Assignment 3 results for Sagar Manubhai Patel

Score for this attempt: **70** out of 100
Submitted 20 Oct 2021 at 21:51
This attempt took 8,809 minutes.

Question 1

20 / 20 pts

As discussed in the lecture on AES, bit strings can represent both numbers and polynomial over $\mathbb{Z}_2$. Specifically, a sequence of 8 bits $b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$ can represent both the number $\sum_{i=0}^{7} b_i 2^i$ and the polynomial $\sum_{i=0}^{7} b_i x^i$. For example, the bit sequence 1,0,0,1,0,0,0,1 can represent the number 145=0x91 and the polynomial $x^7 + x^4 + 1$. Following convention we use the two hexadecimal digit to represent both the number and the polynomial. That is, to represent the polynomial in the example we use 91.

In the following questions you need to compute the value of polynomial addition and multiplication (modulo the AES polynomial $x^8 + x^4 + x^3 + x + 1$). That is, 91+03 is the sum of the polynomials $x^7 + x^4 + 1$ and $x + 1$, which is $x^7 + x^4 + x$. Hence 91+03=92. Similarly, 91×03, the product of the polynomial is $(x^7 + x^4 + 1) \times (x + 1) = x^8 + x^7 + x^5 + x^4 + x + 1$. Reducing by the AES polynomial we get $x^7 + x^5 + x^3$. Hence, 91×03=A8.

The expected output format is two hexadecimal digits, using uppercase letters.

1) 75+A7= d2

2) 56+7B= 2d

3) 37×04= dc

4) 89×0A= 2d

5) 25×81= 38

Answer 1:

Correct! D2

Answer 2:

ou Answered 2d

rrect answer 2E

Answer 3:

Correct! DC

Answer 4:

Correct! 2D

Answer 5:

Correct! 38

Question 2

20 / 20 pts

An AES block of 16 bytes $b_0, b_1, \ldots, b_{15}$ represents a 4×4 matrix:

$$\begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{bmatrix}$$

Assuming the AES state 52,20,B6,19,B4,7C,47,F0,52,20,B6,19,B4,7C,47,F0, what would the state be after the following operations:

1. AddRoundKey with key 42,C7,C8,3C,08,FF,2F,9D,42,C7,C8,3C,08,FF,2F,9D, 10,E7,7E,25,BC,83,
2. SubBytes (you can find the AES S-Box here (https://en.wikipedia.org/wiki/Rijndael_S-box)) 00,B7,4E,D4,8D,10
3. ShiftRows 52,7C,B6,F0,B4,20
4. MixColumns (See here (https://en.wikipedia.org/wiki/Rijndael_MixColumns) for more description) 6B,CA,2E,52,40,75

The expected answer format is a sequence of 16 numbers. Each number consisting of two hexadecimal digits, using uppercase letters. Numbers should be separated by a single comma with no spaces.

Answer 1:

Correct!

10,E7,7E,25,BC,83,68,6D,10,E7,7E,25,BC,83,68,6D

Answer 2:

Correct!

00,B7,4E,D4,8D,10,A0,8C,00,B7,4E,D4,8D,10,A0,8C

Answer 3:

ou Answered

52,7C,B6,F0,B4,20,47,19,52,7C,B6,F0,B4,20,47,19

orrect answer

52,20,B6,19,7C,47,F0,B4,B6,19,52,20,F0,B4,7C,47

Answer 4:

ou Answered

6B,CA,2E,52,40,75,4D,07,6B,CA,2E,52,40,75,4D,07

orrect answer

85,98,4f,d0,63,c4,be,39,85,98,4f,d0,63,c4,be,39

Question 3

20 / 20 pts

When using AES to encrypt the plaintext 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 with a secret key, we denote the (unknown) state after the first AES round with $s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}$. Encrypting the plaintext 01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 we denote the (unknown) state after the first AES round with $s'_0, s'_1, s'_2, s'_3, s'_4, s'_5, s'_6, s'_7, s'_8, s'_9, s'_{10}, s'_{11}, s'_{12}, s'_{13}, s'_{14}, s'_{15}$.

1. What is the value of $s_4 + s'_4$ (where '+' is a polynomial addition) 00 (5 points)
2. For which indices i we have that $s_i \neq s'_i$? 0,1,2,3 (5 points)
3. If $s_0 + s'_0 = 02$, what are the possible values of the first byte of Round Key 0? CE,CF (10 points)

Expected answer formats: (1) two hexadecimal digits, (2) a comma separated list of DECIMAL numbers between 0 and 15, no spaces, (3) a sequence of comma separated numbers, each consisting of two hexadecimal digits. All hexadecimal numbers using uppercase letters. No spaces anywhere.

