HW 7 CMSC 456. Morally DUE Oct 21
SOLUTIONS
NOTE- THE HW IS FOUR PAGES LONG

1. (0 points) READ the syllabus- Content and Policy. What is your name? What is the day and time of the midterm?

2. (25 points) This is a programming assignment. You will write code that uses the low-e attack to crack a message m encrypted with RSA.

   (a) Begin by inputting multiple lines from standard input. On the first line, the value e will be given, and on the second line, the value L will be given. (Note that e might be larger than L.) The first two lines will be followed by $2L$ more lines. The next L lines will consist of the values $N_1, N_2, \ldots, N_L$ (one value on each line). You can assume that N_i is relatively prime to N_j for each $i \neq j$. The last L lines will consist of the values $x_1, x_2, \ldots, x_L$ (one value on each line), where each $x_i \equiv m^e \pmod{N_i}$.

   (b) You will print two lines to standard output. First, you will use the Chinese Remainder Theorem and the input provided to calculate $m^e \pmod{N_1 \cdots N_L}$; call this value x. (Recall that $0 \leq x < N_1 \cdots N_L$.) Print this value on the first line.

   After that, you will check to see if x has an positive integer e-th root. If it does, then print the root on the second line; this should be the value m. (This may exist even when $e > L$.) If this root does not exist, then print "failed" on the second line. (This will happen only when $e > L$.)

   NOTE: The integer values given via input will be of arbitrary length, so make sure the language you are using supports arbitrary-length integers. Wtih that said, be careful about how you are checking for / finding roots, as built-in functions may give you unsatisfactory results. For a simple algorithm that computes integer roots without loss of precision, see `https://stackoverflow.com/a/15979957`.

GO TO NEXT PAGE FOR MORE INFO ON THIS PROBLEM

You may use C, C++, Java, Python2/3, and Ruby for this problem. You will be submitting a zip file containing all code files you used to complete this problem to the Gradescope assignment called

"hw07 - problem 2".

Upon submission, your code will be automatically run on a Linux machine and tested against various test cases to ensure correctness. You are allowed to submit your code as many times as you want. As with previous programming assignnments, upload a bash script called `run` and (if necessary) another bash script called `build`. These files must begin with the shebang `#!/usr/bin/env bash` on the very first line. If you have any questions or confusions, or if you encounter any technical difficulties, feel free to ask for help on Piazza.

GOTO NEXT PAGE

3. (25 points)

   (a) (6 points) Write Rabin's Encryption algorithm (the original version, not the one modified).

   (b) (6 points) What is the big advantage of Rabin's Encryption?

   (c) (6 points) What is the big disadvantage of Rabin's Encryption?

   (d) (7 points) Give a scenario where that disadvantage is not a problem. (We assume that Eve ONLY has access to seeing what Bob sends. She CANNOT trick Bob into sending anything.)

   SOLUTION TO PROBLEM THREE

   (a) Omitted

   (b) The big adv is that breaking Rabin is equivalent to factoring.

   (c) The big disadvantage is that Alice decodes and may get several possibilities for what the message is.

   (d) If Bob is sending ENGLISH texts (or something else easily recognized) then when Alice gets several decodings she can tell which one it's supposed to be.

   END OF SOLUTION TO PROBLEM THREE

   GOTO NEXT PAGE FOR NEXT PROBLEM

4. (25 points) (You can assume there is an algorithm that will, given A, B rel prime, can find $A^{-1} \pmod{B}$.)

   Write a program in pseudocode to do the following (this is the $L = 3$ case of CRT).

   We call a set of N_1, N_2, N_3 JUSTINIAN if

   N_1 is rel prime to N_2N_3

   N_2 is rel prime to N_1N_3

   N_3 is rel prime to N_1N_2

   Given N_1, N_2, N_3 JUSTINIAN and x_1, x_2, x_3, show that there exists x such that

$$\begin{aligned} x &\equiv x_1 \pmod{N_1} \\ x &\equiv x_2 \pmod{N_2} \\ x &\equiv x_3 \pmod{N_3} \end{aligned}$$

   SOLUTION TO PROBLEM FOUR

   (a) Input $N_1, N_2, N_3, x_1, x_2, x_3$
   (b) Find the inverse of N_1N_2 mod N_3. We call this $(N_1N_2)^{-1}$.
   (c) Find the inverse of N_1N_3 mod N_2. We call this $(N_1N_3)^{-1}$.
   (d) Find the inverse of N_2N_3 mod N_1. We call this $(N_2N_3)^{-1}$.
   (e) Output

$$x_1 \times (N_2N_3)^{-1}N_2N_3 + x_2 \times (N_1N_3)^{-1}N_1N_3 + x_3 \times (N_1N_2)^{-1}N_1N_2.$$

   END OF SOLUTION TO PROBLEM FOUR

GOTO NEXT PAGE

5. (25 points) (We do this problem in BASE 10. Replace $\oplus$ with addition of digits mod 10.) Alice and Bob are doing the Blum-Goldwasser cryptosystem with $p = 1019$, $q = 1051$ (remember, this is in base 10, so p, q are of length 4), $r = 5432$, and $m = 8761$. What does Bob send? Show all of your work.

SOLUTION TO PROBLEM FIVE

$N = 1019 \times 1051 = 1070969$.

$(m_1, m_2, m_3, m_4) = (8, 7, 6, 1)$.

We generate the r's and hence the b_i's.

All $\equiv$ is mod 1070969.

$r = 5432$.

$x_1 = 5432^2 \equiv 590461$, hence $b_1 = 1$.

$x_2 = 590461^2 \equiv 944261$, hence $b_2 = 1$.

$x_3 = 944261^2 \equiv 20985$, hence $b_3 = 5$.

$x_4 = 20985^2 \equiv 201966$, hence $b_4 = 6$.

$x_5 = 201966^2 \equiv 268853$.

Bob computes the following (all arithmetic mod 10)

$(b_1+m_1, b_2+m_2, b_3+m_3, b_4+m_4) = (1+8, 1+7, 5+6, 6+1) = (9, 8, 1, 7)$.

He then sends $((9, 8, 1, 7), x_5 \equiv 268853)$.

END OF SOLUTION TO PROBLEM FIVE