

Homework #2

Due Thursday, January 31, at NOON

Lecture Contents

Secret Key Cryptography

Instructions for Preparation and Submission

- Your answer file format may be .txt (text, any platform), or PDF. Do **not** submit WORD files.
- Submit electronically using submit.ncsu.edu, with filenames hw2.txt or hw2.pdf.
- **Include your name and Student ID #** on the first page
- Do individual work, and submit individually. I do not object if you wish to ask fellow students for help / advice / tips / information, but don't copy someone else's solution.

Problems

DES

1. (5) Suppose the DES F function mapped every 32-bit input R, regardless of the value of the round key, to the 32-bit complement of R. What function would DES then compute? What would the decryption look like?
2. (5) Consider a Feistel cipher composed of 16 rounds with block length 128 bits and key length 128 bits. Suppose that, for a given key k, the key scheduling algorithm somehow (doesn't matter how) determines values for the first 8 round keys $k_1..k_8$, and then sets $k_9=k_8$, $k_{10}=k_7$, ..., $k_{16}=k_1$. Given a ciphertext C, explain how, if you are able to conduct a chosen plaintext attack, you can decrypt C and determine the corresponding plaintext P using just a single chosen plaintext, for this cipher.
3. (5) Are all the 56 bits of the DES key used in an equal number of round keys? Specify, for each bit, how many rounds it is used in, and explain how you got your answer.
4. (5) Show that DES encryption and decryption are identical except for the order of the 48-bit keys. (Hint: running a round backwards is the same as running it forwards but with the halves swapped, and DES has a swap after round 16 when run forwards).
5. (5) Are there any f functions that are simply invalid, i.e., the resulting DES-like cipher cannot be reversed to recover the original plaintext from the ciphertext? Justify your answer. A function that is simply easy to break, but for which the cipher can be reversed, does not count.
6. (10) Using the posted DES program, or writing your own, or using another program (identify where you got it), answer the following questions:
 - (a) Run encryption with an input block of
0x1F1F1F1F1F1F1F1F

and a key of

0x2E2E2E2E2E2E2E2E

(note: parity bits not set to correct parity value)

What result do you get?

(b) Run encryption with a block that differs only in the leftmost bit from the input block of part (a), and use the same key as in part (a). For each round of DES encryption, state how many bits the results of (a) and (b) differ from each other. Does DES exhibit the avalanche effect for this example?

(c) Answer the same questions as (b), but using an input that is the same as that in (a), and a key that differs from the key in (a) in the leftmost bit only.

(I encourage you to do the following by hand, or with code you wrote yourself. It is fine to use the posted DES program to verify your answers before submission, but if you simply "plug-and-chug", you will lose the benefit of doing this problem.)

7. (25) This problem provides a numerical example of encryption using a one-round version of DES. Let the plaintext block be

0xFEDCBA9876543210

and the key be

0x0123456789ABCDEF

(note: parity bits not set to correct parity values)

For the following questions, please give your answers in hexadecimal.

(a) Do the initial permutation of the input and show the result

(b) Derive K_0 , the 56-bit key that results after discarding 8 parity bits and permuting the remaining bits

(c) Derive K_1 , the first round subkey

(d) Derive L_0 , R_0

(e) Expand R_0 to get $E[R_0]$, where $E[\]$ is the expansion function

(f) Calculate $A = E[R_0] \text{ XOR } K_1$

(g) Group the 48-bit result A into eight groups of 6 bits and evaluate the corresponding S-box substitutions for each group.

(h) Concatenate the eight 4-bit results of the above step to get a 32-bit result, B

(i) Apply the permutation to get $P(B)$

(j) Calculate $R_1 = P(B) \text{ XOR } L_0$

(k) Write down the ciphertext output by this first round.

AES

8. (5) Verify the MixColumn result in figure 3-25 of our text, by using the same method (in conjunction with figure 3-28's table) to compute InvMixColumn of the MixColumn result, and checking that you produce the MixColumn input. Show your work.

9. (5) It has been suggested that implementing ciphers, like AES, with table lookups for the basic operations, rather than actual computations, will help thwart timing attacks.

(a) Why is that?

(b) Suggest an alternative that will also help thwart timing attacks.

(Similarly to the above DES problem, I highly recommend you compute the following by hand, or using a short program you write yourself, and then use any programs you wish to confirm your results are right)

10. (20) Given the plaintext block
0x012345678 012345678 012345678 FFFFFFFF
and the key

0xFFFFFFFF 00000000 FFFFFFFF 00000000

- (a) Show the original contents of state, as a 4x4 table with values shown in hexadecimal.
- (b) Show the value of state after the initial AddRoundKey.
- (c) Show the value of state after SubBytes.
- (d) Show the value of state after ShiftRows.
- (e) Show the value of state after MixColumns.

11. (10) Using the posted AES program, or writing your own, or using another program (identify where you got it), answer the following questions:

(a) Run encryption with an input block of
0x00112233445566778899AABBCCDDEEFF

and a key of

0x88888888444444442222222211111111

What result do you get (give answer in hex please)?

(b) Run encryption with a block that differs only in the leftmost bit from the input block of part (a), and use the same key as in part (a). For the result of AES encryption, state how many bits the results of (a) and (b) differ from each other. Does AES exhibit the avalanche effect for this example?

(c) Answer the same questions as (b), but using an input that is the same as that in (a), and a key that differs from the key in (a) in the leftmost bit only.

EXTRA CREDIT

(15) Implement your own DES cracking system. You are allowed to use any existing program or library you wish that does DES encryption. You must add the part that does the cracking (e.g., brute force key trials, if that is what you use). Explain how your cracking system works, and use it to find what DES key maps

0xF0F0F0F0F0F0F0F0

to

0xC9C056814E213B9C

Show your answer as a 56-bit key, in hex, without the parity bits shown.

Explain your cracking system, and how long it took your program to find the answer.