Hw4

# Problem1

1) Identify where the instructions have been removed (write the step number, e.g., 8.5 if an instruction is missing between 8 and 9) (5 points)

7.5 ， 9.5 , 11.5   
2) Provide the correct instructions to fill the missing steps in order to correctly implement AES (10 points)

7.5 : y , The second array is initialized to n ⊕ k.

9.5 Replace (x0,x1,x2,x3) with (e ⊕ x0,e ⊕ x0 ⊕ x1,e ⊕ x0 ⊕ x1 ⊕ x2,e ⊕ x0 ⊕ x1 ⊕ x2 ⊕ x3).

11.5 modifies y again; modifies x again, using ⊕4; modifies y again; and so on for a total of ten rounds. The constants for the x modifications are 1, 2, 4, 8, 16, 32, 64, 128, 27, 54.

3) For each removed step, explain why it is necessary and how the output would be affected if that step remains skipped (5 points)

7.5, it makes the expanded key correct.

9.5, it generates the expanded key correctly and makes the output more difficult to decrypt.

11.5, it generates the expanded key correctly and makes the output more difficult to decrypt. Because it repeats ten rounds. A 128-bit key requires ten rounds. The longer the key, the more secure the encryption. The trade-off is that the encryption will take much more time.

# Problem2

1) Read and understand the attack explained below (no grade)

2) Extract the results using correlate.c and paste, in your

answers, the *last three lines* (5 points)

command： (tail -4096 study.400 ; tail -4096 attack.400) | ./correlate > attack

194 13 89 81 b6 91 99 95 9a 97 bf a9 a4 a0 94 85 a8 87 b9 a2 9d b7 b2 a7 ac b1 83 b4 8f 88 aa b5 a1 ab 9c ad 84 be 92 80 a3 93 af 9e 8a 9f bb 8e 98 bc 9b 90 ba 86 2e 8d 8c bd b3 b0 ae 11 eb f4 e2 22 8b a5 0b c5 1e 82 ff ea 96 e5 23 c3 b8 a6 00 26 fb 3a 10 32 0a fc 05 dd f3 2d 27 35 04 3b c9 cb e0 0c d7 2b d3 19 29 33 f1 17 ee cf c2 f0 25 2f 3c 14 cd 02 fa 24 0d 3d e4 39 c8 31 d5 f6 d4 f9 e6 e9 f5 03 c1 34 12 ed e3 1c e8 fd d1 3f 1b d8 d2 30 1a 20 f8 1d e1 0f 06 36 da 08 dc 16 2a 21 18 c0 38 ce e7 15 d9 01 fe de 1f ec 0e d6 37 2c ca c6 c7 28 09 13 07 db 3e c4 cc f7 d0 df ef f2 41 5e

256 14 b4 9a 98 b2 8e ad b0 80 93 81 95 a7 b1 a6 b9 af bb 8c b6 a5 90 8b ac 9f 8f be a8 8a 85 b3 a9 84 bf 91 9c aa 86 24 8d 92 96 bd a4 62 a0 82 ba 7e 97 68 87 40 bc b7 89 88 ae 9d b8 a3 67 99 7c 9b 6f 9e ab 83 05 7f 6b 61 6a 57 74 a2 32 43 69 56 76 5c ec 5b 41 65 60 52 4b 4d 58 48 5e 5d 7a 45 6c 7d 75 54 59 73 94 4a 39 3a 0c 49 27 6d b5 77 26 15 70 12 18 71 10 cd 63 30 5a 0e 4c 1b 2b a1 44 e8 17 1f 1a 08 29 6e 04 55 01 66 23 d7 79 e1 38 02 42 0f 34 4e 53 2e 2a 5f 0a 46 ca 13 3c 3f e9 07 3b 0d 11 50 fd 1c e6 00 7b 3e f1 c0 2c 1e 28 d5 3d 47 fe e2 ce db 51 22 f3 0b e0 64 e3 dd eb e4 16 09 21 1d c2 ea 78 36 cc ff 72 c8 c3 33 c9 4f cb cf ee de fc ef 2f d2 31 06 03 f2 d1 2d dc f8 f0 c7 37 f7 e5 d9 fb d6 c4 f6 f5 c6 35 d8 14 d4 c5 20 19 f9 fa d3 c1 e7 ed da df d0 f4 25

