ALGEBRA NOTE : 3

JOHNEW ZHANG

1. Cryptography

Problem 1. *Send a message from person A(Alice), B(Bob), in such a way that no-one else can read the message but the message, but the message cannot be read by anyone else if intercepted.*

Once-time pad : Shift each character by some amount, given by the pad, and that is the encrypted message.

Application 1. *Deffie-Hellman Key Exchange (the way to generate common secret :*

Use a large prime p and some $g \pmod p$.
Alice chooses a and publish $g^a \pmod p$
Bob choose b and publish $g^b \pmod p$.
Both know $g^{ab} \pmod p$. ■

For someone else to find out what $g^{ab} \pmod p$ is, they need to solve $g^a \pmod p$.

To solve this compute $g^k \pmod p$ for $0 \leq k < p-1$.

We hope that the computations Alice and Bob need to do are a lot faster than the one Eve needs to do.

Successive Squaring : Very fast way to compute $g^a \pmod p$.

The Algorithm to compute $g^a \pmod p$:

(1) Write a as a sum of distinct powers of 2:

$$a = b_0 + 2b_1 + 2^k b_k$$
$$b_1 = 0, or, 1$$

(2) To compute:

Date: Oct. 18.

$$\begin{aligned} g^2 &\pmod p \\ (g^2)^2 \equiv g^2 &\pmod p \\ &\cdots \\ (g^2)^k &\pmod p \end{aligned}$$

(3) $g^a \equiv g^{b_0+2\cdot b_1++b_k\cdot 2^k} \equiv g^{b_0\ldots(b^{2^k})^{b_k}} \pmod p$

Example : Compute $5^{10275} \pmod{22447}$
$10275 = 2^{13} + 2^{11} + 2^5 + 2^1 + 2^0$
$5^2 \equiv 25 \pmod{22447}$
$5^4 \equiv 25^2 \equiv 625 \pmod{22447}$
...
$5^{2^{13}} \equiv 10583 \pmod{22447}$
$5^{10275} \equiv 5^{2^{13}+2^{11}+2^5+2^1+2^0} \equiv (10583)(18470)(12253)(25)(5) \equiv 10009 \pmod{22447}$

Suppose you have a function (on a computer) multiply-mod-$p(x, y)$, which takes a fixed amount of time (depending on p).

Calculating $g^a \pmod p$ by repeated multiplication takes about a uses of this function.

For successive squaring, we need to square k times, where 2^k is largest power of two less than (or equal to) a.

$2^k \le a \implies k \le \log_2 a = \frac{\log a}{\log 2}$.

To construct $g^a \pmod p$ from this, we have to do at most k more multiplications. Total number of times calling "multiplication" is $\le 2\log_2 a$.

The graph is on the notes.

For Alice and Bob set up the key takes about $4\log_2 a$ multiplications (if a is roughly equal to b). For Eve to break the key, about a multiplications.

Increasing p has about the same effect,

Alice and Bob takes roughly $4\log_2 a$ (a is the largest number evolved).

Eve takes the largest number evolved.

Assume that multiplication takes about 10^{-3} s.

a	$Alice + Bob$	Eve
1000	$0.053s$	$1s$
10^6	$0.079s$	$17min$
10^8	$0.106s$	$28hrs$
10^{20}	$0.2657s$	$3,170,000,000years$

Conclusion for Encryption : The Diffie-Hellmon key exchange generates the common secret. Using successive squaring, Alice and Bob can generate a key vey quickly. Eve takes a long time to figure out the key.

Reasonable for communication between two equal parties, Alice and Bob less reasonable for things like e-commerce.

It would be nice if Alice, say, could set things up just once.

Alice should be able to post a "public key" which people can use to send to her messages.

Application 2. *RSA (Rivest-Shamir-Adelman) :*

(1) Alice choose 2 large primes, p and q, and compute

$m = pq$

$\varphi(m) = (p-1)(q-1)$

She then chooses $1 \leq e \leq \varphi(m)$ *with* $gcd(e, p-1) = 1$. *(helpfully not* $e = 1$, *or* $e = \varphi(m) - 1$ *).*

Then compute d such that $ed \equiv 1 \pmod{\varphi(m)}$, *(Bezout, and Euclidean Algorithm)*

Public key : m *and* e

Private key : $\varphi(m)$ *and d (forget p and q).*

If Bob wants to send a message, a, (first check that $gcd(a, m) = 1$*), Bob needs his message to satisfy* $1 \leq a < m$.

