Tyler Fontana
CSCI 4130
Professor Phani
March 2 2020

Homework Assignment 2

Question 1 [10 points]

Addition in GF(2^5): Compute A(x) + B(x) mod P(x) in GF(2^5) using the irreducible polynomial P(x) = $x^5 + x^3 + 1$.

1. $A(x) = x^2 + 1$, $B(x) = x^3 + x^2 + 1$
2. $A(x) = x^2 + x$, $B(x) = x + 1$

If the operands are given in their standard form that is if the polynomial degree is less than *n*, then addition is GF (p^n) doesn't depend on the irreducible polynomial that we use. So, we must add each pair of coefficients and reduce it to mod P(x).

Addition of A(x) + B(x) (Part A):

$$A(x)_1 = x^2 + 1$$
$$B(x)_1 = x^3 + x^2 + 1$$

$$R(x) = A(x)_1 + B(x)_1$$
$$R(x) = (x^2 + 1) + (x^3 + x^2 + 1)$$
$$R(x) = x^3 + 2x^2 + 2$$

Seeing as this polynomial is a representation of a binary number, we cancel out any terms that have even coefficients. Thus, the $2x^2$ and 2 cancel out, leaving us with:

$$R(x) = x^3$$

The irreducible polynomial is $P(x) = x^5 + x^3 + 1$.
The maximum power of P(x) is 5.
The maximum power of R(x) is 3
Thus, the power of R(x) is less than (<) P(x) which implies that:
$R(x) = x^3$ is within GF(2^5)

Therefore,
$A(x)_1 + B(x)_1 \bmod P(x) = x^3$

Addition of A(x) + B(x) (Part B):

$$A(x)_2 = x^2 + x$$
$$B(x)_2 = x + 1$$

$$S(x) = A(x)_2 + B(x)_2$$
$$S(x) = (x^2 + x) + (x + 1)$$

$$S(x) = x^2 + 2x + 1$$

Seeing as this polynomial is a representation of a binary number, we cancel out any terms that have even coefficients. Thus, the 2x cancels out, leaving us with:

$$S(x) = x^2 + 1$$

The irreducible polynomial is $P(x) = x^5 + x^3 + 1$.
The maximum power of P(x) is 5.
The maximum power of S(x) is 2
Thus, the power of S(x) is less than (<) P(x) which implies that:
$S(x) = x^2 + 1$ is within $GF(2^5)$

Therefore,
$A(x)_2 + B(x)_2 \bmod P(x) = x^2 + 1$

Question 2 [10 points]
Compute in $GF(2^8)$:

$$(x^4 + x^3 + 1) / (x^3 + x^2 + x)$$

where the irreducible polynomial is the one used by AES, $P(x) = x^8 + x^4 + x^3 + x + 1$.
Note that Table 4.2 contains a list of all multiplicative inverses for this field.

$$P(x) = x^8 + x^4 + x^3 + x + 1 = (100011011)_2 = (11B)_{16}$$

$$R(x) = S(x) / T(x)$$
$$S(x) = x^4 + x^3 + 1$$
$$T(x) = x^3 + x^2 + x$$

$$(x^3 + x^2 + x) = (1110)_2$$
$$(1110)_2 = (14)_{10}$$
$$(14)_{10} = (0E)_{16}$$
$$T(x) = (0E)_{16}$$

$$T^{-1}(x) = (E5)_{16}$$
$$(E5)_{16} = (19)_{10} = (00010011)_2$$
$$(00010011)_2 = (x^4 + x + 1)_{polynomial}$$
$$T^{-1}(x) = x^4 + x + 1$$

$$R(x) = S(x)/T(x) * T^{-1}(x)/T^{-1}(x)$$
$$R(x) = (x^4 + x^3 + 1)/(x^3 + x^2 + x) * (x^4 + x + 1)/(x^4 + x + 1)$$
$$R(x) = (x^4 + x^3 + 1)(x^4 + x + 1) / T(x) * T^{-1}(x)$$
$$R(x) = x^8 + x^5 + x^4 + x^7 + x^4 + x^3 + x^4 + x + 1$$
$$R(x) = x^8 + x^7 + x^5 + 3x^4 + x^3 + x + 1$$

