Lecture 14

2.5 More on the Euclidean algorithm. Modular arithmetic & cryptography: RSA.

Using the "extended" Euclidean algorithm, we saw how to find integers m, n such that

$$\gcd(a,b) = ma + nb$$

When a and b are coprime, this is also an effective method to compute modular inverses. Indeed, if $\gcd(a,b) = 1$ then we find m, n such that

$$1 = ma + nb$$

if we look at this identity "modulo a" we get

$$1 \equiv nb \pmod{a}$$

(analogous for b)

which yields: n is (congruent to) the multiplicative inverse of b mod a.

Example Find the multiplicative inverse of 63 in $\mathbb{Z}_{104}$.

→ use the Euclidean algorithm to find the gcd

$104 = 63 \cdot 1 + 41$
$63 = 41 \cdot 1 + 22$
$41 = 22 \cdot 1 + 19$
$22 = 19 \cdot 1 + 3$
$19 = 3 \cdot 6 + 1$ ← is gcd
$3 = 3 \cdot 1 + 0$

$$1 = 19 - 3 \cdot 6 = 19 - (22 - 19) \cdot 6 = 19 \cdot 7 - 22 \cdot 6$$
$$= (41 - 22) \cdot 7 - 22 \cdot 6 = 41 \cdot 7 - 22 \cdot 13$$
$$= 41 \cdot 7 - (63 - 41) \cdot 13 = 41 \cdot 20 - 63 \cdot 13$$
$$= (104 - 63) \cdot 20 - 63 \cdot 13$$
$$= 104 \cdot 20 - 63 \cdot 33$$

that is: $1 = 104 \cdot 20 + 63 \cdot (-33)$ Now take this mod 104:

$1 \equiv 63 \cdot (-33) \pmod{104}$ since $-33 \equiv 71 \pmod{104}$, we get

$$63^{-1} = 71 \text{ in } \mathbb{Z}_{104}.$$

Computing modular inverses is one key step in the widely used method of public key cryptography known as the RSA algorithm.

RSA (Rivest – Shamir – Adleman) algorithm was first described in 1977, and became public in 1997. It is one of the first public-key cryptosystems.

Algorithm description:

- Key generation: – Alice picks two (different, large, random) prime numbers p and q and computes their product $n = p \cdot q$
  – She then chooses a (random) integer e with $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$

→ Her public key is $\boxed{(n, e)}$

- using the extended Euclidean algorithm, she computes the inverse of e mod $\varphi(n)$:

d such that $d \cdot e \equiv 1 \mod \varphi(n)$

$\rightsquigarrow$ Her private key is $\boxed{(n, d)}$

How can Alice & Bob encrypt/decrypt messages?

(we will assume that the message before and after the encryption is in numeric form (elts of $\mathbb{Z}_n$)

Encryption Given a message $m \in \mathbb{Z}_n$, Bob uses Alice's public key (n, e) to compute the ciphertext (the encrypted message)

$$c = m^e \mod n$$

Decryption After receiving c Alice uses her private key (n, d) to decrypt:

$$m = c^d \mod n.$$

Euler's theorem makes sure that this all works! That is:

$$\boxed{m \equiv c^d \equiv m^{d \cdot e} \mod n}$$

Indeed: $n = pq$ so $\varphi(n) = (p-1)(q-1) =: k$.

Turns out that in our situation this is "likely" so we will assume it holds (upon slightly changing our message m

Euler's theorem tells us that if $\gcd(m, n) = 1$ then $m^{\varphi(n)} \equiv 1 \mod n$.

That is $m^k \equiv 1 \mod n$

But $de \equiv 1 \mod k$ which means that there exists r such that $de - 1 = kr$, that is $de = kr + 1$. Therefore

$$m^{de} \equiv m^{kr+1} \equiv m \cdot (m^k)^r \equiv m \cdot \underbrace{(m^{\varphi(n)})^r}_{\equiv 1 \text{ by Euler's thm}} \equiv m \mod n.$$