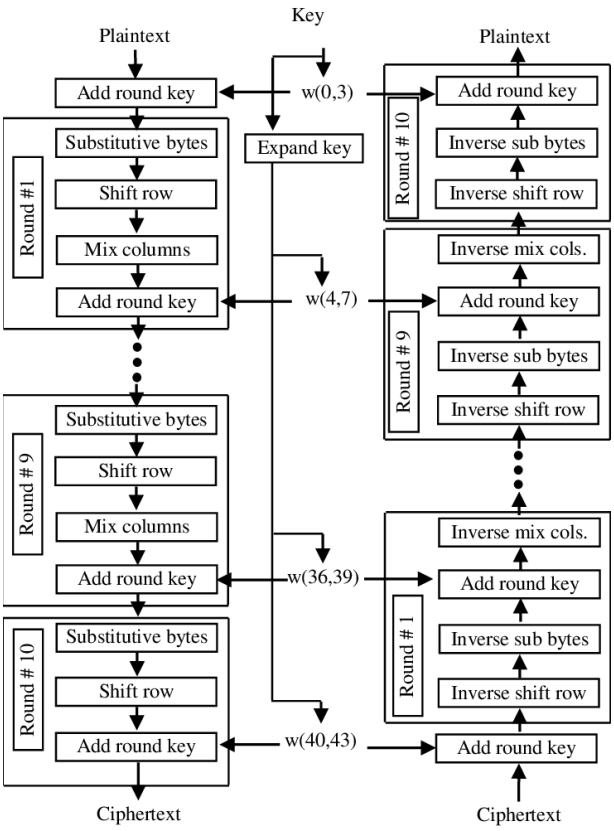
Timothy Hoang
Prof. Rivas
MSCS630
4/12/20

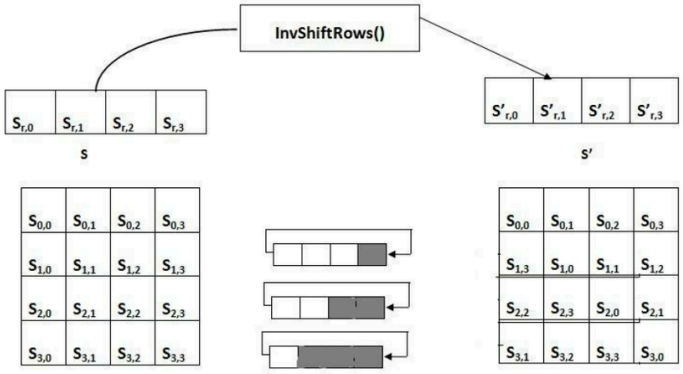
Milestone

I have finished the lexicon reading and password generation portion of my project. The idea was to have a lexicon for the program to read that the user creates. From the lexicon of words, a random "readable" password will be generated. Each word in the txt file is separated by line and although you can add any length word, the program will only choose words that are 16 characters or less since that is what is required for the AES 128. If the smallest word is too big to fill in the gap, randomly generated characters will be chosen for the remainder of the password. In addition to this, the user is able to toggle between the "readable" password and a completely random one where each character is randomly generated and not read at all. These programs are only intended for English letters and numbers and may produce problems when introduced to other characters.

For the decryption portion, the methodology is outlined, I will have create inverse programs for ShiftRow(), NibbleSub(), and MixColumn(). The only major changes to these programs is that in InvMixColumn() where I will have to doing Galois multiplication in different fields. In addition to these inverse functions, I will have to reverse the order of key operations. After all of these operations are implemented and run in the correct order, all that will be left would be to create a java applet so that I can run it on a server to be distributed online.



To the left is the order of operations to be done for encryption and decryption. Since the encryption part is complete from lab5, I just need to worry about the right side of the picture from the bottom up. I will start by adding the round key. Then for the next 9 keys, I will invShiftRow(), invNibbleSub(), add the round key, then InvMixColumn(). Lastly, for the last key I will do the same process except leave out the InvMixColumn().



In InvShiftRow(), all I am doing in changing the rows opposite to how it was shifted in ShiftRow(). That is, instead of taking the last 1, 2, and 3 subjects of the bottom three rows in the matrix and bringing them to the front, I take the first 3, 2, and 1 subjects from those rows and take them to the back

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

For InvNibbleSub(), I am doing the exact same operation but using the substitution values shown on the left to retrieve the inverse of the S-Box.

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \times \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}$$

For InvMixColumn(), I will be performing operations as shown to the left, where the numbers in the second matrix are the Galois field numbers.