CS 408 – Cryptography & Internet Security (Fall 2024)

Assignment #2

(due November 21, 2024, 2:30pm)

Problem 1 (25 points)

Let n be a positive integer. The *Euler function* $\phi(n)$ is defined as the number of positive integers smaller than n that are relatively prime to n.
Thus, if p is a prime, then $\phi(p) = p - 1$. Prove that $\phi(p^c) = p^c(1 - \frac{1}{p})$, for any positive integer c.

(Note that you cannot just apply the general formula for $\phi(n)$ when $n = p^c$. You need to *prove* this formula for $n = p^c$. In general, there are two common ways to prove this: 1) use the technique of mathematical induction, or 2) count the numbers, with enough details and arguments to justify a correct counting of all the numbers)

Problem 2 (25 points)

Let a, b, e_1, e_2, n be publicly-known positive integers, such that $gcd(e_1, e_2) = 1$ and $n = pq$ (where p and q are large primes numbers that are kept secret).

Show that if a and b are chosen such that:

$$a^{e_1} = b^{e_2} \bmod n$$

then anyone can compute c such that $c^{e_1} = b \bmod n$.

Problem 3 (25 points)

(a) The exponents $e = 1$ and $e = 2$ should not be used in RSA. Why? (argue why for each exponent)

(b) Show that if $n = 35$ is used as an RSA modulus, then the encryption exponent e always equals the decryption exponent d.

(c) Suppose you encrypt message m by computing $c \equiv m^3 \pmod{101}$. How do you decrypt? (That is, you want to determine a decryption exponent d such that $c^d \equiv m \pmod{101}$; note that 101 is prime).

(d) Let p be a large prime. Suppose you encrypt a message x by computing $y \equiv x^e \pmod{p}$ for some (suitably chosen) encryption exponent e. How do you find a decryption exponent d such that $y^d \equiv x \pmod{p}$?

Problem 4 (25 points) The textbook RSA encryption scheme is *deterministic* (if the same message m is encrypted twice, then we get the same ciphertext). Moreover, when the set of possible plaintext messages is small, one can simply check if a ciphertext is an encryption of all possible messages. This means that textbook RSA cannot offer confidentiality.
(Note: Even though in RSA a message m must satisfy $0 \leq m < n$, a small message set does not imply a small n. For example, n can still be very large, but in a practical application m may only take the value 1 or 2 out of of a very large set of integers between 0 and $n-1$.)

Consider instead the following scheme. Let (e, n) be an RSA public key, with $n = pq$, and let (d, p, q) be the secret key, with $ed \equiv 1 \bmod \phi(n)$. To encrypt a message $m \in \{0, 1, 2, \ldots, n-1\}$, compute a random $r \in \mathbb{Z}_n^*$ and one of the following encryption pairs (all operations are modulo n):

(a) $[A = r^e, B = m + r]$. To decrypt, compute $B - A^d$.

(b) $[A = r, B = (m + r)^e]$. To decrypt, compute $B^d - A$.

Do any of these encryption pairs improve the security of textbook RSA? Why?