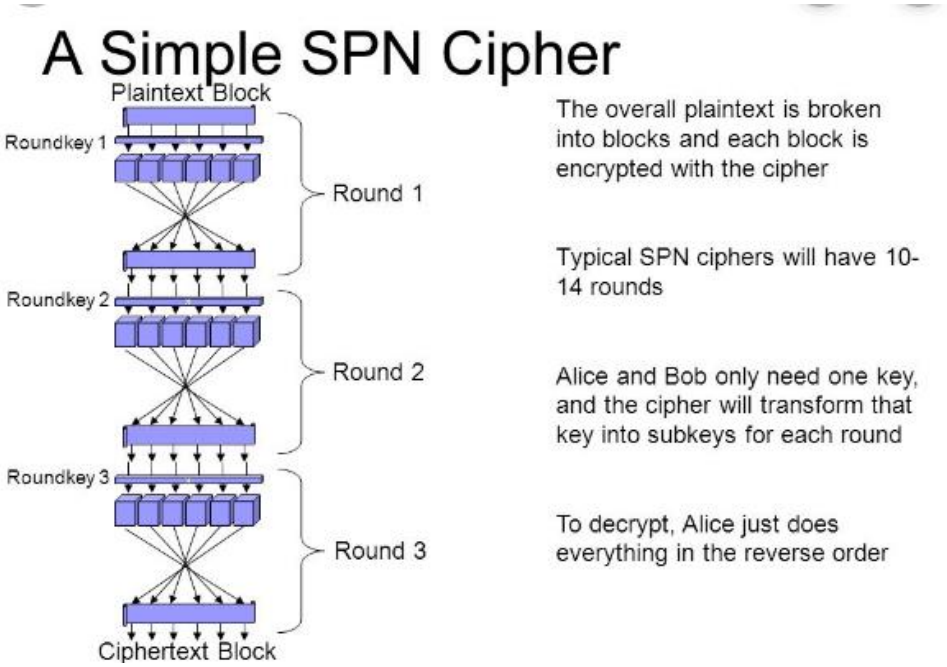
应用密码学作业三

龚彦韬 1901210635

一、Answer why a final key mixing is required by a cipher



异或操作被称为 key mixing 即轮密钥混合。SPN 的第一个和最后一个操作都是异或轮密钥，叫做白化。如果一个攻击者不知道密钥的话，将无法开始一个加密或解密操作。并且如果不进行最后的轮密钥的话，很容易根据输出结果分析出最后四个 S 盒的构造。