But: $(x^8)\bmod P(x) = (x^4 + x^3 + x + 1) \bmod P(x)$
Thus: $Z(x) = T^{-1}(x) + R(x)$
$Z(x) = (x^4 + x^3 + x + 1) + (x^7 + x^5 + 3x^4 + x^3 + x + 1)$

R(x) in the Z(x) equation above is the Term we received in the previous calculation minus the x^8 variable.

$$Z(x) = x^7 + x^5 + 4x^4 + 2x^3 + 2x + 2$$

Seeing as this is supposed to be a polynomial representation of a binary number, every two variables combined together cancel out. Thus, the $4x^4$, $2x^3$, 2x, & 2 terms cancel out, leaving us with the equation:

$$Z(x) = x^7 + x^5$$
The Highest power in Z(x) is 7
The Highest power in $GF(2^8)$ is 8,
Which implies that Z(x) is within $GF(2^5)$.

Therefore, Our Final Result is:
$$\boldsymbol{Z(x) = x^7 + x^5}$$

Question 3 [10 points]

We consider the first part of the ByteSub operation, i.e, the Galois field inver- sion.

1. Using Table 4.2, what is the inverse of the bytes 19, 3F and 3A, where each byte is given in hexadecimal notation?

$A(x) = (19)_{16}$
$B(x) = (3F)_{16}$
$C(x) = (3A)_{16}$

A(x) Conversion to $A^{-1}(x)$:
$(19)_{16} = (00011001)_2$
$\boldsymbol{A^{-1}(x) = (3F)_{16}}$
$(3F)_{16} = (00111111)_2$
$A^{-1}(x) = (00111111)_2$
$\boldsymbol{A^{-1}(x) = x^5 + x^4 + x^3 + x^2 + x + 1}$

B(x) Conversion to $B^{-1}(x)$:
$(3F)_{16} = (00111111)_2$
$\boldsymbol{B^{-1}(x) = (19)_{16}}$
$(19)_{16} = (00011001)_2$
$B^{-1}(x) = (00011001)_2$
$\boldsymbol{B^{-1}(x) = x^4 + x^3 + 1}$

C(x) Conversion to $C^{-1}(x)$:
$(3A)_{16} = (00111010)_2$
$\boldsymbol{C^{-1}(x) = (20)_{16}}$
$(20)_{16} = (00100000)_2$
$C^{-1}(x) = (00100000)_2$
$\boldsymbol{C^{-1}(x) = x^5}$

2. Verify your answer by performing a GF(2^8) multiplication with your answer and the input byte. Note that you have to represent each byte first as polynomials in GF(2^8). The MSB of each byte represents the x^7 coefficient.

Rijindael's Finite Field:

$$P(x) = x^8 + x^4 + x^3 + x + 1$$

$$A(x) = (19)_{16} = (00011001)_2$$
$$B(x) = (3F)_{16} = (00111111)_2$$
$$C(x) = (3A)_{16}\ (00111010)_2$$

$$A(x) = x^4 + x^3 + 1$$
$$A^{-1}(x) = x^5 + x^4 + x^3 + x^2 + x + 1$$

$$R_1(x) = (x^4 + x^3 + 1) * (x^5 + x^4 + x^3 + x^2 + x + 1)$$
$$R_1(x) = x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^5 + x^4 + x^3 + x^2 + x + 1$$
$$R_1(x) = x^9 + 2x^8 + 2x^7 + 2x^6 + 3x^5 + 3x^4 + 2x^3 + x^2 + x + 1$$
$$R_1(x) = x^9 + x^5 + x^4 + x^2 + x + 1$$
$$R_1(x) = (100110111)_2$$
$$Z_1(x) = R_1(x) \bmod x\ P(x)$$
$$Z_1(x) = (x^9 + x^5 + x^4 + x^2 + x + 1) \bmod x(x^8 + x^4 + x^3 + x + 1)$$
$$Z_1(x) = (x^9 + x^5 + x^4 + x^2 + x + 1) \bmod (x^9 + x^5 + x^4 + x^2 + x)$$
$$\mathbf{Z_1(x) = 1}$$

