CIS 3362 Quiz #3: Bitwise Operators, DES, AES Solution

1) (5 pts) Consider a DES-like cipher that operates on 16 bit blocks with the following IP matrix:

$$\begin{bmatrix} 3 & 9 & 15 & 7 \\ 13 & 11 & 6 & 5 \\ 16 & 2 & 14 & 10 \\ 8 & 4 & 1 & 12 \end{bmatrix}$$

What is the corresponding matrix IP^{-1}? Please fill in the blanks below. You may use the space below the blanks for any "work".

$$\begin{bmatrix} 15 & 10 & 1 & 14 \\ 8 & 7 & 4 & 13 \\ 2 & 12 & 6 & 16 \\ 5 & 11 & 3 & 9 \end{bmatrix}$$

We are supposed to take the third bit from the input and place it 1st. This means that to invert this action, we would want to take the first bit in the output answer and place it back in position 3. Thus in IP^{-1}, position 3 should store 1, position 9 should store 2, and so forth. What is above is what you get when you follow this pattern.

Grading: 5 pts – if all correct, 4 pts – if 14-15 items correct, 3 pts – if 10-13 correct
2 pts – if 6-9 correct, 1 pt – if 4 or 5 correct

2) (8 pts) Provide the output for the designated inputs for each of the four S-boxes described below. Please give your answers as **4 binary bits.** (Each answer is worth 2 pts, no partial credit, so carefully make sure you are **using the correct S-box** and look up the correct row and column.)

(a) $S_1(110100)$ = **1001** (row 10 = 2, col 1010 = 10, entry is 9.)

(b) $S_2(000111)$ = **0111** (row 01 = 1, col 0011 = 3, entry is 7.)

(c) $S_5(111011)$ = **0100** (row 11 = 3, col 1101 = 13, entry is 4.)

(d) $S_8(001000)$ = **0110** (row 00 = 0, col 0100 = 4, entry is 6.)

Grading: 2 pts if correct, 1 pt if correct row and column is clearly indicated but answer is wrong, 0 otherwise, give full credit if they put 9, 7, 4, 6.

3) (10 pts) Consider the task of writing a function (**in C**) that takes in an array of integers(**values**), its length(**len**), and prints out the bit positions where an odd number of the integers in the array have that bit position set to 1. For example, if the array stored [20, 17, 6, 9, 16], your function should print 1, 3 and 4 because only 6 has bit position 1 on (value 2^1) only 9 has bit position 3 on (value 2^3) and 20, 17 and 16 have bit position 4 on. Note: an integer has 32 bits, so your function should only print values in between 0 and 31, inclusive. To help you, here is the example in binary with the bit values highlighted:

```
10100 (20)
10001 (17)
00110 (6)
01001 (9)
10000 (16)
```

Fill in the function prototype given below.

```
void printOddBits_v1(int* values, int len) {
    int bits = 0;                                   // 1 pt
    for (int i=0; i<len; i++)                       // 1 pt
        bits ^= values[i];                          // 4 pts
    for (int i=0; i<32; i++)                        // 1 pt
        if (bits&(1<<i)) printf("%d ", i);          // 3 pts
    printf("\n");                                   // 0 pts
}

void printOddBits_v2(int* values, int len) {
    for (int i=0; i<32; i++) {                      // 1 pt
        int cnt = 0;                                // 1 pt
        for (int j=0; j<len; j++)                   // 1 pt
            if ( (values[j] & (1<<i)) != 0)         // 4 pts
                cnt++;                              // 1 pt
        if (cnt&1) printf("%d ", i);                // 2 pts
    }
    printf("\n");                                   // 0 pts
}
```

4) (4 pts) Let the state matrix to AES right before the SubBytes step be the matrix shown below. Show the state of the matrix right AFTER the SubBytes step:

01	23	45	67
FE	DC	BA	98
0F	1E	2D	3C
4B	5A	69	78

7C	26	6E	85
BB	86	F4	46
76	72	D8	EB
B3	BE	F9	BC

Grading: all correct = 4, 12-15 correct = 3, 8-11 correct = 2, 4-7 correct = 1

5) (8 pts) Consider the DES Key Schedule algorithm. Imagine that in the middle of the algorithm, we have the following values for C_8 and D_8, in binary:

C_8 = 0000 0101 1010 1100 1001 0110 0011
D_8 = 1011 0111 1110 0010 0100 1101 1111

Please show your work and determine the **last** 16 bits of K_9, the round 9 key. (Note: since random luck will get 8 bits correct, the scoring on this question will simply be max(0, correctbits-8).)

First, on both buffers, we must perform a left cyclic shift for round 9, which is 1 bit (see "Schedule of Left Shifts" on the DES Key Schedule Calculation handout.) Performing this we get:

C_9 = 000 0101 1010 1100 1001 0110 0011 0
D_9 = 011 0111 1110 0010 0100 1101 1111 1

For ease of reading, I've placed the one bit that got moved to the right end separately and kept the rest of the groups together. Note that for each grouping, the last bit position is equivalent to 3 mod 4 (except for the last bit).

Now, we want the **last 16 bits**. Thus, we look at the last 16 entries in PC-2, which are:

51, 45, 33, 48, 44, 49, 39, 56, 34, 53, 46, 42, 50, 36, 29 and 32.

These bit positions are all in D_9. Here is what we get when we grab the bits in those positions (I've highlighted the bits in yellow, green, blue and gray for each group of four):

1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0

Grading: max(0, correctbits-8) is score

6) (4 pts) Consider the process of AES Key Expansion. Imagine that we have:

w[30] = D4 69 C2 0A (in hex)
w[33] = 1B 37 E5 8F (in hex)

Calculate w[34] and express your answer as **8 HEX characters.**

Since 34 isn't divisible by 4, we just XOR the two buffers shown above. Use the Hex XOR chart to get each Hex character quickly!

C F 5 E 2 7 8 5

Grading: ½ per item, please round down (so 7 correct is 3, 5 correct is 2, etc.)

7) (10 pts) Let the state matrix to AES right before the MixCols step be the matrix shown below. What is the value of the entry in **row 4, column 2,** right AFTER the MixCols step? Express your answer as **2 HEX characters. (Note:** Max grade for the answer in row 2 column 4 is 3 points, so please be careful to make sure you calculate the appropriate item.**)**

32	97	8D	19
A4	AC	5E	E3
B9	3B	21	46
47	D6	D7	6F

The desired sum of products (in the field FG(2^8)) is 03x97+01xAC+01x3B+02xD6. Let's work out the first and last products:

03 x 97 = 02 x 97 + 01 x 97 = 35 + 97 = **A2** (work below)

02 x 97 = 100101110 mod m(x)

```
        =  00101110
          ^   11011
          ---------
           00110101 = 35 (in hex)
```

02 x D6 = 110101100 mod m(x)

```
        =  10101100
          ^   11011
          ---------
           10110111 = B7 in hex
```

Final result = (A2 + AC) + (3B + B7) = 0E + 8C = **82**

Grading: 1 pt each mult by 1, 2 pts mult by 2, 3 pts mult by 3, 3 pts final XOR

8) (1 pt) Today is National Dessert Day! To celebrate this joyous occasion, what should people be eating today?

Dessert (Give to all)