CRYPTOGRAPHY #05

AES-128 Supplement

Jacek Tchórzewski, jacek.tchorzewski@pk.edu.pl

Hint: Write functions in the same order as I present them in this document.

1. SubWord Examples

   - SubWord(0x01020304) -> 0x7c777bf2
   - SubWord(0x04030201) -> 0xf27b777c

2. Expand Key example presented in NIST FIPS 197.

Cipher Key = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

for $Nk = 4$, which results in

$w_0 =$ **2b7e1516** $w_1 =$ **28aed2a6** $w_2 =$ **abf71588** $w_3 =$ **09cf4f3c**

i (dec)	temp	After RotWord()	After SubWord()	Rcon[i/Nk]	After XOR with Rcon	w[i-Nk]	w[i]= temp XOR w[i-Nk]
4	09cf4f3c	cf4f3c09	8a84eb01	01000000	8b84eb01	2b7e1516	a0fafe17
5	a0fafe17					28aed2a6	88542cb1
6	88542cb1					abf71588	23a33939
7	23a33939					09cf4f3c	2a6c7605
8	2a6c7605	6c76052a	50386be5	02000000	52386be5	a0fafe17	f2c295f2
9	f2c295f2					88542cb1	7a96b943
10	7a96b943					23a33939	5935807a
11	5935807a					2a6c7605	7359f67f
12	7359f67f	59f67f73	cb42d28f	04000000	cf42d28f	f2c295f2	3d80477d
13	3d80477d					7a96b943	4716fe3e
14	4716fe3e					5935807a	1e237e44
15	1e237e44					7359f67f	6d7a883b
16	6d7a883b	7a883b6d	dac4e23c	08000000	d2c4e23c	3d80477d	ef44a541
17	ef44a541					4716fe3e	a8525b7f
18	a8525b7f					1e237e44	b671253b
19	b671253b					6d7a883b	db0bad00
20	db0bad00	0bad00db	2b9563b9	10000000	3b9563b9	ef44a541	d4d1c6f8
21	d4d1c6f8					a8525b7f	7c839d87
22	7c839d87					b671253b	caf2b8bc
23	caf2b8bc					db0bad00	11f915bc

24	11f915bc	f915bc11	99596582	20000000	b9596582	d4d1c6f8	6d88a37a
25	6d88a37a					7c839d87	110b3efd
26	110b3efd					caf2b8bc	dbf98641
27	dbf98641					11f915bc	ca0093fd
28	ca0093fd	0093fdca	63dc5474	40000000	23dc5474	6d88a37a	4e54f70e
29	4e54f70e					110b3efd	5f5fc9f3
30	5f5fc9f3					dbf98641	84a64fb2
31	84a64fb2					ca0093fd	4ea6dc4f
32	4ea6dc4f	a6dc4f4e	2486842f	80000000	a486842f	4e54f70e	ead27321
33	ead27321					5f5fc9f3	b58dbad2
34	b58dbad2					84a64fb2	312bf560
35	312bf560					4ea6dc4f	7f8d292f
36	7f8d292f	8d292f7f	5da515d2	1b000000	46a515d2	ead27321	ac7766f3
37	ac7766f3					b58dbad2	19fadc21
38	19fadc21					312bf560	28d12941
39	28d12941					7f8d292f	575c006e
40	575c006e	5c006e57	4a639f5b	36000000	7c639f5b	ac7766f3	d014f9a8
41	d014f9a8					19fadc21	c9ee2589
42	c9ee2589					28d12941	e13f0cc8
43	e13f0cc8					575c006e	b6630ca6

NOTE: RotWord is a left binary rotation by 8 bits (1 byte).

3. AddRoundKey

   Is perfroming 4 XORS on state integers and round key integers. Example:
   key = [0x00010203, 0x04050607, 0x08090a0b, 0x0c0d0e0f]
   state = [0x00112233, 0x44556677, 0x8899aabb, 0xccddeeff]
   state' = [0x00102030, 0x40506070, 0x8090a0b0, 0xc0d0e0f0]

4. ShiftRows example:

State:		After shifting:
0x63cab704		0x6353e08c
0x0953d051	→	0x0960e104
0xcd60e0e7		0xcd70b751
0xba70e18c		0xbacad0e7

   Note: You need to perform State transposition, then apply ShiftRows function and finally make a transposition of result.

5. MixColumns

   Is performing matrix multiplication. Note, that calculations are done under Galois Filed (see PDF, chapter 2.1.7). You can use algorithm fastMul in AESarrays.java. It is appropriately multiplying two numbers uder GF(2^8).