$$B(x) = x^5 + x^4 + x^3 + x^2 + x + 1$$
$$B^{-1}(x) = x^4 + x^3 + 1$$

$$R_2(x) = (x^5 + x^4 + x^3 + x^2 + x + 1) * (x^4 + x^3 + 1)$$
$$R_2(x) = x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^5 + x^4 + x^3 + x^2 + x + 1$$
$$R_2(x) = x^9 + 2x^8 + 2x^7 + 2x^6 + 3x^5 + 3x^4 + 2x^3 + x^2 + x + 1$$
$$R_2(x) = x^9 + x^5 + x^4 + x^2 + x + 1$$
$$R_1(x) = (100110111)_2$$
$$Z_1(x) = R_2(x) \bmod x\ P(x)$$
$$Z_2(x) = (x^9 + x^5 + x^4 + x^2 + x + 1) \bmod x(x^8 + x^4 + x^3 + x + 1)$$
$$Z_2(x) = (x^9 + x^5 + x^4 + x^2 + x + 1) \bmod (x^9 + x^5 + x^4 + x^2 + x)$$
$$\mathbf{Z_2(x) = 1}$$

$$C(x) = x^5 + x^4 + x^3 + x$$
$$C^{-1}(x) = x^5$$

$$R_3(x) = (x^5 + x^4 + x^3 + x) * (x^5)$$
$$R_3(x) = (x^{10} + x^9 + x^8 + x^6)$$
$$R_3(x) = (11101000000)_2$$
$$Z_3(x) = R_3(x) \bmod x\ P(x)$$
$$Z_3(x) = (x^{10} + x^9 + x^8 + x^6) \bmod x^2(x^8 + x^4 + x^3 + x + 1)$$
$$Z_3(x) = (x^{10} + x^9 + x^8 + x^6) \bmod (x^{10} + x^6 + x^5 + x^3 + x^2)$$
$$\mathbf{Z_3(x) = 1}$$

Question 4 [10 points]

Your task is to compute the S-Box, i.e., the ByteSub, values for the input bytes 19, 3F and 3A, where each byte is given in hexadecimal notation.

1. First, look up the inverses using Table 4.2 to obtain values B′. Now, perform the affine mapping by computing the matrix–vector multiplication and addition.

$A(x) = (19)_{16}$
$B(x) = (3F)_{16}$
$C(x) = (3A)_{16}$

Inverse Substitution Box:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	01	8D	F6	CB	52	7B	D1	E8	4F	29	C0	B0	E1	E5	C7
1	74	B4	AA	4B	99	2B	60	5F	58	3F	FD	CC	FF	40	EE	B2
2	3A	6E	5A	F1	55	4D	A8	C9	C1	0A	98	15	30	44	A2	C2
3	2C	45	92	6C	F3	39	66	42	F2	35	20	6F	77	BB	59	19
4	1D	FE	37	67	2D	31	F5	69	A7	64	AB	13	54	25	E9	09
5	ED	5C	05	CA	4C	24	87	BF	18	3E	22	F0	51	EC	61	17
6	16	5E	AF	D3	49	A6	36	43	F4	47	91	DF	33	93	21	3B
7	79	B7	97	85	10	B5	BA	3C	B6	70	D0	06	A1	FA	81	82
8	83	7E	7F	80	96	73	BE	56	9B	9E	95	D9	F7	02	B9	A4
9	DE	6A	32	6D	D8	8A	84	72	2A	14	9F	88	F9	DC	89	9A
A	FB	7C	2E	C3	8F	B8	65	48	26	C8	12	4A	CE	E7	D2	62
B	0C	E0	1F	EF	11	75	78	71	A5	8E	76	3D	BD	BC	86	57
C	0B	28	2F	A3	DA	D4	E4	0F	A9	27	53	04	1B	FC	AC	E6
D	7A	07	AE	63	C5	DB	E2	EA	94	8B	C4	D5	9D	F8	90	6B
E	B1	0D	D6	EB	C6	0E	CF	AD	08	4E	D7	E3	5D	50	1E	B3
F	5B	23	38	34	68	46	03	8C	DD	9C	7D	A0	CD	1A	41	1C