二、Compute the DDT and LAT tables of ZUC S0 and S1

查阅资料可知，ZUC 加密算法的密码盒 S0 和 S1 为 8 比特输入，所以其映射关系为 256 种。

```
Sboxes = [ [0x3E, 0x72, 0x5B, 0x47, 0xCA, 0xE0, 0x00, 0x33, 0x04, 0xD1, 0x54, 0x98, 0x09, 0xB9, 0x6D, 0xCB,
           0x7B, 0x1B, 0xF9, 0x32, 0xAF, 0x9D, 0x6A, 0xA5, 0xB8, 0x2D, 0xFC, 0x1D, 0x08, 0x53, 0x03, 0x90,
           0x4D, 0x4E, 0x84, 0x99, 0xE4, 0xCE, 0xD9, 0x91, 0xDD, 0xB6, 0x85, 0x48, 0x8B, 0x29, 0x6E, 0xAC,
           0xCD, 0xC1, 0xF8, 0x1E, 0x73, 0x43, 0x69, 0xC6, 0xB5, 0xBD, 0xFD, 0x39, 0x63, 0x20, 0xD4, 0x38,
           0x76, 0x7D, 0xB2, 0xA7, 0xCF, 0xED, 0x57, 0xC5, 0xF3, 0x2C, 0xBB, 0x14, 0x21, 0x06, 0x55, 0x9B,
           0xE3, 0xEF, 0x5E, 0x31, 0x4F, 0x7F, 0x5A, 0xA4, 0x0D, 0x82, 0x51, 0x49, 0x5F, 0xBA, 0x58, 0x1C,
           0x4A, 0x16, 0xD5, 0x17, 0xA8, 0x92, 0x24, 0x1F, 0x8C, 0xFF, 0xD8, 0xAE, 0x2E, 0x01, 0xD3, 0xAD,
           0x3B, 0x4B, 0xDA, 0x46, 0xEB, 0xC9, 0xDE, 0x9A, 0x8F, 0x87, 0xD7, 0x3A, 0x80, 0x6F, 0x2F, 0xC8,
           0xB1, 0xB4, 0x37, 0xF7, 0x0A, 0x22, 0x13, 0x28, 0x7C, 0xCC, 0x3C, 0x89, 0xC7, 0xC3, 0x96, 0x56,
           0x07, 0xBF, 0x7E, 0xF0, 0x0B, 0x2B, 0x97, 0x52, 0x35, 0x41, 0x79, 0x61, 0xA6, 0x4C, 0x10, 0xFE,
           0xBC, 0x26, 0x95, 0x88, 0x8A, 0xB0, 0xA3, 0xFB, 0xC0, 0x18, 0x94, 0xF2, 0xE1, 0xE5, 0xE9, 0x5D,
           0xD0, 0xDC, 0x11, 0x66, 0x64, 0x5C, 0xEC, 0x59, 0x42, 0x75, 0x12, 0xF5, 0x74, 0x9C, 0xAA, 0x23,
           0x0E, 0x86, 0xAB, 0xBE, 0x2A, 0x02, 0xE7, 0x67, 0xE6, 0x44, 0xA2, 0x6C, 0xC2, 0x93, 0x9F, 0xF1,
           0xF6, 0xFA, 0x36, 0xD2, 0x50, 0x68, 0x9E, 0x62, 0x71, 0x15, 0x3D, 0xD6, 0x40, 0xC4, 0xE2, 0x0F,
           0x8E, 0x83, 0x77, 0x6B, 0x25, 0x05, 0x3F, 0x0C, 0x30, 0xEA, 0x70, 0xB7, 0xA1, 0xE8, 0xA9, 0x65,
           0x8D, 0x27, 0x1A, 0xDB, 0x81, 0xB3, 0xA0, 0xF4, 0x45, 0x7A, 0x19, 0xDF, 0xEE, 0x78, 0x34, 0x60 ],
          [0x55, 0xC2, 0x63, 0x71, 0x3B, 0xC8, 0x47, 0x86, 0x9F, 0x3C, 0xDA, 0x5B, 0x29, 0xAA, 0xFD, 0x77,
           0x8C, 0xC5, 0x94, 0x0C, 0xA6, 0x1A, 0x13, 0x00, 0xE3, 0xA8, 0x16, 0x72, 0x40, 0xF9, 0xF8, 0x42,
           0x44, 0x26, 0x68, 0x96, 0x81, 0xD9, 0x45, 0x3E, 0x10, 0x76, 0xC6, 0xA7, 0x8B, 0x39, 0x43, 0xE1,
           0x3A, 0xB5, 0x56, 0x2A, 0xC0, 0x6D, 0xB3, 0x05, 0x22, 0x66, 0xBF, 0xDC, 0x0B, 0xFA, 0x62, 0x48,
           0xDD, 0x20, 0x11, 0x06, 0x36, 0xC9, 0xC1, 0xCF, 0xF6, 0x27, 0x52, 0xBB, 0x69, 0xF5, 0xD4, 0x87,
           0x7F, 0x84, 0x4C, 0xD2, 0x9C, 0x57, 0xA4, 0xBC, 0x4F, 0x9A, 0xDF, 0xFE, 0xD6, 0x8D, 0x7A, 0xEB,
           0x2B, 0x53, 0xD8, 0x5C, 0xA1, 0x14, 0x17, 0xFB, 0x23, 0xD5, 0x7D, 0x30, 0x67, 0x73, 0x08, 0x09,
           0xEE, 0xB7, 0x70, 0x3F, 0x61, 0xB2, 0x19, 0x8E, 0x4E, 0xE5, 0x4B, 0x93, 0x8F, 0x5D, 0xDB, 0xA9,
           0xAD, 0xF1, 0xAE, 0x2E, 0xCB, 0x0D, 0xFC, 0xF4, 0x2D, 0x46, 0x6E, 0x1D, 0x97, 0xE8, 0xD1, 0xE9,
           0x4D, 0x37, 0xA5, 0x75, 0x5E, 0x83, 0x9E, 0xAB, 0x82, 0x9D, 0xB9, 0x1C, 0xE0, 0xCD, 0x49, 0x89,
           0x01, 0xB6, 0xBD, 0x58, 0x24, 0xA2, 0x5F, 0x38, 0x78, 0x99, 0x15, 0x90, 0x50, 0xB8, 0x95, 0xE4,
           0xD0, 0x91, 0xC7, 0xCE, 0xED, 0x0F, 0xB4, 0x6F, 0xA0, 0xCC, 0xF0, 0x02, 0x4A, 0x79, 0xC3, 0xDE,
           0xA3, 0xEF, 0xEA, 0x51, 0xE6, 0x6B, 0x18, 0xEC, 0x1B, 0x2C, 0x80, 0xF7, 0x74, 0xE7, 0xFF, 0x21,
           0x5A, 0x6A, 0x54, 0x1E, 0x41, 0x31, 0x92, 0x35, 0xC4, 0x33, 0x07, 0x0A, 0xBA, 0x7E, 0x0E, 0x34,
           0x88, 0xB1, 0x98, 0x7C, 0xF3, 0x3D, 0x60, 0x6C, 0x7B, 0xCA, 0xD3, 0x1F, 0x32, 0x65, 0x04, 0x28,
           0x64, 0xBE, 0x85, 0x9B, 0x2F, 0x59, 0x8A, 0xD7, 0xB0, 0x25, 0xAC, 0xAF, 0x12, 0x03, 0xE2, 0xF2 ] ]
```