   Example:

State:		After mixing columns:
0x6353e08c		0x5f726415
0x0960e104	→	0x57f5bc92
0xcd70b751		0xf7be3b29
0xbacad0e7		0x1db9f91a

6. Whole process presented in [NIST FIPS 197]:

AES-128 (*Nk*=4, *Nr*=10)

```
PLAINTEXT:          00112233445566778899aabbccddeeff
KEY:                000102030405060708090a0b0c0d0e0f

CIPHER (ENCRYPT):

round[ 0].input     00112233445566778899aabbccddeeff
round[ 0].k_sch     000102030405060708090a0b0c0d0e0f
round[ 1].start     00102030405060708090a0b0c0d0e0f0
round[ 1].s_box     63cab7040953d051cd60e0e7ba70e18c
round[ 1].s_row     6353e08c0960e104cd70b751bacad0e7
round[ 1].m_col     5f72641557f5bc92f7be3b291db9f91a
round[ 1].k_sch     d6aa74fdd2af72fadaa678f1d6ab76fe
round[ 2].start     89d810e8855ace682d1843d8cb128fe4
round[ 2].s_box     a761ca9b97be8b45d8ad1a611fc97369
round[ 2].s_row     a7be1a6997ad739bd8c9ca451f618b61
round[ 2].m_col     ff87968431d86a51645151fa773ad009
round[ 2].k_sch     b692cf0b643dbdf1be9bc5006830b3fe
round[ 3].start     4915598f55e5d7a0daca94fa1f0a63f7
round[ 3].s_box     3b59cb73fcd90ee05774222dc067fb68
round[ 3].s_row     3bd92268fc74fb735767cbe0c0590e2d
round[ 3].m_col     4c9c1e66f771f0762c3f868e534df256
round[ 3].k_sch     b6ff744ed2c2c9bf6c590cbf0469bf41
round[ 4].start     fa636a2825b339c940668a3157244d17
round[ 4].s_box     2dfb02343f6d12dd09337ec75b36e3f0
round[ 4].s_row     2d6d7ef03f33e334093602dd5bfb12c7
round[ 4].m_col     6385b79ffc538df997be478e7547d691
round[ 4].k_sch     47f7f7bc95353e03f96c32bcfd058dfd
```

```
round[ 5].start    247240236966b3fa6ed2753288425b6c
round[ 5].s_box    36400926f9336d2d9fb59d23c42c3950
round[ 5].s_row    36339d50f9b539269f2c092dc4406d23
round[ 5].m_col    f4bcd45432e554d075f1d6c51dd03b3c
round[ 5].k_sch    3caaa3e8a99f9deb50f3af57adf622aa
round[ 6].start    c81677bc9b7ac93b25027992b0261996
round[ 6].s_box    e847f56514dadde23f77b64fe7f7d490
round[ 6].s_row    e8dab6901477d4653ff7f5e2e747dd4f
round[ 6].m_col    9816ee7400f87f556b2c049c8e5ad036
round[ 6].k_sch    5e390f7df7a69296a7553dc10aa31f6b
round[ 7].start    c62fe109f75eedc3cc79395d84f9cf5d
round[ 7].s_box    b415f8016858552e4bb6124c5f998a4c
round[ 7].s_row    b458124c68b68a014b99f82e5f15554c
round[ 7].m_col    c57e1c159a9bd286f05f4be098c63439
round[ 7].k_sch    14f9701ae35fe28c440adf4d4ea9c026
round[ 8].start    d1876c0f79c4300ab45594add66ff41f
round[ 8].s_box    3e175076b61c04678dfc2295f6a8bfc0
round[ 8].s_row    3e1c22c0b6fcbf768da85067f6170495
round[ 8].m_col    baa03de7a1f9b56ed5512cba5f414d23
round[ 8].k_sch    47438735a41c65b9e016baf4aebf7ad2
round[ 9].start    fde3bad205e5d0d73547964ef1fe37f1
round[ 9].s_box    5411f4b56bd9700e96a0902fa1bb9aa1
round[ 9].s_row    54d990a16ba09ab596bbf40ea111702f
round[ 9].m_col    e9f74eec023020f61bf2ccf2353c21c7
round[ 9].k_sch    549932d1f08557681093ed9cbe2c974e
round[10].start    bd6e7c3df2b5779e0b61216e8b10b689
round[10].s_box    7a9f102789d5f50b2beffd9f3dca4ea7
round[10].s_row    7ad5fda789ef4e272bca100b3d9ff59f
round[10].k_sch    13111d7fe3944a17f307a78b4d2b30c5
round[10].output   69c4e0d86a7b0430d8cdb78070b4c55a
```

Exercise:

Write a program which will be ciphering and deciphering message (plaintext) given in example above. Hints:
- write a function which will be performing matrix transposition, eg.:

0x6353e08c		0x6309cdba
0x0960e104	→	0x536070ca
0xcd70b751		0xe0e1b7d0
0xbacad0e7		0x8c0451e7

Of course the problem is that rows are stored as integers.