A(x) Conversion to $A^{-1}(x)$:
$(19)_{16} = (00011001)_2$
$\boldsymbol{A^{-1}(x) = (3F)_{16}}$
$(3F)_{16} = (00111111)_2$
$A^{-1}(x) = (00111111)_2$
$\boldsymbol{A^{-1}(x) = x^5 + x^4 + x^3 + x^2 + x + 1}$

B(x) Conversion to $B^{-1}(x)$:
$(3F)_{16} = (00111111)_2$
$\boldsymbol{B^{-1}(x) = (19)_{16}}$
$(19)_{16} = (00011001)_2$
$B^{-1}(x) = (00011001)_2$
$\boldsymbol{B^{-1}(x) = x^4 + x^3 + 1}$

C(x) Conversion to $C^{-1}(x)$:
$(3A)_{16} = (00111010)_2$
$\boldsymbol{C^{-1}(x) = (20)_{16}}$
$(20)_{16} = (00100000)_2$
$C^{-1}(x) = (00100000)_2$
$\boldsymbol{C^{-1}(x) = x^5}$

(M * $A^{-1}(x)$) + V Graph:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} b_0\prime \\ b_1\prime \\ b_2\prime \\ b_3\prime \\ b_4\prime \\ b_5\prime \\ b_6\prime \\ b_7\prime \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

Substitution Box:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Now, we have to perform the affine mapping which involves a bitwise matrix multiplication (AND) and a bitwise XOR which is represented by the picture above. Let it be known that b_0! represents the least significant bit of our inversed byte. In other words, in $(00111111)_b$, the 1 furthest to the right will represent b_0!.

$A^{-1}(x) = (00111111)_2$

$Sub(A(x)) = M * A^{-1}(x) + V$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ 3 \\ 4 \\ 5 \\ 6 \\ 5 \\ 3 \end{pmatrix}$$

Using Mod 2 on the resulting vector above, we are left with the binary number:
11010100 = D4

Upon looking up the value for the original input of A(x) = 19 in the Substitution Graph, we indeed discover that the value ***19*** maps to the value ***D4***.

$B^{-1}(x) = (00011001)_2$
$Sub(B(x)) = M * B^{-1}(x) + V$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 1 \\ 2 \\ 3 \\ 3 \\ 3 \\ 2 \end{pmatrix}$$

Using Mod 2 on the resulting vector above, we are left with the binary number:
01110101 = 75

Upon looking up the value for the original input of B(x) = 3F in the Substitution Graph, we indeed discover that the value ***3F*** maps to the value ***75***.

$C^{-1}(x) = (00100000)_2$
$Sub(C(x)) = M * C^{-1}(x) + V$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 0 \\ 0 \\ 0 \\ 2 \\ 2 \\ 1 \end{pmatrix}$$

Using mod 2 on the resulting vector above, we are left with the binary number:
10000000 = 80

Upon looking up the value for the original input of C(x) = 3A in the Substitution Graph, we indeed discover that the value ***3A*** maps to the value ***80***.

2. Verify your result using the S-Box Table 4.3.

Result Verified using S-Box Table in Previous Step.

