Special Topic: RSA Algorithm

Division of Computer Science and Engineering
HANYANG UNIVERSITY

Fall, 2016

1. Euler's Phi-Function

Definition 1.1. For $n \geq 1$, let $\phi(n)$ denote the number of positive integers not exceeding n that are relatively prime to n. The function ϕ is called the *Euler phi-function* or *totient*.

- $\phi(30) = 8$; the following are the positive integers not exceeding 30 that are relatively prime to 30.

$$1, 7, 11, 13, 17, 19, 23, 29$$

Similarly,

$$\begin{array}{lllll} \phi(1) = 1, & \phi(2) = 1, & \phi(3) = 2, & \phi(4) = 2, & \phi(5) = 4, \\ \phi(6) = 2, & \phi(7) = 6, & \ldots & & \end{array}$$

- For any $n > 1$,

$$\phi(n) = n - 1 \quad \text{if and only if } n \text{ is prime.}$$

Theorem 1.1. If p is a prime and $k > 0$, then

$$\phi(p^k) = p^k - p^{k-1} = p^k(1 - \frac{1}{p}).$$

Proof. gcd(n, p^k)=1 iff $p \nmid n$. There are p^{k-1} integers between 1 and p^k divisible by p, namely,

$$p, 2p, 3p, \ldots, (p^{k-1})p$$

Thus, $\{1, 2, \ldots, p^k\}$ contains exactly $p^k - p^{k-1}$ integers that are relatively prime to p^k.

Lemma. Given integers a, b, c, gcd$(a, bc) = 1$ if and only if gcd$(a, b) = 1$ and gcd$(a, c) = 1$.

Theorem 1.2. The function ϕ is a multiplicative function, i.e., $\phi(mn) = \phi(m)\phi(n)$ wherever $\gcd(m, n) = 1$.

Theorem 1.3. If the integer $n > 1$ has the prime factorization $n = p_1^{k_1} \cdots p_r^{k_r}$, then

$$\begin{aligned}\phi(n) &= (p_1^{k_1} - p_1^{k_1-1})(p_2^{k_2} - p_2^{k_2-1}) \cdots (p_r^{k_r} - p_r^{k_r-1})\\ &= n(1 - \frac{1}{p_1})(1 - \frac{1}{p_2}) \cdots (1 - \frac{1}{p_r}).\end{aligned}$$

Example 1.1. Since $360 = 2^3 \cdot 3^2 \cdot 5$, we have

$$\phi(360) = 360(1 - \frac{1}{2})(1 - \frac{1}{3})(1 - \frac{1}{5}) = 96.$$

Theorem 1.4. For $n > 2$, $\phi(n)$ is an even integer.

Another proof of Euclid's theorem:

Assume that there are only a finite number of primes, we call $p_1, \ldots, p_r$. Consider the integer

$$n = p_1 p_2 \cdots p_r.$$

Then for any $1 < a \leq n$, $\gcd(a, n) \neq 1$. (Why?) Thus, $\phi(n) = 1$, which contradicts to Theorem 1.4.

1.2. Euler's Theorem

Theorem 1.5. (**Euler's Generalization of Fermat's Theorem**)
If $n \geq 1$ and $\gcd(a, n) = 1$, then $a^{\phi(n)} \equiv 1 \pmod{n}$.
(Fermat's Little Theorem): If p is a prime and a any positive integer such that $p \nmid a$, then $a^{p-1} \equiv 1 \pmod{p}$.

Example 1.2. Let us find the last two digits in the decimal representation of 3^{256}. This is equivalent to obtaining the smallest nonnegative integer to which 3^{256} is congruent modulo 100. Because $\gcd(3, 100) = 1$ and

$$\phi(100) = \phi(2^2 \cdot 5^2) = (2^2 - 2)(5^2 - 5) = 40,$$

Euler's theorem yields

$$3^{40} \equiv 1 \pmod{100}.$$

Thus,

$$3^{256} = 3^{6 \cdot 40 + 16} = (3^{40})^6 3^{16} \equiv 3^{16} \equiv 21 \pmod{100}.$$

- If n is an odd integer that is not a multiple of 5, then n divides an integer all of whose digits are equal to 1

Proof. Because $\gcd(n, 10) = 1$ and $\gcd(9, 10) = 1$, we have $\gcd(9n, 10) = 1$. By Euler's theorem,

$$10^{\phi(9n)} \equiv 1 \pmod{9n}.$$

This says that $10^{\phi(9n)} - 1 = 9nk$ for some integer k or, what amounts to the same thing,

$$kn = \frac{10^{\phi(9n)} - 1}{9} = 111 \cdots 111$$

1.3. RSA Algorithm (Rivest, Shamir and Adleman)

Key Generation

1. Let p and q be large prime numbers ($p \neq q$).
2. $N = pq$
3. $\phi(N) = \phi(p)\phi(q) = (p-1)(q-1)$ (By Theorem 1.3)
4. Find e such that $1 < e < \phi(N)$ and $\gcd(e, \phi(N)) = 1$.
5. Find d which is the modular multiplicative inverse of e (mod $\phi(N)$). Then, $de \equiv 1(\text{mod } \phi(N))$. (Recall that if $\gcd(a, m) = 1$ and $m > 1$, then a has a unique (modulo m) inverse a'.)

Now, the public and private keys are $< N, e >$ and $< N, d >$, respectively.

Assume that A wants to send his(her) message m to B.

Encryption (by A)

1. Acquire the public key of B, namely $< N, e >$
2. Encrypt m into c such that $c = m^e \pmod{N}$
3. Send c to B

Decryption (by B)

1. Recieve the cipher text c from A
2. Decrypt c by $m = c^d \pmod{N}$

Proof of correctness (using Euler's Theorem)

We want to show that $c^d \pmod N = m^{ed} \pmod N \equiv m \pmod N$. Since $de \equiv 1 \pmod{\phi(N)}$, there exist a positive integer k which satisfies $de = k\phi(N) + 1$. Then,

$$m^{de} = m^{k\phi(N)+1} = m(m^{\phi(N)})^k$$

By Euler's Theorem, $m^{\phi(N)} \equiv 1 \pmod N$. Hence, $m^{de} \equiv m(1)^k \equiv m \pmod N$.

Security of the RSA Algorithm

If we know $\phi(N)$, others can find the private key $< N, d >$ using public key $< N, e >$. However, if we do not know p and q, it is extremly difficult (actually, almost impossible) to find $\phi(N)$. Also, it is known that the prime factorization of a sufficiently large number is an intractable problem. Hence, if we use sufficiently large primes p and q and keep them securely, others cannot factorize $N(= pq)$ and so cannot figure out $\phi(N)$. Hence, RSA algorithm is secure provided that sufficiently large primes, p and q, are used.