YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CPSC 467b: Cryptography and Computer Security — *Handout #7*

Professor M. J. Fischer — *March 20, 2013*

Problem Set 4

Due on Thursday, April 4, 2013.

Instructions

Work the problems below, prepare your answers in electronic form, and submit your solutions using the submit script on the Zoo. Remember to specify "4" for the problem number argument to `submit`.

Some of the problems require computation. You may use a calculator or computer for the calculations, but you **must** show your work.

Some of the problems use terminology that we have not covered in class, even though we have talked about the concepts. You may use external resources to find out what these terms mean. As always, you must properly cite all resources that you use to solve the problems.

Part A. Written problems

Problem 1: Elliptic Curves Arithmetic [Cf. textbook, p. 370, problem 16.7-2]

Consider the elliptic curve $E : y^2 \equiv x^3 - 2 \pmod{7}$.

(a) Is E a singular curve? Why or why not?

(b) What is the order of E?

(c) List the points on E.

(d) Find the sum $(3, 2) + (5, 5)$ on E.

(e) Find the sum $(3, 2) + (3, 2)$ on E.

Problem 2: EC ElGamal

Happy Hacker and Clever Charlie discovered the benefits of Elliptic Curve Crypto and decided to use EC ElGamal. They chose an elliptic curve $E : y^2 \equiv x^3+x+6 \pmod{11}$ and a point $\alpha = (2, 7)$ on E. However, they need a little help to exchange a message.

(a) Happy chose $a = 7$ as his private key. Calculate Happy's public key β.

(b) Charlie wants to send a message $M = (10, 9)$, already expressed as a point on E, to Happy. Encrypt M using $k = 3$ to obtain a ciphertext C that Charlie will send to Happy.

(c) Decrypt C and show that Happy can recover Charlie's message.

Problem 3: Elliptic Curve Crypto [Cf. textbook, p. 370, problem 16.7-8]

(a) Devise an analog of the following procedure based on elliptic curves rather than on discrete logs. (See textbook, p. 215, problem 7.6-8(a).)

> Suppose you have a random 500-digit prime p. Suppose some people want to store passwords, written as numbers. If x is the password, then the number $2^x \bmod p$ is stored in a file. When y is given as a passwords, the number $2^y \bmod p$ is compared with the entry for the user in the file.

(b) Suppose someone gains access to the password file. Why is it hard to deduce the passwords?

Problem 4: ElGamal Signature Scheme

Happy Hacker was having trouble understanding the ElGamal signature scheme presented on slide #26, lecture 12. He didn't see why the signer should bother choosing a random y in step 1 and decided instead to simply fix $y = 1$. Help Happy out and explain why this is not a good idea.

Problem 5: Security of Digital Signatures

(a) What is existential forgery of a digital signature scheme?

(b) What are practical mechanisms to prevent existential forgery? Describe the mechanisms and explain why it would be hard to produce a forged message-signature pair.

(c) Is the ElGamal signature scheme susceptible to the existential forgery attack? If yes, show how an attacker can produce a valid message-signature pair. If no, explain why the scheme is resistant to this attack. State all assumptions you make.

Part B. Programming problem

Problem 6: Factoring the RSA modulus knowing the public and private keys

Bob's public RSA key is $n = 1501$, $e = 323$. Eve manages to learn that his decryption key is $d = 539$. Implement the randomized factoring algorithm shown on slide #33, lecture 10 notes. Use your program to factor n. Once you have the factorization of n, compute $\phi(n)$, and check your answer by verifying that $ed \equiv 1 \pmod{\phi(n)}$.