CS 3510 - Design & Analysis of Algorithms

Co-Instructor : Prasad Tetali, office: Skiles 132, email: tetali@math.gatech.edu

Lecture on Monday, February 5, 2018:

The general overview for this week : *A main objective is to describe the RSA cryptosystem and discuss its correctness and efficiency; and towards this objective, to review and develop the necessary modulo arithmetic:* Computing the GCD of two numbers, finding inverse of an integer mod N, modular exponentiation, testing whether an integer is *prime* etc.

I. *The Rivest-Shamir-Adleman (RSA) cryptosystem*

a) Motivation: Why use encryption, decryption or *Public-key* Crypto? Why not some simple *substitution* cipher?

One explanation – the story about the Cal State Prison cipher, Stanford stats team (Persi Diaconis and students) breaking it using consecutive-letter frequency data and a (Metropolis) Markov chain on permutations with random transpositions of letters.

b) High-level description of the RSA Cryptosystem: In the following, M, e, d and N are all (positive) integers:

Alice encrypts the message M that she wants to send to Bob:

Encryption – given a (digital) message M, compute via modular exponentiation an encrypted message:

$$E(M) = M^e (mod\ N)$$

Bob receives $E(M)$ and recovers M using the following decryption:

Decryption – involves another modular exponentiation on $E(M)$:

$$D(E(M)) = (E(M))^d \equiv (M)^{ed} \equiv M (mod\ N)\,.$$

Several natural questions arise:

Q1. What does it mean to say $a \equiv b \pmod{m}$, when a, b and m are all integers.

Ans. It means m divides $(a - b)$ evenly with remainder zero. It also means the remainder obtained, when we take away as many multiples of m as we can from a, is the same as the remainder obtained when we take away as many m's as we can from b.

Q2. How to perform modular exponentiation, $M^e \pmod{N}$?

Ans. Suppose N is an n-bit integer, and we may also supposed that N is the largest of M, e and N. Then we shall see that this can be done using $O(n)$ steps (of squaring or multiplying by M), and each operation will take $O(n^2)$ bit operations. Thus in all, we will use $O(n^3)$ bit operations. (Of course, these are general facts, not just for the special RSA parameters.)

Q3. Is N in the RSA system specially chosen?

Ans. Yes! $N = pq$, where p and q are large primes.

Q4. Do e and d in the RSA system have to be specially constructed and do they depend on $N = pq$?

Ans. Yes, in fact e and d are "inverses" of each other, not with respect to mod N, but instead:

$$ed \equiv 1\ (mod\ (p-1)(q-1))\,.$$

This raises the further questions: when do inverses exist and why $(p-1)(q-1)$? We shall see the answers shortly...

Q5. How does one find large primes?

Ans. Towards this, we will first discuss Fermat's little theorem (see below).

II. *Fermat's little theorem.* For p: prime and a such that $1 \leq a \leq p-1$, we have $a^{p-1} \equiv 1 \pmod{p}$.

Proof. *Step 1.* The key to the proof is first to observe that there is a 1-1 correspondence between $\{1, 2, \ldots, p-1\}$ and the set of integers obtained by multiplying each of these by a and taking (mod p): namely the set $\{a \ mod \ p, 2a \ mod \ p, \ldots, (p-1)a \ mod \ p\}$.

To illustrate with an example, consider $p = 7$ and $a = 4$. Then the two sets in question are: $\{1, 2, 3, 4, 5, 6\}$ and the set with each element multiplied by 4, namely the set:

$$\{4, 8, 12, 16, 20, 24\} \text{ which when reduced mod 7 would becomes the set } \{4, 1, 5, 2, 6, 3\}.$$

So how to see more generally that there is a 1-1 correspondence between these two sets? Note that this set $\{a \ mod \ p, 2a \ mod \ p, \ldots, (p-1)a \ mod \ p\}$ has $p-1$ elements, each between 1 and $p-1$. So if we show that they are all distinct, then we would be done!

Suppose not. Then there must be two distinct $1 \leq i, j \leq p-1$, which give us a "collision": $ia \equiv ja \pmod{p}$.
But then, since $GCD(a, p) = 1$, we may cancel a from both sides (in other words, multiply both sides by a^{-1}), and conclude that $i \equiv j \pmod{p}$. This contradicts the fact that i and j are distinct and both are between 1 and $p-1$.

Step 2. We may now complete the proof easily: Simply multiply the elements in each set. Since they are the same set mod p, we may say:

$$\prod_{i=1}^{p-1} i \equiv \prod_{i=1}^{p-1} (i \times a) \pmod{p} \equiv a^{p-1} \prod_{i=1}^{p-1} i \pmod{p}.$$

Since each of the integers i less than p is coprime to p (meaning has GCD equal to 1), we may cancel $\prod_{i=1}^{p-1} i$ from both sides, completing the proof of Fermat's little theorem. □

III. *Euler's ϕ function.* The question is if we can generalize any of the above stuff that we have done, for the prime case, to arbitrary positive integers. It turns out there is in fact a way to do it for an arbitrary integer n, as long as we restrict ourselves to integers in the range $[1..n]$ that are *coprime* to n – namely those that have GCD with n to be equal to 1. Towards this, let

$$\phi(n) = \#\{m : \ 1 \leq m \leq n, \text{ such that } GCD(m, n) = 1\}.$$

So clearly, $\phi(p) = p - 1$. It is also easy to derive the following facts.

Theorem. i) $\phi(pq) = (p-1)(q-1)$, when p and q are primes.
ii) $\phi(p^k) = p^k - p^{k-1} = p^{k-1}(p-1)$, where p is prime and $k \geq 1$ is an integer.

Proof. i) Let $N = pq$, and let us count the number of integers that are *not* coprime to N. Any integer that is not coprime to N must have a factor either p or q in it. So such integers are either multiples of p or multiples of q (or both). There are q multiple of p between 1 and pq, and similarly p multiples of q. And the integer pq itself is a multiple of both. Thus subtracting these from the total, we get the assertion in the theorem: $\phi(pq) = pq - q - p + 1 = (p-1)(q-1)$.

ii) Left as an exercise. (Use the same argument as above; subtract the number of integers that are not coprime to p from p^k.)
Remark. In fact, more generally, the following is true (which is not that hard to prove):

$$\phi(mn) = \phi(m)\phi(n) \text{ whenever } GCD(m, n) = 1,$$

which, in turn, easily implies the following.

$$\phi(p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}) = p_1^{a_1 - 1}(p_1 - 1) p_2^{a_2 - 1}(p_2 - 1) \cdots p_k^{a_k - 1}(p_k - 1).$$

Next lecture:

i) The GCD and the extended GCD algorithms – how to find an inverse $a^{-1} \pmod{m}$, whenever GCD$(a, m) = 1$. (Recall that a^{-1} is an integer x such that $ax \equiv 1 \pmod{n}$?

ii) Modular exponentiation – given integers a, e, and n, how to compute $a^e \pmod{n}$.

iii) Generalization of Fermat's, known as Euler's theorem, for composite integers N.

iv) Testing primality – given a large integer p, how to test if p is *prime*?