ASSIGNMENT REPORT

APPLIED CRYPTOGRAPHY (CS-6530), IIT MADRAS
Designing and Implementing a Block Cipher Similar to AES

Faculty

Asst Prof. Chester Rebeiro

Members

Prateek Bhamri (CS17M003)
Vivek Kumar Agrawal (CS17M049)

1. Pick a name for your cipher.

AES Clone

2. Add the last two digits of your's and your partner's roll number modulo 30. If this happens to be, say i, then pick the i-th irreducible polynomial from the list at the end of this document. You would be designing a cipher with this irreducible polynomial.

Prateek Bhamri (CS17M003)
Vivek Kumar Agrawal (CS17M049)
03 + 49 = 52
52 mod 30 = 22
463 is corresponding coefficient of irreducible polynomial as per the table given.
The irreducible polynomial is:
$463_{10} = 111001111_2$
Polynomial $x^8 + x^7 + x^6 + x^4 + x^3 + x^2 + 1$

3. Write functions in C or x86 assembly as efficiently as possible for performing finite field operations in your chosen finite field.

The addition, subtraction, multiplication, division and inverse operations in finite fields is illustrated along with comments in the code in the **AES_CLONE.cpp** file..

4. Design the SBox using the same technique as that of the AES SBox, i.e. using field inversion. You would need to choose the affine transformation considering the desirable SBox properties (next question).

The Affine transformation that has been used is similar to AES as the optimal results of the S-Box properties were obtained using it. However, the code has been modelled in a way to cater for other Affine Transformations as well, the **affine_mat[ ][ ]** in the program stores the affine transformation matrix and **c[ ] = "10110001"** is the constant matrix both of which are similar to the one used in AES.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Fig 1: AFFINE TRANSFORMATION

--------------------------INVERSE TABLE---------------------------------

0	1	231	186	148	211	93	66	74	62	142	254	201	138	33	244
37	233	31	123	71	221	127	195	131	250	69	175	247	239	122	18
245	14	147	119	232	16	218	224	196	53	137	86	216	101	134	170
166	160	125	226	197	41	176	215	156	65	144	235	61	60	9	75
157	57	7	92	174	26	220	20	116	236	8	63	109	89	112	103
98	190	253	96	163	94	43	136	108	77	213	159	67	6	85	162
83	252	80	191	217	45	113	79	133	229	243	173	88	76	140	181
78	102	199	185	72	237	146	35	249	151	30	19	227	50	194	22
169	183	251	24	228	104	46	171	87	42	13	200	110	180	10	255
58	234	118	34	4	210	248	121	209	164	203	240	56	64	212	91
49	167	95	84	153	208	48	161	182	128	47	135	242	107	68	27
54	214	193	207	141	111	168	129	198	115	3	230	205	222	81	99
206	178	126	23	40	52	184	114	139	12	241	154	223	188	192	179
165	152	149	5	158	90	177	55	44	100	38	225	70	21	189	204
39	219	51	124	132	105	187	2	36	17	145	59	73	117	246	29
155	202	172	106	15	32	238	28	150	120	25	130	97	82	11	143

Fig 2: INVERSE VALUES TABLE

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

Fig 3: S-BOX (Hexadecimal representation)

5. Write programs that would evaluate the following properties of your SBox and compare it with that of the AES SBox.

– Balancedness property

```
--------------------------------BALANCEDNESS--------------------------------
It does satisfies balanceness property
```

Fig 4

The balanceness property condition is met if the 0's and 1's appear with equal probability. As the S-Box has non-repeating and 265 unique values(represented in the Hexadecimal representation of the S-Box). The uniqueness of each element in the S-Box was checked to obtain the validity. The S-Box designed was found similar to AES wrt meeting the Balancedness property.

– Fixed Points

```
-------------------------------FIXED POINT--------------------------------
It does satisfies fixed point property
```

Fig 5

Fixed point property states that for no input to the S-Box the output value should be similar to the input value as in such a scenario the attach is easy and the encryption itself fails. The same was checked in the program and it was found that the S-Box designed is having no Fixed Points similar to AES S-Box.

