GEORGIA INSTITUTE OF TECHNOLOGY
School of Electrical and Computer Engineering

ECE 6280
Cryptography

HWK #3, Due: Tuesday March 4, 2014

Problems 1 Let p be a prime. Show that $x^b = x \mod p$ has $d = \gcd(p-1, b-1)$ distinct solutions.

Problem 2: Let p be a prime. Prove that if a is not divisible by p and if $n \equiv m \bmod (p-1)$, then $a^n \equiv a^m \bmod p$.

Problem 3: Let p be a prime and $p \equiv 3 \bmod 4$. Let y be an integer and $x \equiv y^{(p+1)/4} \bmod p$. Show the following:

(a) If y has a square root mod$p$, then the square roots of y mod p are $\pm x$.

(b) If y has no square root mod$p$, then $-y$ has a square root mod$p$, and the square roots of $-y$ are $\pm x$.

Problem 4: Suppose we encipher messages by the rule $C = M^k \mod p$, where p is a large prime, with $1 \leq M \leq p-1$, and k is an integer with $1 \leq K \leq p-1$. Show that, if k is chosen to be coprime with $p-1$, then the decryption algorithm $d(C) = C^D \bmod p$ is correct in that, with $D = k^{-1} \bmod (p-1)$, we have $d(C) = M$.

Problem 5: *Probabilistic Primality Testing:* Let SQRT(a, p) be a (probabilistic) polynomial time algorithm with the property that on inputs a, p it outputs an integer $x \in \{0, \ldots, p-1\}$, such that if p is a prime and if $a \in QR_P$ then $x^2 \equiv a \mod p$. If any of the conditions does not hold then the output value can be anything. Consider the following probabilistic primality test, which takes as input an odd integer p and outputs either "prime" or "composite".

(1) Test if there exist integer $b, c > 1$ such that $p = b^c$. If so, output "composite".

(2) Choose $r \in \{1, \ldots, p-1\}$ uniformly at random and set $a = r^2 \mod p$.

(3) Compute $x = \text{SQRT}(a, p)$.

(4) If $x \equiv r \mod p$ or $x \equiv -r \mod p$, then output "prime". Otherwise, output "composite".

(a) Is the above algorithm polynomial time? Explain your answer.

(b) What is the probability that the above algorithm makes an error when p is really a prime? Prove your answer.

(c) What is the probability that the above algorithm makes an error when p is really a product of two distinct primes? Prove your answer.