

**CSCI 284-162 - Cryptography - Fall 2009**  
**George Washington University**

**Final**  
 due 11 May

**Policy on collaboration:** All examinations, papers, and other graded work products and assignments are to be completed in conformance with The George Washington University Code of Academic Integrity. You may *not* discuss this Final among yourselves. Each student is expected to work independently on his or her own Final; you may not collaborate with others, you may not copy one another's assignments, even in part. You may only refer to your class notes, the text book, and to class material such as slides, notes and links provided on the course website. You may not refer to any other material while working on the Final.

*Any violations will be treated as violations of the Code of Academic Integrity.*

*Show all work. You will be graded almost entirely on how you get to the solution, and not on whether you get there. More efficient and elegant solutions will receive higher marks.*

**For CS 162, undergraduate cryptography, it is sufficient to do three of the following four problems for full credit. Students in CS 284 must do all four for full credit.**

1a. Consider the elliptic curve  $y^2 = x^3 + 1$  over the reals. Compute the value of  $5P$  for  $P = (2, 3)$  using efficient exponentiation showing all steps.

1b. Let  $P$  be a point on an elliptic curve over the reals (any one, not necessarily the curve in part a). Suppose that  $P$  is not the point at  $\infty$ . Give a geometric condition that is equivalent to  $P$  being a point of order (a)2; (b)3; (c)4.

2. Let  $p$  be a prime number  $p = 3 \pmod{4}$  and let  $E$  be the elliptic curve  $y^2 = x^3 + ax + b \pmod{p}$ . Find a polynomial time algorithm that, given  $x \in \mathcal{Z}_p^*$ , computes a point  $(x, y)$  on  $E$  if it exists. Hint: show that, if  $a$  is a square  $\pmod{p}$ ,  $a^{\frac{p+1}{4}}$  is a square root of  $a \pmod{p}$ . Using this algorithm, find the point  $(2, y)$  on the curve  $y^2 = x^3 + x^2 + 1 \pmod{111119}$ . Determine its big-oh complexity.

3. Suppose you are given an algorithm that determines the least significant bit of an RSA-encrypted plaintext from its ciphertext, i.e.  $plsb(y)$  determines the lsb of  $x$  such that  $y = E_K(x)$ . Suppose you are also given the ciphertext corresponding to the plaintext 2, i.e. you are given  $E_K(2)$ . How would you determine the plaintext  $x$  without factoring the RSA modulus?

4. It is trivial that public-key encryption cannot be perfectly secret. If  $\langle K, \hat{K} \rangle$  is a public-private key pair, where  $K$  is the public key, and assuming the system is not trivial, show (in a few lines each) that:

A.  $p(m|E_K(m)) \neq p(m)$  and

B.  $p(\hat{K}|K) \neq p(\hat{K})$ . Hint: if  $K$  is known, so is  $E_K(m)$  for any  $m$ .

You may not assume that, given a public key, a private key is uniquely defined.

Notice that information leakage regarding the private key does not require message encryption by the private key, the public key itself leaks the information.