– SAC (Strict Avalanche Effect)

```
-----------------------------SAC-----------------------------
0.42 0.58 0.58 0.53 0.48 0.50 0.53 0.44 0.47 0.50 0.53 0.47 0.50 0.50 0.52 0.64
0.44 0.39 0.42 0.42 0.55 0.52 0.52 0.47 0.41 0.36 0.55 0.47 0.47 0.39 0.58 0.45
0.45 0.48 0.47 0.52 0.41 0.48 0.47 0.44 0.53 0.55 0.48 0.45 0.45 0.47 0.48 0.47
0.45 0.44 0.44 0.64 0.48 0.45 0.59 0.53 0.45 0.47 0.44 0.50 0.48 0.47 0.56 0.55
0.53 0.47 0.53 0.55 0.53 0.59 0.47 0.62 0.44 0.59 0.50 0.45 0.45 0.52 0.48 0.55
0.64 0.50 0.61 0.53 0.58 0.64 0.59 0.56 0.52 0.55 0.50 0.48 0.50 0.44 0.47 0.44
0.52 0.52 0.45 0.47 0.38 0.48 0.53 0.48 0.48 0.55 0.41 0.45 0.45 0.55 0.55 0.50
0.45 0.41 0.58 0.53 0.48 0.48 0.53 0.48 0.48 0.64 0.55 0.50 0.50 0.55 0.58 0.55
0.47 0.58 0.38 0.48 0.50 0.47 0.62 0.52 0.55 0.48 0.53 0.45 0.53 0.52 0.53 0.48
0.50 0.56 0.36 0.52 0.58 0.55 0.52 0.59 0.47 0.52 0.41 0.58 0.56 0.42 0.50 0.41
0.50 0.53 0.42 0.52 0.48 0.36 0.45 0.52 0.50 0.58 0.59 0.50 0.64 0.53 0.62 0.53
0.53 0.55 0.50 0.47 0.45 0.38 0.38 0.47 0.58 0.62 0.50 0.48 0.58 0.48 0.52 0.50
0.55 0.48 0.47 0.55 0.61 0.58 0.48 0.52 0.52 0.53 0.53 0.41 0.39 0.55 0.47 0.47
0.50 0.50 0.58 0.59 0.47 0.58 0.48 0.53 0.52 0.53 0.55 0.53 0.52 0.48 0.52 0.50
0.48 0.56 0.47 0.56 0.53 0.55 0.55 0.59 0.53 0.44 0.53 0.48 0.55 0.61 0.44 0.41
0.50 0.50 0.72 0.45 0.44 0.47 0.47 0.58 0.44 0.47 0.67 0.48 0.53 0.47 0.61 0.53

Average of SAC table is: 0.51
```

Fig 6: SAC TABLE

The SAC indicates the probability of the change of the output if one bit is changed in the input. For the AES the SAC value is 0.50 the same was computed for the S-Box designed and it was obtained to be **0.51**.

– Non-linearity

```
------------------------------Non Linearity------------------------------
112
```

Fig 7

Non-linearity is the minimum Hamming distance from all the possible linear functions. The Linear Approximation designed in the next question was used to obtain this value. The nonlinearity was obtained to be **112**; further as per studies non-linearity should be <120 and the same was obtained.

