# Cryptography: Problem Set #1

Due on Monday, June 2nd at 10:00pm

*Elena Machkasova*

**Ellis Weglewski**

# Problem 1

**(3.1) Question:**
Show that $S_1(x_1) \oplus S_1(x_2) \neq S_1(x_1 \oplus x_2)$ for:

1. $x_1 = 000000, x_2 = 000001$
2. $x_1 = 111111, x_2 = 100000$
3. $x_1 = 101010, x_2 = 010101$

**Solutions:**

1. (a) $(S_1((x_1 = 0, 0 = 14) = 1110 \oplus S_1((x_2 = 0, 1 = 00 = 0000)) = 1110$
   (b) $S_1(000000 \oplus 0000001) = S_1(000001) = 0000$
   (c) $0000 \neq 1110$
2. (a) $(S_1(x_1 = 15, 3 = 13) = 1101 \oplus S_1(x_2 = 0, 2 = 04) = 0100) = 1001$
   (b) $S_1(111111 \oplus 100000) = S_1(011111) = 1000$
   (c) $1001 \neq 1000$
3. (a) $(S_1(x_1 = 5, 2 = 06) = 0110 \oplus S_1(x_2 = 10, 1 = 12) = 1100) = 1010$
   (b) $S_1(101010 \oplus 010101) = S_1(111111) = 1101$
   (c) $1010 \neq 1101$

# Problem 2

**(3.2) Question:**
We want to verify that $IP(\cdot)$ and $IP^{-1}(\cdot)$ are truly inverse operations. We consider a vector $x = (x_1, x_2, ..., x_{64})$ of 64 bit. Show that $IP^{-1}(IP(x)) = x$ for the first five bits of $x$,i.e. for $x_i = 1, 2, 3, 4, 5$.
**Solution:**
Via pg. 70: $IP(Y) = IP(IP^{-1}(R_{16}L_{16}))$. I take this to imply: $IP^{-1}(Y) = IP^{-1}(IP(R_{16}L_{16}))$. If we look at $IP$ and $IP^{-1}$ boxes on pg 62, we can see where each byte is sent after each operation. The byte in position 1 is sent to position 40 after $IP$ and the byte in position 40 is sent to position 1 after $IP^{-1}$. This means that $IP^{-1}$ is undoing what $IP$ did which makes them mutually inverse. We can construct a flow chart to illustrate where each byte goes in each step of the operation $IP^{-1}(IP(x)) = x$:

1. $x_1 \rightarrow x_{40} \rightarrow x_1$
2. $x_2 \rightarrow x_8 \rightarrow x_2$
3. $x_3 \rightarrow x_{48} \rightarrow x_3$
4. $x_4 \rightarrow x_{16} \rightarrow x_4$
5. $x_5 \rightarrow x_{56} \rightarrow x_5$

# Problem 3

**(3.3) Question:**
What is the output of the first round of the DES algorithm when the plaintext and the key are both all zeros?
**Solution:**
If the plaintext is all zeros than all the permutations and expansions will be composed of entirely zeros as well. As far as the key and its subkeys go, they will all be zeros too. This leaves us only needing to plug zero (000000) into each S-Box, permute the result, and then XOR with the 32 zeros on the left:

1. $S_1(000000) = 14 = 1110$
2. $S_2(000000) = 15 = 1111$
3. $S_3(000000) = 10 = 1010$
4. $S_4(000000) = 07 = 0111$
5. $S_5(000000) = 02 = 0010$
6. $S_6(000000) = 12 = 1100$
7. $S_7(000000) = 04 = 0100$
8. $S_8(000000) = 13 = 1101$

The output of the S-Boxes is: 11101111101001110010110001001101 which still needs to be permuted and XOR'd. Through consulting the permutation table on page 66, our new right side is 11011000110110001101101110111100. XOR with all zeros would give us the same thing so that is our answer.