Answer 1:

Correct!

00

Answer 2:

ou Answered

0,1,2,3

rrect answer

0, 1, 2, 3

Answer 3:

ou Answered

CE,CF

rrect answer

CE, CF

Question 4

10 / 10 pts

Pilsung is a North Korean cipher based on AES. As in AES, the state is represented as a 4×4 matrix of bytes, and the rounds follow the same pattern as AES. Unlike AES, Pilsung uses multiple SBoxes for encryption. Specifically, it uses different SBoxes for each state byte at each round. We use SB[r,b] to denote the SBox used for the byte b (0<=b<16) of round r (0<=r<10). For example SB[3,7] is used for byte number 7 of the state at round 3 of the encryption. Each of the SBoxes consists of 256 entries of size 1 byte each. The SBoxes are stored in consecutive locations in memory.

1. Assuming that cache lines are 64 bytes and that the SBoxes are aligned at the start of a cache line, how many cache lines does each SBox cover? 4
2. Assuming that the cache has 64 sets, which SBoxes fall in the same cache sets as SB[0,0]? SB[0,0],SB[1,0],SB[
3. What is the probability that a cache set is accessed in the SubBytes step of a round? (percents, rounded to the nearest percent) 25
4. What is the probability that a cache set is accessed in the SubBytes step of any of the rounds of the encryption? (percents, rounded to the nearest percent) 94
5. Using a first-round Prime+Probe attack, how many bits can we expect to recover from each key byte? 2

Answer 1:

ou Answered

4

rrect answer

Manual test - 4

Answer 2:

ou Answered

SB[0,0],SB[1,0],SB[2,0],SB[3,0],SB[4,0],SB[5,0],SB[6,0],SB[7,0],SB[8,0],SB[9,0]

rrect answer Manual test - SB[i,0]

Answer 3:

Correct! 25

rrect answer 25%

Answer 4:

Correct! 94

rrect answer 94%

Answer 5:

ou Answered 2

rrect answer Manual test - 2

Question 5

0 / 10 pts

Due to the way Pilsung is implemented, the SBoxes are not aligned with a cache line start. Instead, the SBoxes are shifted by one byte. That is, byte 0 of the SBox is in offset 1 of the cache set. How can you exploit this memory layout to recover all of the bits of Round Key 0 using a first-round Prime+Probe attack?

Your answer:

For the first round attack, we have to generate the 10^6 random plaintext to perform the a synchronous Prime+Probe attack while performing encrypting each cache set to get the baseline cache activity level. By targeting ket bytes, now we can split that results into the corresponding plaintext bytes, from where we can get the average probe times for each of the targeted byte's values. By performing the below algorithm for each of the 16 bytes, we will be able to get the two value for each of the bytes which then we can predict the suitable from the running it on oracle. As it reduces the bits in the entropy however there will be no more detailed information could be found in the first round of the Prime + Probe attack.

.

Unanswered

Question 6

0 / 20 pts

The file https://cs.adelaide.edu.au/~yval/SP21/Assignment3/AESPP.txt (https://cs.adelaide.edu.au/~yval/SP21/Assignment3/AESPP.txt) contains the results of running Prime+Probe on a T-table implementation of AES. Each line in the file shows the results of a single invocation of AES. The format is:

Plaintext Ciphertext CL0 CL1 CL2 ... CL63

Where Plaintext is the plaintext provided to AES, Ciphertext is the ciphertext that AES computed, and CLi is the number of cycles it took to probe cache set i.

Your task is to perform a first-round attack on the data, i.e. find the most significant four bits of each of the key bytes. The xexpected answer format is a comma separated list of hexadecimal digits, such that the first digit is the top four bits of byte 0 of the key, the second digit is the top four bits of byte 1 of the key, and so forth.

Note: Do not share the downloaded file!

You Answered

Correct Answers

Manual check

Question 7

0 / 0 pts

Please upload a tar or tgz file with all the software you used to answer this quiz. (Source files only. Include credits where these are due.)

a1789814as3.tgz
(https://myuni.adelaide.edu.au/files/9834816/download)

Quiz score: **70** out of 100