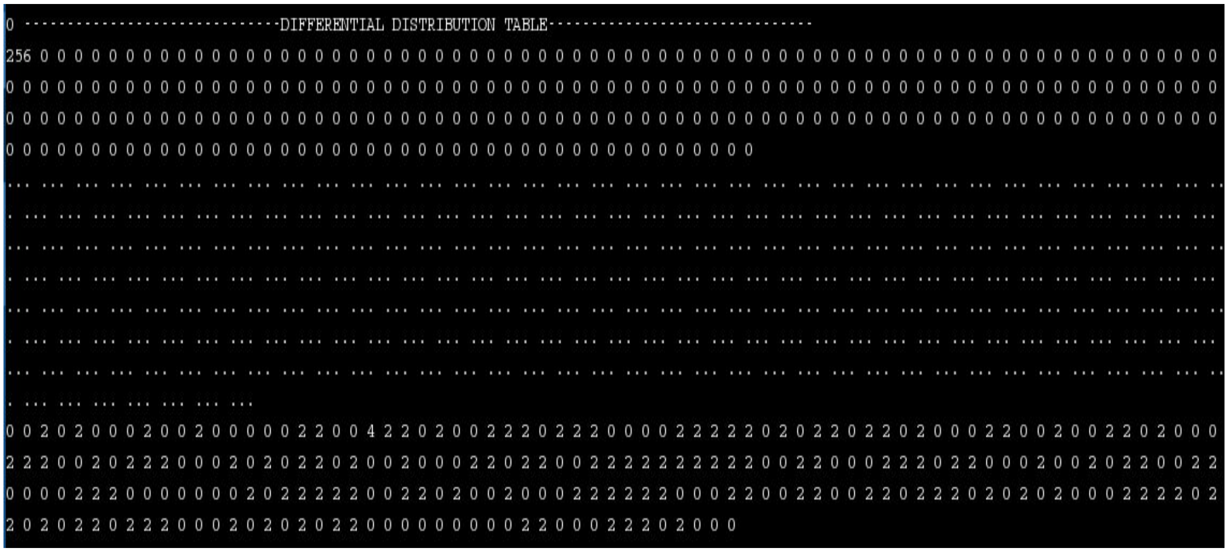
6. Draw the linear approximation table of your SBox.

```
0 ----------------------------LINEAR APPROXIMATION TABLE-----------------------------
256 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 1
28 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 12
8 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128
 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128
128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 1
28 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 12
8 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128
 128 128 128 128 128 128 128 128 128
... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... .
.. ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ..
. ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
 ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... .
.. ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ..
. ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
 ... ... ... ... ... ... ... ... ...
128 132 114 134 126 130 116 128 126 134 124 116 116 140 126 126 118 134 132 124 116 124 118 142 140 120 134 122 130 134 136 124 116 140 134 1
26 138 122 120 128 138 126 136 124 136 116 122 126 122 134 136 124 144 124 138 134 128 128 122 130 142 126 124 124 126 134 116 132 128 120 12
2 130 120 116 122 126 138 118 120 116 128 116 130 118 122 118 112 132 138 122 120 136 132 140 126 126 122 126 112 124 124 136 126 122 132 132
 118 126 134 126 124 132 140 116 126 118 134 126 116 116 126 138 140 136 120 116 138 142 116 124 126 134 122 138 136 128 126 122 140 120 116
136 126 138 130 126 144 132 128 124 114 126 132 124 118 118 130 122 128 120 132 128 134 130 122 118 120 124 126 118 116 124 132 124 126 142 1
30 122 136 136 128 128 130 130 140 136 134 122 122 142 136 140 118 138 140 136 136 132 130 126 124 116 118 134 134 134 124 140 120 128 130 13
8 122 130 136 120 126 130 140 128 120 140 114 126 142 142 124 116 120 120 114 130 140 136 126 130 142 122 116 128 112 124 130 142 122 142 120
 116 134 134 124 140 120 128 114 114
```

Fig 8: LINEAR APPROXIMATION TABLE

Only the first and last row of the table have been printed as it is a 256 X 256 table.

7. Draw the differential distribution table for your SBox.



```
0 ----------------------------DIFFERENTIAL DISTRIBUTION TABLE----------------------------
256 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ..
. ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ..
. ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ..
. ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ..
. ... ... ... ... ... ... ...
0 0 2 0 2 0 0 0 2 0 0 2 0 0 0 0 0 2 2 0 0 4 2 2 0 2 0 0 2 2 2 0 2 2 2 0 0 0 0 2 2 2 2 2 0 2 0 2 2 0 2 2 0 2 0 0 0 2 2 0 0 2 0 0 2 2 0 2 0 0 0
2 2 2 0 0 2 0 2 2 2 0 0 0 2 0 2 0 2 2 0 2 0 0 2 0 0 0 2 2 0 2 2 0 0 2 2 2 2 2 2 2 2 2 2 0 0 2 2 0 0 0 2 2 2 0 2 2 0 0 0 2 0 0 2 0 2 2 0 0 2 2
0 0 0 0 2 2 2 0 0 0 0 0 0 0 2 0 2 2 2 2 2 0 0 2 2 0 2 0 0 2 0 0 0 2 2 2 2 2 2 0 0 0 2 2 0 0 2 2 0 0 2 2 0 2 2 2 0 2 0 2 0 2 0 0 0 2 2 2 2 0 2
2 0 2 0 2 2 0 2 2 2 0 0 0 2 0 2 0 2 0 2 2 0 0 0 0 0 0 0 0 0 2 2 0 0 0 2 2 2 0 2 0 0 0
```

Fig 9: DIFFERENTIAL DISTRIBUTION TABLE

Only the first and last row of the table have been printed as it is a 256 X 256 table.

NOTE: All the figures in this report are screenshots of outputs of the executed code (AES_Clone.cpp).