Aniruddha Jafa March 4, 2019

CS-302 Problem Set 2

Collaborators: *Paul Kurian, Vidur Singh*

Problem 1

Assume: initial XOR in AES is called round 0, following which there are 10 rounds (round 1 to 10). This means there are 11 round keys, although all you do with the round 0 key is an XOR. As suggested in: `https://kavaliro.com/wp-content/uploads/2014/03/AES.pdf` .

a)

round 0: 4 words consumed (initial XOR)
rounds 1 - 4: 4 words consumed each time.

Thus 4 + 4*4 = 20 words consumed.

b)
For round 8, we'll use $w_{32}, w_{33}, w_{34}, w_{35}$

Reasoning:

round 0: 4 words consumed for initial XOR.
rounds 1-7: 4 words consumed each time.

Thus : 4 + 7*4 = 8*4 = 32 words consumed (w_0?w_{31}) so far.

This leaves $w_{32}, w_{33}, w_{34}, w_{35}$ for round 8.

Problem 2

a) 33. 11 times (for rounds 0-10) for each of the 3 blocks.

b) 30. 10 times (only do SUB BYTES for rounds 1-10, not round 0) for each of the 3 blocks.

c) 30. 10 times (only do SHIFT ROWS for rounds 1-10, not round 0) for each of the 3 blocks.

d) 27. 9 times (rounds 1-9 of do mix columns, round 0 and round 10 don't).for each of the 3 blocks.

Problem 3

a) input given: 1111 0101, which is 15, 5 in decimal. Thus, check element in row f, column 5. This is e6 i.e. 1110 0110

b) Input block is: 81cfb5166cee12a75c50660b56357832

We know an input block is 4 words long, each word is 4 bytes. Thus, in column major order, input is: $\begin{bmatrix} 81 & 6c & 5c & 56 \\ cf & ee & 50 & 35 \\ b5 & 12 & 66 & 78 \\ 16 & a7 & 0b & 32 \end{bmatrix}$

After left shift, we get: $\begin{bmatrix} 81 & 6c & 5c & 56 \\ ee & 50 & 35 & cf \\ 66 & 78 & b5 & 12 \\ 32 & 16 & a7 & 0b \end{bmatrix}$

This corresponds to string: 81ee66326c5078165c35b5a756cf120b

c)

Given ciphertext in hex is: 325fb9cdd3e133b7a8975ecdc68353555a8a3cd79bf43f1c079434c6a6ff d0b567287389a192d9ccac94cc6868dd539f5a8a3cd79bf43f1c079434c6a6ffd0b5.

We know each cipherblock block is 4 words long, where each word is 4 bytes (16 bytes, or 128 bits). Thus we get the following four 16-byte blocks :

325fb9cdd3e133b7a8975ecdc6835355
5a8a3cd79bf43f1c079434c6a6ffd0b5
67287389a192d9ccac94cc6868dd539f

5a8a3cd79bf43f1c079434c6a6ffd0b5

Since cipherblock 2 = cipherblock 4, and ECB mode was used, we know that in the plaintext, block 2 = block 4.

Problem 4

See AES code submitted for Q4.

Problem 5

The S-boxes for DSA were modified by the NSA to guard against differential cryptanalysis, which had not become public knowledge at the time. Attacks based on differential cryptanalysis take advantage of the relation between distances of known inputs, and traces output distances across each of the 16 DES rounds, which can be reasoned about probabilistically to deduce the key. The modified Sboxes guard against this by ensuring that even tiny changes in inputs (small input distance) leads to large distances between their respective outputs.

References:
`https://arstechnica.com/information-technology/2013/09/the-nsas-work-to-make-crypto-wors`

`https://web.archive.org/web/20120106042939/http://securespeech.cs.cmu.edu/reports/coppersmith.pdf`

Problem 6

The Pbox provides diffusion to the DES by ensuring the the output from an Sbox in a particular round will be processed by multiple other Sboxes in the next round. Diffusion ensures that even a small change in the input text affects many stages of encryption, which leads to more changes in the cipher text. This increases security by obscuring the relationship between plaintext and cipher texts, which makes things more difficult for an attacker.

References:
https://www.uow.edu.au/~jennie/WEBPDF/142_1990.pdf
https://www.nku.edu/~christensen/diffusionandconfusion

Problem 7

See DES code and README submitted for Q7.