CS454/654 Reliability and Security of Computing Systems

HW 1
Total points: 40

Objective: Implement Data Encryption Standard

- You can choose any of the following programming languages: C++, Java, Python.
- Name your script as DES.py, DES.java, DES.cpp
- With the homework pdf we also provided **sample_input.txt** and **sample_output.txt.**
- We will use auto grader, so format of the output should match to the format of the provided **sample_output.txt.**
- **Include either comment in the assignment or separate attachment with team members, if you choose to work alone on the assignment.**
- **Submit:** submit only your DES.py or DES.java or DES.cpp file.

Input format

Sample_input.txt will contain a **data block** and a **key**, both 8 bytes long and in hexadecimal format. It will also specify an **operation**, which can either be 'encryption' or 'decryption'.
An example of a simple sample_input.txt file will look as follows:

```
data_block: 0123456789ABCDEF
key: 133457799BBCDFF1
operation: encryption
```

Output format

In your output, you should print **intermediary keys C0-C16 and D0-D16** (where each C and D keys are 28 bits) and **subkeys K1-K16** (each of which is 48 bits).
Your output should also print **L0-L16** and **R0-R16** (each of which is 32 bits) final **ciphertext or decrypted text** which is 64 bits. The format of the output should match the format of sample_output.txt.

We should be able to run your code as follows:

```
DES.py <path to input file> <path to output file>
DES.py ./simple_input.txt ./output.txt
```

./simple_input.txt – will contain the input that we provide, and output should be written to ./output.txt.
For ./simple_input.txt your output in ./output.txt should match to provided ./simple_output.txt.

Rubric

- Correct implementation of encryption 20pts
- Correct implementation of decryption 20pts
- Outputs **C0-C16** 10pts
- Outputs **D0-D16** 10pts
- Outputs **K1-K16** 10pts
- Outputs **L0-L16** 10pts
- Outputs **R0-R16** 10pts
- Readable code 10pts
- **Bonus**: We will also measure the **execution time** of the code and the top two fastest codes for each programming language will get an extra 10pts.

You need to use the following (you can also find it in slides, except s2-s7 boxes).

- For key generation
    - Permutation Choice One - PC-1
    - Schedule of left shits - Number of left shits
    - Permutation Choice Two - PC-2
- For encryption and decryption
    - Initial Permutation - IP
    - Expansion permutation - E Bit Selection Table
    - S-Boxes
    - Permutation function - P
    - Inverse Initial Permutation - IP^-1

(b) Permuted Choice One (PC-1)						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

(c) Permuted Choice Two (PC-2)							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

(d) Schedule of Left Shifts																
Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits Rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Table C.1 Permutation Tables for DES

(a) Initial Permutation (IP)							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

(b) Inverse Initial Permutation (IP^{-1})							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

(c) Expansion Permutation (E)					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

(d) Permutation Function (P)							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Table C.2 Definition of DES S-Boxes

S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11