3. What is the value of S(11111111)?

$$(1111111)_2 = (FF)_{16}$$

After we have converted $(11111111)_2$ into the hexadecimal $(FF)_{16}$, we need to find $(FF)_{16}$ inside of the Substitution Graph and see which values it corresponds to. Eventually, we find that $(FF)_{16}$ corresponds to the value 7D. We then take the inverse of 7D and plug it into our earlier equation to find our results:

$$D(x) = 7D, \qquad D^{-1}(x) = FA$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 5 \\ 5 \\ 3 \\ 3 \\ 3 \\ 5 \\ 5 \\ 5 \end{pmatrix}$$

Using Mod 2 on the resulting vector above, we are left with the binary number:
11111111 = FF

Which proves our solution. Therefore, S(11111111) is equivalent to the Substitution Value of **S(FF)** and the original value of **7D**.

Question 5 [60 points]

¡Vamos!

After her last encryption attempt ended up as a fiasco, Alice has now learnt her lesson. She vowed not to use such weak ciphers again.

Now, Alice has another need for encryption. She needs to send some confidential material to Bob. After we covered Chapter 4, she was impressed by AES and decided to use it for this task. She knew that Oscar or for that matter, anyone isn't rich enough to be able to brute force AES-128. So, she decided to use AES-128 to keep the secret away from Bob. She did the following things:

1. Alice and Bob have already agreed upon a 128 bit secret key. (Note that this can be represented by 32 Hexadecimal characters)
2. She converted her sensitive plain text to a hexadecimal string using this encoding tool: https://codebeautify.org/string-hex-converter
3. She input the 32 digit hexadecimal key and her hexadecimal encoded text (which is the sensitive plain text) into this AES tool to get the encrypted text and sent it to Bob. http://extranet.cryptomathic.com/aescalc (She used the default ECB mode for this)

Oscar who was keenly spying on Alice's network got to find out that this was the ciphertext that Alice sent to Bob:

E0ECE8BFB0C9854BC9916246DC1E7EC42994C78EBC0796690E7E0385FA49EA367CD829
E046538A205A27B6848E26C274FD1494A930F64E0E7BE70DDCEC6DB9CAED505D4E8F77
5E4AB8920E02B1010869A96EBBB65B6BA6D78A733735A0D890D6AF11586CB504FDCAD9
8CB1D1BAF7DA4A0F205304D1F7596AE23E9414FD2B56458CC1961C131C52524BF7B2A1
5140E943D61AA53F280340693612F8A9551D2406CE6CF66FCAB6F925BD5EB76CFB25945
740D229F0D125E6DADDFA1FACA411E93AE56DFD27F186F30DB22BC79C17594F16FE414
57D2C769EF08201B0FF91D482BF92EAA0AEE4991009C8717EFB6DC0CD0B535E38EB13E
E4AC65FCE00E82C6587FECBC9EC550DDB66587D5735B1DB78BFB8AF54F1F237D2A2EE
AB2B61D195105CBB6557644B2474ED96DBB918DE09D0B17DED901BE61C97A1CD3B200
A3678369FF4

But, unfortunately for Alice, there was a small security bug in the software that Alice used. The authors of the software forgot to wipe out the variables that are used for storing the sub-keys during encryption. As a result, Oscar was able to log into the same computer that Alice used and obtain the last couple of AES subkeys that were used from unused system memory. These are the keys that Oscar got:

k_{10} = F6FA49F03DBF50565D152248ABC2E463
k_{11} = E593B292D82CE2C48539C08C2EFB24EF

Given this info, will Oscar be able to find the key to decrypt Alice's cipher text? Pretend you are Oscar and try to decrypt the text. What is the text that Alice sent to Bob?

Hint: The goal should be to first obtain the key that was used. For this, you can either try to code this or do this manually. Either way, you should try to retrace the steps of the key schedule. But, it might be easier and less error-prone if you code this. Note that you don't really need two sub-keys for this problem. The tenth one is given just as a check to make sure that you are on the right path.

