CIS 3362 Homework #4: DES, AES Solution

1) Consider creating a DES-like block cipher with a block size of 16. Assume that the cipher has a fixed matrix IP, just like DES, that operates exactly as DES's IP matrix. Given the IP matrix shown below, calculate the corresponding matrix, IP^{-1}.

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

$IP^{-1} =$

7	12	1	14
9	16	4	3
6	15	8	2
11	10	13	5

2) The input to the expansion matrix E in DES, expressed in hexadecimal is A39047FC. What are the first 24 bits of output? Please express your answer in bits and put a little space between each group of **6** bits. Use words to explain how you arrived at the answer so that the grader can verify that the answer is correct **for the right reason.**

Convert A39047FC to binary: 1010 0011 1001 0000 0100 0111 1111 1100

The bits we need to grab are

32, 1, 2, 3, 4, 5,
4, 5, 6, 7,8, 9,
8, 9, 10, 11, 12, 13,
12, 13, 14, 15, 16, 17

Doing so, we get the following:

010100 **000111** **110010** **100000**

Note: The sequential nature of E makes this faster to do by hand, since you can just read off several bits in a row

3) Consider a portion of a single DES round where the ***last*** 24 bits of input (expressed in HEX) to S boxes S_5, S_6, S_7, S_8 is 70D6B9. What are the 16 bits of output from those 4 S-boxes? Express your result in binary. (Note: Partial credit on this question will be limited for obvious reasons, so double check your answers.)

70D6B9 converted to binary is 0111 0000 1101 0110 1011 1001
Grouped by 6 bits a piece we have: 011100 001101 011010 111001

S_5(011100) = entry in row 00, column 1110, so row 0, col 14 in S_5 = **14 (1110)**
S_6(001101) = entry in row 01, column 0110, so row 1, col 6 in S_6 = **9 (1001)**
S_7(011010) = entry in row 00, column 1101, so row 0, col 13 in S_7 = **10 (1010)**
S_8(111001) = entry in row 11, column 1100, so row 3, col 12 in S_8 = **3 (0011)**

1110 1001 1010 0011

4) Consider calculating the round 16 key in DES. Given that the input key, with odd parity bits, when described in HEX is **"A197 47F2 E9C1 B35D"**, **determine the first 10 bits of the round 16 key.** (Note: While it looks like an inordinate amount of calculation is necessary here, it turns out that a particular observation that I might off-hand in class makes this calculation very quick. I would strongly not recommend trying to "brute force" this question. Rather, I'd recommend only doing it if you make the observation that allows you to do this question in a minute or so.) **Credit will only be given if appropriate work is shown since it's easy to randomly put bits down and get around half of them. I'll decide what appropriate work is!!!**

When we look at the schedule of left shifts, we see that by round 16, we've shifted the bits left a total of 28 bits. Note that the buffer in which we are doing the shifts is 28 bits long, which means all the bits return to their original location!!!

So, to calculate WHERE in the original key one needs to grab the round 16 key, all you have to do is go to PC-2, grab the number there, and then look at that location in PC-1. This then gives you the bit location **in the original key**. So, let's first calculate the following values for i = 1 to 10: PC-1[PC-2[i]]

Here is the first calculation in detail: PC-1[PC-2[1]] = PC-1[14] = 18

Here is the sequence of 10 values obtained: 18, 59, 42, 3, 57, 25, 41, 36, 10, 17.
Go to the key given above, and find each of these bits to obtain:

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

Specifically, these are the (2nd bit 4, 3rd bit 5, 2nd bit C, 3rd bit A, 1st bit 5, 1st bit F, 1st bit C, 4th bit E, 2nd bit 9, and 1st bit 4)

5) In the past I offered a DES challenge where I gave students 20 of the 56 bits, so that they would have to brute force the remaining 36 bits. Consider a student X, who is approaching the assignment and has two options for tackling it:

(a) Use the professor's slow Java implementation (so no time to code the DES portion of the code) and add the brute force mechanics (say this takes 1 hour), and then check each key at a rate of 200,000 keys per second.

(b) Rewrite the professor's code in C++ and write the corresponding brute force mechanics (takes 2 days/48 hours, including adding the brute force mechanics), and then check each key at a rate of 2,000,000 keys per second.

How much time (please answer in days/hours/minutes/seconds) would each option take? (Please use a calculator to do the arithmetic but set up the fraction on paper.) What option should the student choose?

