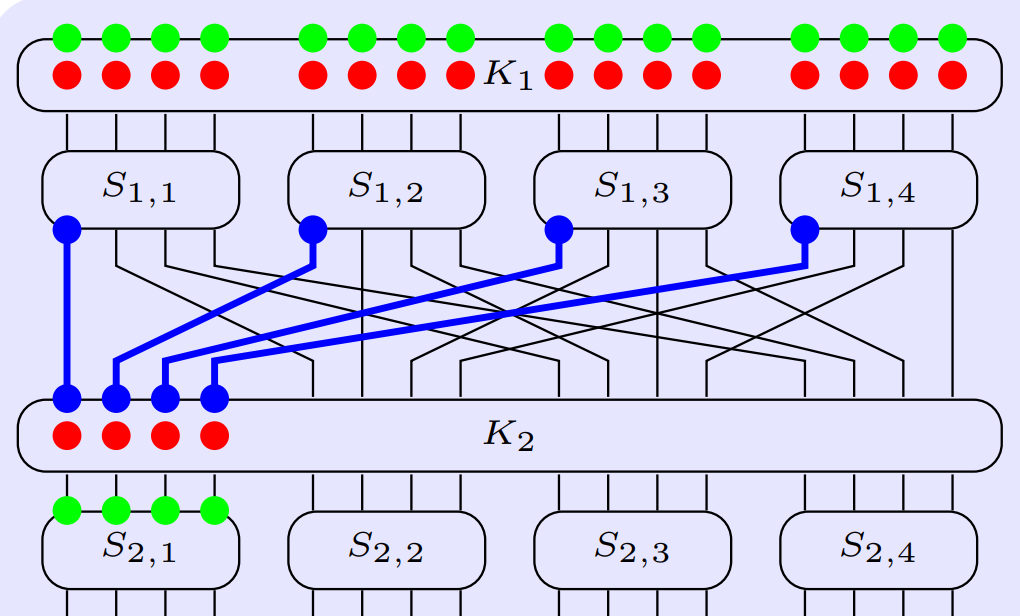
HomeWork3

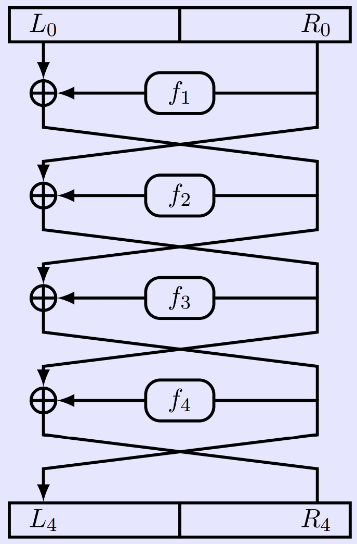
120L020614-刘昕烨

**Problem 1：**

1. 此时我们先猜测8+8\*8=72位的密钥，其中8位属于另外的64位属于。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。并且由于我们要遍历所有的密钥，所以遍历的次数为。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（）。此时我们的复杂度为。
2. 同样的我们先猜测72位的密钥，其中8位属于另外的64位属于，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。
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，且）。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：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，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（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：，S不变，B增加百分之a变化率为：。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轮的密钥是原来的第i轮密钥的按位取反即。

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，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。

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，观察其构造我们容易得出：。

首先我们有：，。接下来我们分析当x与k反转后得到的 与原来的有何区别：





并且我们可以由异或的性质得到以下的公式：



所以有



假设对于第i轮我们的假设，成立，我们对第i+1轮进行推理：





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，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：





**Problem 3：**

我认为这个函数不是单向函数，因为加法给定一个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：



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。假设均由单向函数构造而来，且均定义如下：。构造如下的函数f：



其中均为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位作为第385到512位作为然后返回。实验每次都会成功，所以以作为参数的函数f不一定是可逆的。



**Problem 5：**

第一问我认为不一定是单向函数，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：。且有，。

构造f如下：



我们很容易知道的值恒为为长度为2n的0串，因此不是单向函数。

下证f是单向函数：

我们假设，下面我们构造攻击函数h，函数h向输入(）、然后将(）输入给，并把的输出的前n位直接当作的输出。我们可以得到：



由于我们有：



化简得：



及：



由于h是单向函数，且可忽略，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，其中：，。假设有一个对其的攻击者，我们有一个对单向函数f的攻击者利用进行攻击。

首先收到f(x)后计算f(f(x))然后将传给，然后将的输出直接输出，那么我们可以得到：



又因为f是单向函数所以我们得到g也是单向函数。