256 15 ea ec cd c1 d9 dc dd de d4 f4 f6 ef da fb c2 d7 fd f3 c8 d0 e2 ee 02 e5 b6 e7 f2 d8 db 6e cb f1 ca cf c5 fe f9 e0 5e 21 71 3b a7 c4 f8 c3 98 8f 13 25 8c eb c9 e8 c7 c0 2f 3c 27 79 4b e9 bc e1 5c d3 88 9d bf ba 92 78 7f f0 bb ae b2 0b 42 1a 8e d1 ed 9b c6 f7 15 17 d2 ff 86 3d 80 38 b4 89 68 2c 84 2a 0f df 83 08 39 a8 cc e4 e6 14 d6 55 b9 e3 ce a3 4f 58 85 b0 f5 04 95 a1 8a 8d 65 9f 94 fa 72 99 6d 31 67 be 8b b8 3f 19 56 11 2e 1e fc 91 a0 9e 1f 87 aa ab 70 1d 44 07 28 22 3e 1c 81 63 57 75 64 32 93 d5 82 06 5d 73 20 52 4d 0d ad 50 6c 37 97 6b 5b 00 16 a6 5a 30 bd 29 36 1b 41 7b 76 34 01 43 69 26 a5 b5 4a 7d 60 a4 45 90 a9 a2 0a b1 7e 54 61 12 35 46 7a 9c b3 40 6a 51 24 3a 49 0c 53 74 b7 77 0e 2b 96 2d 48 af 05 4c 47 18 7c 9a 23 62 5f 10 33 6f 66 ac 09 4e 59 03  
3) Find for which bytes the range of possibilities for each k[i] is lower than 256 (5 points)

0 and 13

4) Explain, in their own words, how a particular k[i] was selected as having fewer than 256 possible values during the analysis. This must be a high-level explanation reflecting your understanding of the attack and how to interpret the correlated data (just copying the relevant line of code will yield no point) (10 points)

Because during the study, 'n' is sent after encryption, the time taken for different encryptions is not the same. We can analyze the correlation between the time obtained in the preparation phase and the time in the attack phase. We can generate a correlating relationship for each byte. Some characters clearly take too long or too short to encrypt, so we can exclude the possibility of these characters. The possible key timing is close enough together to be statistically indistinguishable. Code is

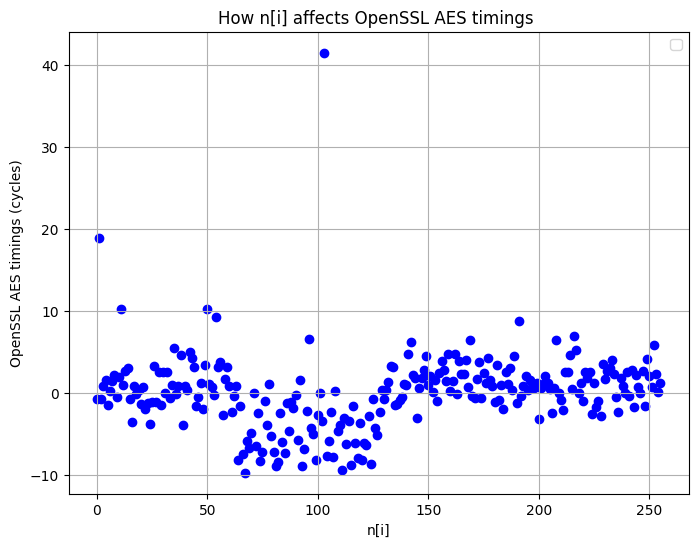
if (c[cpos[0]] - c[cpos[i]] < 10 \* sqrt(v[cpos[i]])) ++numok;

correlated data has 16 lines representing each byte. The first column of the resulting attack file shows the range of possibilities for each k[i] (second column). The rest columns show the corresponding values.

5) For one specific k[i] index which had fewer than 256 possibilities (if there are multiple options, pick one), draw 2 graphs:

We choose k[i] = 0,

a. How n[i] affects OpenSSL AES timings for k = i inside the targeted server (10 points)



b. How n[i] affects OpenSSL AES timings for random selection of k inside the targeted server (10 points)