(a) Total number of keys to try is 2^{36}. Number of seconds to try them is $\frac{2^{36}}{2\times10^5}$. There are 24 x 60 x 60 = 86,400 seconds in a day, so if we convert to days we get $\frac{2^{36}}{2\times10^5\times86400}\sim3.976821570$. To complete the conversion, let's convert $.976821570$ to hours, minutes and seconds (by multiplying this by 24 to get complete hours, then taking the leftover part of that and multiplying by 60 to get minutes, and repeating the process to get seconds);

The end result is 3 days, 23 hours, 26 minutes, 37 seconds.

Adding the one hour to write the brute force mechanics, we get:

4 days, 0 hours, 26 minutes and 37 seconds.

(b) Do the same math, but add 2 days, and change the 2 x 10^5 to 2 x 10^6:

Days for code to run: $\frac{2^{36}}{2\times10^6\times86400}\sim.3976821570$. Convert this to hours, minutes and seconds using the method previously described:

This run time is 9 hours, 32 minutes and 40 seconds.

Thus, this strategy would take a total of:

2 days, 9 hours, 32 minutes and 40 seconds.

The student should choose the second option! (Sorry no smart aleck responses about spending less time on the assignment will be accepted…)

6) 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:

5F	02	37	1A
B4	4B	C6	7E
83	29	D8	95
F1	EC	6D	A0

CF	**77**	**9A**	**A2**
8D	**B3**	**B4**	**F3**
EC	**A5**	**61**	**2A**
A1	**CE**	**3C**	**E0**

7) Let the state matrix to AES right before the ShiftRows step be your answer from problem 7. Show the state of the matrix right AFTER the ShiftRows step. (This will be graded solely based on what your answer to problem 7 was. You can get this one correct even if you got problem 7 incorrect.)

CF	**77**	**9A**	**A2**
B3	**B4**	**F3**	**8D**
61	**2A**	**EC**	**A5**
E0	**A1**	**CE**	**3C**

8) Consider the process of AES Key Expansion. Imagine that we have:

w[24] = 01 32 45 76 (in hex)
w[27] = 89 BA CD FE (in hex)

Calculate w[28], showing each of the following intermediate results: RotWord(temp), SubWord(RotWord(temp)), Rcon[i/4], and the result of the XOR with Rcon[i/4]

No work needs to be shown for the first set of steps. In the first one, we just do a one byte cyclic left rotation. In the second we look up each of the four bytes in the SubBytes box. Rcon is given in the documentation as well. The following XOR is easy to mentally since only one hex character changes. The only work I need to "show" is the last XOR:

```
B4 BD BB A7
01 32 45 76 (xor), use HEX XOR chart!
----------------
B5 8F FE D1
```

RotWord	SubWord	Rcon[i/4]	XOR	FinalResult
BA CD FE 89	F4 BD BB A7	40 00 00 00	B4 BD BB A7	B5 8F FE D1

9) In class we discussed multiplication in the AES field GF(2^8) with the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$. Based on this discussion, derive the answer for the calculation of 07 x 26. Display your final result with two hexadecimal characters.

In binary (which really represents polynomials), we have:

111 x 0010 0110 =

```
1    x 0010 0110 xor
10   x 0010 0110 xor
100  x 0010 0110
```

Solve for each product separately, noting that each is just a bit shift and then XOR:

```
1    x 0010 0110 = 0010 0110
10   x 0010 0110 = 0100 1100
100  x 0010 0110 = 1001 1000
                   -------------
                   1111 0010
```

In hex, this is **F2.**

10) Let the input to the MixCols (during AES encryption) be $\begin{bmatrix} A0 & 74 & 65 & 96 \\ 2B & 8D & 2E & E3 \\ 99 & 1F & C8 & 87 \\ C5 & E5 & F7 & BB \end{bmatrix}$.

What’s the output in row 3 col 4? (The matrix by which to “multiply” is $\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$.)

The product we want to calculate is

01 x 96 + 01 x E3 + 02 x 87 + 03 x BB

01 x 96 = 96
01 x E3 = E3

```
02 x 87 = 10000111 << 1 = 100001110 = 00001110
                                          11011 (doing mod)
                                      -------------
                                      00010101 (15)

03 x BB = 03 x 10111011 =  10111011   (BB)
                          101110110   (02 x BB)
                          --------------
                            10111011
                            01110110
                            00011011
                            ------------
                            11010110 (D6)
```

Final Answer = (96 + E3) + (15 + D6) = 75 + C3 = **B6**