

Problem Set 1 (due September 26, 2004)

Submit by email to: tmorgan@ccs.neu.edu ; CC: noubir@ccs.neu.edu

Problem 1 [10 points]:

1. Describe an encryption scheme that you might have used as a kid to communicate securely, and that you thought is fairly secure. If you never did, then imagine a scheme or discuss with your little cousin \mathcal{J}
2. Discuss the deficiencies of that scheme and how it might be broken.

Problem 2 [20 points]:

(a) Evaluate the following expression in the most efficient way possible. Show how you did it!

• $4457 \bmod 17 =$

• $(-4457) \bmod 17 =$

• $3^{64} \bmod 17 =$
Notice that $64 = 2^6$.

• $3^{65} \bmod 17 =$
Notice that $65 = 2^6 + 1$.

(b) Devise an efficient algorithm for computing $a^b \bmod c$, where a , b , and c , are positive integers. Use the binary encoding $b_1b_2\dots b_n$ for b .

Problem 3 [30 points]:

Textbook problem 1.21, page 40 [except the c question (affine cipher)].

Problem 3 [20 points]:

Theorem 2.1 (page 9) states that if the prime power factorization of a number is

$$m = \prod_{i=1}^n p_i^{e_i}, \text{ where } p_i\text{'s are distinct primes and } e_i > 0, \text{ then the number of integers in } Z_m$$

that are relatively prime with m is: $\phi(m) = \prod_{i=1}^n (p_i^{e_i} - p_i^{e_i-1})$.

In class I have shown this theorem for m that is a power of a prime. Prove this theorem for the general case.

Problem 4 [20 points]: Excel Worksheet Protection

<http://chicago.sourceforge.net/devel/docs/excel/encrypt.html>

MS Excel uses a very simple scheme to protect a sheet from modification. An Excel sheet can be protected using the menu: Tools -> Protection -> Protect Sheet.

Protection Scheme: the protection password is hashed using a simple scheme then stored in the file. Let the password be $p_1p_2\dots p_k$. The hash H is computed as follows:

- $H = (p_1 \ll_{15} 1) \oplus (p_2 \ll_{15} 2) \oplus (p_3 \ll_{15} 3) \oplus \dots \oplus (p_k \ll_{15} k) \oplus k \oplus 0xCE4B$.
- H is a 16 bits word (2 bytes) and is stored in the file.
- To unprotect the sheet Excel requests a password, computes its hash and verifies that it equals the stored hash.

$a \ll_{15} b$ denotes shifting (w/ rotation) the ascii code of character 'a' to the left 'b' times using only the lowest 15 bits and rotating the highest bits (exceeding 15) to the right.

Example: 'j' = 0x6A; 'a' $\ll_{15} 10 = 0x2803$.

Example: password "test" of 4 characters leads to hash 0xCBEB.

- a -> $0x61 \ll 1 == 0x00C2$
- b -> $0x62 \ll 2 == 0x0188$
- c -> $0x63 \ll 3 == 0x0318$
- d -> $0x64 \ll 4 == 0x0640$
- e -> $0x65 \ll 5 == 0x0CA0$
- f -> $0x66 \ll 6 == 0x1980$
- g -> $0x67 \ll 7 == 0x3380$
- h -> $0x68 \ll 8 == 0x6800$
- i -> $0x69 \ll 9 == 0x5201$ (unrotated: 0xD200)
- j -> $0x6A \ll 10 == 0x2803$ (unrotated: 0x1A800)

$$H = 0x00C2 \oplus 0x0188 \oplus 0x0318 \oplus 0x0640 \oplus 0x0CA0 \oplus 0x1980 \oplus 0x3380 \oplus 0x6800 \oplus 0x5201 \oplus 0x2803 \oplus 0x000A \oplus 0xCE4B = 0xFE11$$

1. Protect an Excel using password 'test'. What is the value of the hash H ?
2. Use a Hex Editor (such as: Hex Edit: <http://www.expertcomsoft.com/>) to open the file you protected.
 - a. Identify where the 2 bytes of the hashed password are located? Hint: remember that there are two system formats: Little-Endian and Big-Endian (http://www.webopedia.com/TERM/b/big_endian.html). Which one is used by Excel on PC?
 - b. How can you identify the location of where the hashed password is stored in any file? Hint: look at the sequence of characters preceding/following the hash. What can you say about it? How about the password of other worksheets.

3. Find two passwords different from 'test' that can allow you to unprotect a sheet protected with password 'test'. You might find helpful to use the table of ascii encoding: <http://www.cplusplus.com/doc/papers/ascii.html>.
4. Discuss why the excel protection scheme is weak.
5. **Bonus points:** Write a program that automatically unprotect all sheets in an excel file.