Some Applications to Cryptography

Suppose we want to send a message to someone, but there is a possibility that a third party can capture our message. Because of that, we want to encode our message before sending it.

<u>Example:</u> (Caesar's cipher)

$A \to B$, $B \to C$, $C \to D$, ..., $X \to Y$, $Y \to Z$, $Z \to A$

encodedmessage $\longrightarrow$ fodpefenfttbhf

To decode, we use

$A \to Z$, $B \to A$, $C \to B$, ..., $Y \to X$, $Z \to Y$

fodpefenfttbhf $\longrightarrow$ encodedmessage

Clearly, this is not a very strong encryption

Goal: Find a way to encode so that it is difficult to decode for third parties.

Let's agree on this: we just need to deal with sending a number to another person instead of an arbitrary message.

Because we can express every letter with a number before encoding it such that $A \to 101$, $B \to 102$, $C \to 103$, ..., $Z \to 126$ etc.

Modular Exponentian Cipher

numbers

Two parties A and B want to exchange messages.

Say x is the message that A wants to send B.

① They choose a prime p (very large, larger than x).
p is a public information, everyone knows

② They agree on a secret key e such that $(e, p-1) = 1$ before starting to exchange messages
e is known by A and B only

③ A will compute $m \equiv x^e \pmod{p}$ and send the encoded message m to B.

④ To decode the received message, B first finds the inverse of e modulo $p-1$ (say f, i.e. $ef \equiv 1 \pmod{p-1}$) and computes $m^f \pmod{p}$ which is equivalent to $x \pmod{p}$ by Fermat's theorem.

$$m^f \equiv (x^e)^f \equiv x^{ef} \equiv x^{k\cdot(p-1)+1}$$

$$\equiv (x^{p-1})^k \cdot x \equiv x \pmod{p}$$

Two important things:

1. B can easily decode because

- finding $f \equiv e^{-1} \pmod{p}$ is easy (Euclidean algoritm)

- computing $m^f \pmod{p}$ is also easy because taking powers mod p is easy.

2. A third party cannot decode easily without knowing e.

Example: (a) Encode $x = 7$ using $e = 26$ and $p = 101$.

$7^{26} \equiv 7^{16} \cdot 7^{8} \cdot 7^{2}$

$7^2 \equiv 49 \pmod{101}$

$7^4 \equiv 49^2 \equiv 2401 \equiv 78 \pmod{101}$

$7^8 \equiv 78^2 \equiv (-23)^2 \equiv 529 \equiv 24 \pmod{101}$

$7^{16} \equiv 24^2 \equiv 576 \equiv 71 \pmod{101}$

$7^{16} \cdot 7^8 \cdot 7^2 \equiv 71 \cdot 24 \cdot 49 \equiv 70 \pmod{101}$

$m = 70$.

(b) Decode $m = 13$ with $e = 7$ and $p = 101$.

Find f such that $7f \equiv 1 \pmod{100}$

$7a + 100b = 1$. Use Euclidean algorithm,

$7 \cdot 43 - 3 \cdot 100 = 1 \Rightarrow f = 43$

Compute $m^f \equiv 13^{43} \equiv 13^{32} \cdot 13^{8} \cdot 13^{2} \cdot 13^{1} \equiv 9 \pmod{101}$

$\Rightarrow x = 9$

<u>Remark:</u> We should choose p in a way that $\mathrm{ord}_p(x)$ is large otherwise there won't be many possibilities for $m^f \equiv (x^e)^f \pmod{p}$.

Imagine $p = 101$ and $x = 10$ and $e = 3$. If a third party captures $m \equiv 1000 \equiv -10 \pmod{p}$, then they know that x is one of the followings:

$(-10)^1 \equiv 91$, $(-10)^2 \equiv 100$, $(-10)^3 \equiv 10$, $(-10)^4 \equiv 1 \pmod{101}$

Only four possibilities: almost as good as decoding it.

Diffie-Hellman Key Exchange

A and B want to agree on a key securely to use later.

① They pick a large prime p and an integer $1 < g < p$.
p and g are public information, everyone knows.

② A chooses a secret integer a and B chooses a secret integer b.
a is only known by A.
b is only known by B.

③ A computes $a' \equiv g^a \pmod{p}$ and sends it to B.
B computes $b' \equiv g^b \pmod{p}$ and sends it to A.
a' and b' are public information, everyone knows.

④ A and B compute $(a')^b \equiv (b')^a \pmod{p}$ using their secret integers, this will be their key.

$$(g^a)^b \equiv (g^b)^a \pmod{p}$$

Even if a third party knows g, p, g^a, g^b, it is still difficult to compute $g^{ab} \pmod{p}$. This is known as Diffie-Hellman problem (D.H.P)
One way to solve D.H.P is to find a and b first.

Discrete Logarithm Problem (D. L. P) : Given g, p and $g^a \pmod p$, can you find a ?

Clearly, DHP is not harder than DLP because if we can solve DLP, then we can solve DHP as well.

Example: (a) Suppose $g=2$, $p=101$

If $g^a \equiv 53$, $g^b \equiv 48 \pmod{101}$, then $g^{ab} \equiv ? \pmod{101}$

Difficult

(b) $g=2$, $p=101$

If $a=23$ and $b=73$, then find the key.

$$2^{23 \cdot 73} = 2^{1679} = (2^{100})^{16} \cdot 2^{79} \equiv 2^{79} \pmod{101}$$

$$2^{79} \equiv 2^{64} \cdot 2^{8} \cdot 2^{4} \cdot 2^{2} \cdot 2^{1} \equiv 42 \pmod{101}$$

RSA Public Key

In RSA public key, our goal is to come up with a system that allows a random person to send us a message securely.

① We pick two very large distinct primes p and q and an encryption key e such that $(e, (p-1)\cdot(q-1)) = 1$

We publish e and the value of pq, say on our website. pq and e are public information, everyone knows. The pair (pq, e) is called the public key

The values of p and q are only known to us. (Even if pq is publicly known, factorizing to find p and q is not easy. Practically not possible when p and q are large enough)

② Say someone wants to send us the message x securely. They will compute $m \equiv x^e \pmod{pq}$ and send the encoded message m to us.

③ To decode the received message, we first find the inverse of e modulo $(p-1)\cdot(q-1)$ (say f, i.e. $ef \equiv 1 \pmod{(p-1)(q-1)}$) and compute $m^f \pmod{pq}$ which is equivalent to $x \pmod{pq}$ by Euler's theorem, $\phi(pq) = (p-1)\cdot(q-1)$

(f will be called our private key) $m^f \equiv x^{ef} \equiv x^{(p-1)\cdot(q-1)\cdot k+1}$

$\equiv x \pmod{pq}$

Remark: A third party cannot decode m because they cannot compute f without knowing $(p-1)\cdot(q-1)$.

Example: (a) Suppose $pq = 10147$, $e = 119$ and $m = 9247$, decode to find x.

This is not very easy, even factorizing 10147 is not so easy.

(b) $p = 73$, $q = 139$, $e = 119$, $m = 9247$, decode to find x.

Find f such that $119f \equiv 1 \pmod{72 \cdot 138}$

$\Rightarrow f = 167$ after Euclidean algorithm

$$(9247)^{167} \equiv 9247^{128} \cdot 9247^{32} \cdot 9247^{4} \cdot 9247^{2} \cdot 9247$$

$$\equiv 3 \pmod{10147}.$$