Hint 2: Snippets of this code might serve you as hints for you - https://gist.github.com/raullenchai/2920069 (Note that this is for AES-256 though)

The Round Keys Generated In the Key Expansion Process are Provided on the Next Page:

```
Round Key [10]: 0xE5 0x93 0xB2 0x92 0xD8 0x2C 0xE2 0xC4 0x85 0x39 0xC0 0x8C 0x2E 0xFB 0x24 0xEF
Round Key [9]:  0xF6 0xFA 0x49 0xF0 0x3D 0xBF 0x50 0x56 0x5D 0x15 0x22 0x48 0xAB 0xC2 0xE4 0x63
Round Key [8]:  0xE3 0x4E 0xB8 0xB2 0xCB 0x45 0x19 0xA6 0x60 0xAA 0x72 0x1E 0xF6 0xD7 0xC6 0x2B
Round Key [7]:  0x9C 0xC3 0x2E 0x22 0x28 0x0B 0xA1 0x14 0xAB 0xEF 0x6B 0xB8 0x96 0x7D 0xB4 0x35
Round Key [6]:  0x93 0x5D 0x73 0x05 0xB4 0xC8 0x8F 0x36 0x83 0xE4 0xCA 0xAC 0x3D 0x92 0xDF 0x8D
Round Key [5]:  0x8B 0x04 0x8E 0xAB 0x27 0x95 0xFC 0x33 0x37 0x2C 0x45 0x9A 0xBE 0x76 0x15 0x21
Round Key [4]:  0x25 0x57 0x64 0x0C 0xAC 0x91 0x72 0x98 0x10 0xB9 0xB9 0xA9 0x89 0x5A 0x50 0xBB
Round Key [3]:  0x3C 0x49 0xAD 0xE2 0x89 0xC6 0x16 0x94 0xBC 0x28 0xCB 0x31 0x99 0xE3 0xE9 0x12
Round Key [2]:  0x27 0xDA 0x8B 0xDD 0xB5 0x8F 0xBB 0x76 0x35 0xEE 0xDD 0xA5 0x25 0xCB 0x22 0x23
Round Key [1]:  0x1A 0xCC 0xCF 0x17 0x92 0x55 0x30 0xAB 0x80 0x61 0x66 0xD3 0x10 0x25 0xFF 0x86

Original Key:   0x00 0x22 0x33 0x77 0x88 0x99 0xFF 0xBC 0x12 0x34 0x56 0x78 0x90 0x44 0x99 0x55
```

The Decrypted Text of the HEXIDECIMAL Message in Question 5 is provided below:

"Geaux" is a term usually used in sports to root for a Louisiana sports team. It is pronounced "go" because many French words or names end with -eaux and are pronounced with the long O sound. So, if you see "Geaux Tigers" or "Geaux Saints", it just is rooting for them by saying "Go Tigers" or "Go Saints"

In order to Generate all of the previously used Round Keys in addition to find the Decrypted Text, I created a Java Program Called **InverseKeyGeneration**. This Class contains methods for the *Key Expansion Reversal Process*, the *Key Generation Mixing Columns Step*, the *Key Generation Substitution Step*, and the *Key Generation Round Constant Addition Step*. Additionally, this Class also provides an AES-128 bit Decryption Method, which takes the Original Key in its Byte Array Form, along with the Encrypted Message in its Byte Array Form as its Parameters. From there, the method will attempt to Decrypt the Message Using the *AES-128 bit ECB Block Cipher Mode (Due to ECB Mode Being the Only Block Cipher Mode with Doesn't Require an Initialization Vector (IV))*. The Application Also makes use of an Object Java Class called **RoundKey** which is used in order to save the Details of Each Round Key found during the Reverse Key Generation Process. (***Note**: The Encrypted Message in the Question Above is Hard Coded into the **InverseKeyGeneration** Java Class along with the HEXIDECIMAL Byte Entries of the Subsitution Box, Inverse Subsitution Box, and Round Constant Box*).

Both the **InverseKeyGeneration.java** and **RoundKey.java** files will be included in the Submission Archive File.