Bob computes $a^e \pmod{m}$ *(by successive squaring), and sends that.*

Public : $a^e \pmod{m}$

Alice gets $a^e \pmod{m}$, *SHe computes* $(a^e)^d \equiv a^{ed} \equiv a \pmod{m}$ *by Euler's Theorem, since* $ed \equiv 1 \pmod{\varphi(m)}$.

How can Eve, using m, e, and $a^e \pmod{m}$ *to find* $a \pmod{m}$*?*

Eve knows she needs to solve $ex \equiv 1 \pmod{\varphi(m)}$.

She needs to know $\varphi(m)$.
If you factor m, you can write down $\varphi(m)$.

How do you factor m?
Just find a prime $p|m$.
In fact, if $m = pq$, it turns out that computing $\varphi(m)$ is just as hard as as factoring. Suppose we know m and $\varphi(m)$.

Why?

The $pq = m$, and $(p-1)(q-1) = \varphi(m)$.
$(p-1)(\frac{m}{p} - 1) = \varphi(m)$.
then $(p-1)(m-p) = \varphi(m)p$.
$pm - p^2 - m + p = \varphi(m)p$,
or $p^2 + (\varphi(m) - m - 1)p + m = 0$, solve for p.
We suspect that it is hard (not polynomial time) to factor integers.

$\frac{m}{p}$ possible, a which are divisible by p and $\frac{m}{q}$ which are divisible b yq, so $m - \frac{m}{p} - \frac{m}{q} = m(1 - \frac{1}{p} - \frac{1}{q}$ and ok (have no common factor with m). The proportion of message which will work is $\geq 1 - \frac{1}{p} - \frac{1}{q}$.

Example : Alice chooses $p = 31, q = 37$, $\varphi(m) = 30 \cdot 36 = 1080$.
The public key is $m = 1147, e = 419$.
$d = 299$ because $299 \cdot 419 \equiv 1 \pmod{1080}$.
Bob wants to send a message "917".
Bob computs $917^{419} \equiv 763 \pmod{1147}$
$763 \pmod{1147}$
Then Alice gets this and computes $763^{299} \equiv 917 \pmod{1147}$.

$\star$ Creating the key and encrypting/decrypting use
(1) successive squaring,
(2) Euclidean Algorithm.

Successive squaring is fast (polynomial time). The time it takes is roughly proportional to the number of bits or digits of the numbers involved (linear time).
The Euclidean Algorithm is also polynomial time.
Breaking the key requires factoring which is slow.

How do you factor?

Application 3. *Pollard $p-1$ "Algorithm :*
Idea : want to factor $m = pq$ (or anything).
Pick $1 < a < m$, if $gcd(a, m) \neq 1$, we are done.
If $(p-1)|k$, then $a^k \equiv 1 \pmod{p}$.
so $p|a^k - 1$, if we compute $b \equiv a^k - 1 \pmod{m}$.
$p|gcd(a^k - 1, m) = gcd(b, m)$.
In general, it is possible that $a^k \equiv 1 \pmod{p}$ for some k smaller than $p-1$,
For example, $2^3 \equiv 1 \pmod{7}$.
We choose a "likely candidate" k, compute $b \equiv a^k - 1 \pmod{m}$,and $gcd(a^k - 1, m) = gcd(b, m)$.
This is a divisor of m if it is 1 or m, this tells us nothing, but maybe it isn't.

What kind of k that we should choose?
This works best if k has a lot of small prime factors.
$k = lcm(2, 3, 4, 5 \ldots)$.

Example : $m = 143, a = 2, k = lcm(2, 3, 4) = 12$
Calculate $b \equiv 2^{12} - 1 \pmod{143}$.
$2^{12} \equiv 92 \pmod{143}$
so $b \equiv 92 \pmod{143}$
$gcd(2^{12} - 1, 143) = gcd(91, 143) = 13$.
So $143 = 13 \cdot 11$.

Example : $m = 391, a = 2, k = lcm(2, 3, 4) = 12$.
Compute
$2^{12} - 1 \pmod{143}$
$2^{12} \equiv 185 \pmod{391}$
$gcd(185, 391) = 1$ (fail)
Change $k = lcm(2, 3, 4, 5, 6, 7) = 420$.
Compute
$2^{420} - 1 \pmod{143}$
$2^{420} \equiv 49 \pmod{391}$
$gcd(49, 391) = 1$ (fail)
Change $k = lcm(2, \ldots 8) = 840$.
so $2^{420} \equiv 153 \pmod{391}$
$gcd(153, 391) = 17$