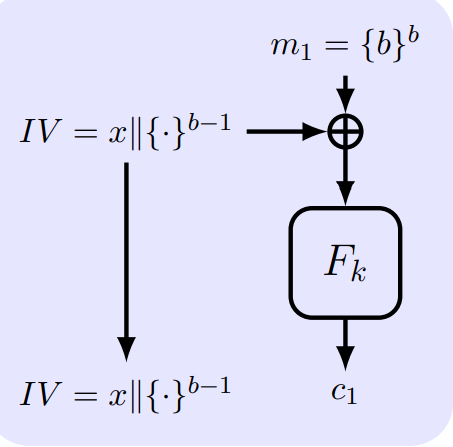
我们可以构造一个攻击者，为了简单假设其只加密两块长的消息，假设：， ，我们假设攻击者受到密文后将其第一位反转后输入解密数据库，，根据定义这是可行的，我们接下来对其进行分类讨论与破解：

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，我们观察收到的解密值的第一位，若其为1则说明其为我们输出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其中表示原文第二比特，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。