DDT

分别对两个 S 盒进行操作，并把结果写入到 excel 表格中。

```
def printAllDDT( SBoxes, typ ):
    # create excel
    wbk = xlwt.Workbook()
    for i in range(0,2):
        print "#### Differential Distribution Table of " + typ + "SBox-" + str(i) + " ####"
        # create 2 DDT excel table's tittle
        sheet = wbk.add_sheet('sheet %d' %(i+1))
        sheet = printDDT(DDT(SBoxes, i), sheet)
        print "\n"
    wbk.save('DDT.xls')
```

关键函数。首先创建 256*256 的二维矩阵，遍历 Δx 的取值（在十进制下为 0-255），生成输入比特对 x' 、x''，使得 $x' \oplus x'' = \Delta x$，将生成的比特对分别通过 S 盒置换，获得 y' 、y''，对它们进行异或计算得到 Δy。在二维矩阵中，Δx 对应出现 Δy 的计数加一，获得最终的 DDT 表。

```
def DDT(SBoxes, SBoxNumber):
    DDT = createEmpty2DList(256)
    differentialUniformity = 0
    for x in range(0,256):
        pairs = generatePairsWithDifference(x)
        for pair in pairs:
            diff = outputDifferenceOfPair(pair, SBoxes, SBoxNumber)
            DDT[x][diff] += 1
            if x != 0 and DDT[x][diff] > differentialUniformity:
                differentialUniformity = DDT[x][diff]
    print "\nDifferential Uniformity of SBox" + str(SBoxNumber) + " : " + str(differentialUniformity)
    return DDT
```

LAT

分别对两个 S 盒进行操作，并把结果写入到 excel 表格中。

```
def LAT(sBoxes):
    # create excel
    wbk = xlwt.Workbook()

    i = 0
    while i < len(sBoxes):
        print "######## LAT of SBox-" + str(i) + " ########"

        # create 2 LAT excel table's tittle
        sheet = wbk.add_sheet('sheet %d' %(i+1))

        sheet = LATofSBOX( sBoxes[i] ,sheet)
        i += 1

    wbk.save('LAT.xls')
```

关键函数。对于每一个 8 比特输入x，从 S 盒中找到对应的y，x每一比特对应$x_1 \cdots x_8$，同理y每一比特对应$y_1 \cdots y_8$。$x_1 \cdots x_8$排列组合异或，$y_1 \cdots y_8$排列组合异或，将两者的结果再异或，并累加为 1 的结果，减去 128 取反，作为最终结果记录到 LAT 表中。

```
def LATofSBOX( sBox ,sheet):
    lat = []
    for i in range(0,256):
        for j in range(0,256):
            r = 0
            for x in range(0,256):
                sx = sBox[x]
                r += (mask(i,x) ^ mask(j,sx))
            r = -(r - 128)
            lat.append(r)

            sheet.write(i, j, r)
        print i
    printLAT(lat)
    return sheet
```