

## Homework #3

Due Tuesday, February 19 at 11:45PM

### Lecture Contents

Message block chaining, MACs, Triple DES, Message Digests, MD5, SHA-1

### 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 hw3.txt or hw3.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 (some problem numbers intentionally omitted)

2. (7) An authentication algorithm is to use CBC with DES, with an initialization vector of 0, and the message to be authenticated as input, with the residue used as the MAC. Show that the same result can be produced using CFB mode instead.

3. (7) Suppose a message with three 128-bit plaintext blocks P1, P2, and P3 is encrypted in the following way, using a shared key K:

$$C1 = P1 \text{ XOR MD5}(K \mid IV)$$

$$C2 = P2 \text{ XOR MD5}(K \mid C1)$$

$$C3 = P3 \text{ XOR MD5}(K \mid C2)$$

Explain how this message is decrypted.

4. (7) Below is the description of the New Block Chaining Mode (NBC). For message block  $M_i$  and ciphertext block  $C_i$ ,

$$C_i = E_K(M_{i-1} \text{ XOR } M_i) \text{ XOR } C_{i-1}$$

$C_0$  and  $M_0$  are constant initialization vectors (IVs).

- (a) Describe how decryption is accomplished.
- (b) If a 1-bit error occurs in ciphertext block  $C_i$ , what plaintext blocks will be affected?
- (c) Is it possible for an attacker to modify the ciphertext in such a way that some predictable change will occur in the plaintext?
- (d) Does information about the plaintext leak to the ciphertext output?
- (e) Is it possible to encrypt blocks in parallel?
- (f) Is it possible to decrypt blocks in parallel?

5. (7) Consider the following alternative method of encrypting a message. To encrypt a message, use the algorithm for doing a CBC decrypt. To decrypt a message, use the algorithm for doing a CBC encrypt.

- (a) Would this work?

(b) What are the security implications of doing this, if any, as contrasted with the "normal" CBC?

6. (7) What is a practical method for finding a triple of keys that maps a given plaintext to a given ciphertext using EDE? (Hint: it is like the meet-in-the-middle attack)

8. (7) Suppose it is desired to use an error correction code along with encryption of message, to provide both error recovery, and message confidentiality. What are the benefits of

(a) encrypting the message and computing the ECC of the ciphertext result and transmitting  $E(M) \parallel ECC(E(M))$ , versus

(b) computing the ECC of the of the plaintext message, and transmitting the encrypted version of the combination,  $E(M \parallel ECC(M))$ , versus

(c) computing the ECC of the plaintext message and transmitting  $E(M) \parallel ECC(M)$ ?

9. (7) Consider using encryption with a known key to compute a hash function. Process a message as follows:

Encrypt the first plaintext block, XOR the result with the second block, encrypt this result, etc.

Show that this is not secure by solving the following problem. Given a two-block message  $M_1, M_2$ , and its hash  $H(B_1, B_2) = E(E(B_1) \text{ XOR } B_2)$ . For an arbitrary block  $X_1$ , find a block  $X_2$  so that  $H(X_1, X_2) = H(B_1, B_2)$ .

10. (7) For each of the following suggested message digest functions, identify if it has the "desirable properties" for a digest.

(a) Adding the squares of the blocks together and using the sum

(b)  $H_i = E_{\{M_i\}}(H_{\{i-1\}} \text{ XOR } H_{\{i-2\}})$ , use the last output as the message digest (and  $H_0$  and  $H_{\{-1\}}$  are constants)

12. (7) How do you decrypt the encryption specified in section 5.2.3.2 of the text (Mixing in the Plaintext)?

15. (7) Assume AND, OR, XOR, PLUS, INVERT, and LEFTROTATE all take about the same amount of time. Estimate the relative performance of MD5 and SHA-1.

17. (7) What value or useful property, if any, is provided by appending the message length to a message before it is hashed?

18. (7) Show what the new value of A/B/C/D/E will be after step  $t=18$  if the old value (i.e., input to step  $t=18$ ) is

A/B/C/D/E = 0x00000000 0x11111111 0x22222222 0x33333333 0x44444444  
and  $W_0 \dots W_{15} =$

0x 00000000 11111111 22222222 33333333 44444444 55555555 66666666 77777777  
88888888 99999999 AAAAAAAAAA BBBBBBBB CCCCCCCC DDDDDDDD EEEEEEEE  
FFFFFFFF

19. (7) The text suggests how to generate a pseudo-random stream of MD-sized blocks in section 5.2.3.1. This stream must eventually repeat. Will the first block necessarily be the first to be repeated? How can you tell, or why can you not tell?

20. (7) If  $d_0 = 0x1111$ ,  $d_1 = 0x2222$ ,  $d_2 = 0x4444$ , and  $d_3 = 0x8888$ , and  $m_2 = 0x01234567$ , what will be output by the first step of the third pass of MD5?

### EXTRA CREDIT

Using a 64-bit key  $K = 0x\ 01234567\ 89ABCDEF$ , and the SHA-1 method of computing a message digest, compute the HMAC of the key and the message if the message is the 400-bit message consisting of alternating 1's and 0's, starting with a 1. You are welcome to use any program you like to generate the SHA-1 results, but implement your own program for HMAC, and include your code.

### EXTRA EXTRA CREDIT

Using any SHA-1 implementation you wish:

- a. Test whether SHA-1 has the avalanche property
- b. Do research on how "random" the output appears. Do research first on how to test for randomness. You can use any tool you wish that does randomness testing, but understand the tests it is using. Use as input to SHA-1 randomly-generated bit strings of size 512-65 bits long and then pad as required by the SHA-1 standard. Indicate what SHA-1 implementation you used, and how you tested for randomness, as well as what your results show.

### EXTRA EXTRA EXTRA CREDIT

Suppose a message digest MD6 is implemented as follows.

- Messages are not padded, but must be multiples of 512 bits in length.
- The initial message digest, which is input to the processing of the first message block, is set to all 1's.
- MD6 does just a single pass of processing, equal to the first pass of MD5, but in that pass, only the first 4 steps (instead of 16) are computed, resulting in new values for  $d_0..d_3$ .
- The shift amount ( $s$ ) in every step is always 1.
- The constants  $T[1]$  through  $T[64]$  are all set equal to  $0x5555555555555555$ . In other words, MD6 is a much simpler message digest than even MD5. Try to find two messages that will produce the same hash value using MD6. Explain your strategy or finding two such messages, and show the results (both input, and output), if you can find two such messages, in hex.

How much harder do you think each of the following would make your attempt:

1. Using the actual values of  $T[1]...T[64]$ ?
2. Using the standard shift amounts
3. Doing all 16 steps of pass 1.
4. Using the standard initial message digest value.
5. Doing the standard message padding.