Computational Tools

AND

KNAPSACK CIPHERS

Computational Tools

I We are interested in computing

$$a^m \bmod n$$

where a m and n are given positive integers. Always

$$m = x_0 2^0 + x_1 2^1 + x_2 2^2 + \cdots + x_t 2^t$$

where x_i is 0 or 1 and $x_t = 1$

Ex $21 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4$

Then $a^m = a^{x_0 2^0 + \cdots + x_t 2^t} =$

$$a^{x_0 2^0} a^{x_1 2^1} \cdots a^{x_t 2^t}$$

We can compute the exponents efficiently as follows:

$$a^{2^j} = a^{2 \cdot 2^{j-1}} = (a^{2^{j-1}})^2$$

In short, we compute a list of a^{2^j} by squaring each element, one at a time

Ex $3^{218} \bmod 1000$

$$218 = 128 + 64 + 16 + 8 + 2$$
$$= 2^7 + 2^6 + 2^4 + 2^3 + 2$$

$$3^{218} = 3^{2^7} 3^{2^6} 3^{2^4} 3^{2^3} 3^2 \bmod 1000$$

The list is

$3^{2^1} = 3^2 = 9$

$3^{2^2} = 9^2 = 81$

$3^{2^3} = 81^2 = 561$

$3^{2^4} = 561^2 = 721 \mod 1000$

$3^{2^5} = 721^2 = 841 \mod 1000$

$3^{2^6} = 841^2 = 281 \mod 1000$

$3^{2^7} = 281^2 = 961 \mod 1000$

$3^{218} = 3^{2^7} \cdot 3^{2^6} \cdot 3^{2^4} \cdot 3^{2^3} \cdot 3^2 \mod 1000$

$= 961 \cdot 281 \cdot 721 \cdot 561 \cdot 9 =$

$489 \mod 1000$

To compute $a^m \mod n$ where t is the highest power of 2 that appears in the expansion of m, it takes $t \le \log_2 m$ steps to compute the table and at most t steps to multiply the answers in the table, so at most $t \log_2 m$ multiplications

This is a big savings over rasing a to powers m times.

II Euclidean Algorithm

To compute inverses mod n, the Euclidean algorithm is used. which goes as follows.

First we compute the g.c.d of a and b: Suppose $a > b$

$$a = b\,q_1 + r_1 \qquad 0 \le r_1 < b$$
$$b = r_1\,q_2 + r_2 \qquad 0 \le r_2 < r_1$$
$$r_1 = r_2\,q_3 + r_3 \qquad 0 \le r_3 < r_2$$
$$\vdots$$
$$r_{n-2} = r_{n-1}\,q_n + r_n \qquad 0 \le r_n < r_{n-1}$$
$$r_{n-1} = r_n\,q_{n+1}$$

Since $r_1, \ldots r_n$ are decreasing positive integers, this sequence terminates. r_n is the gcd

This can be seen by working with the equations

Ex $a = 81 \qquad b = 64$

$$81 = 64 \cdot 1 + 17$$
$$64 = 17 \cdot 3 + 13$$
$$17 = 13 \cdot 1 + 4 \rightarrow \gcd(81, 64) = 1$$
$$13 = 4 \cdot 3 + 1$$
$$4 = 1 \cdot 4$$

64 has an inverse mod 81 if and only if $\gcd(64, 81) = 1$ which it is. So we look for the inverse using the Euclidean algorithm Table

Row	R	Q	U	V
-1	a	–	1	0
0	b	–	0	1
1	r_1	q_1	u_1	v_1
2	r_2	q_2	u_2	v_2
$\vdots$				
n	r_n	q_n	u_n	v_n

The R and Q columns come from the list that finds the gcd. $u_{-1} = 1$ $u_0 = 0$ $v_{-1} = 0$ $v_0 = 1$ are initial conditions and are always the same. Note that

$$r_{i+1} = r_{i-1} - r_i q_{i+1}$$

Then $$u_{i+1} = u_{i-1} - u_i q_{i+1}$$

$$v_{i+1} = v_{i-1} - v_i q_{i+1}$$

is the way u_{i+1} and v_{i+1} are defined and are easy to remember since the follow r_{i+1}.

The reason for all this is, for each row

$$* \quad r_j = a u_j + b v_j$$

So the last row has

$r_n = a u_n + b v_n$. If $r_n = 1$, then

$1 = a u_n + b v_n$ and

$b v_n \equiv 1 \mod a$.

Hence v_n is $b^{-1} \mod a$

To show $*$ use induction

The first and second rows are seen by substitution. Assume up to $j-1$ that the result holds

Then $r_j = r_{j-2} - r_{j-1} q_j$

$= (a u_{j-2} + b v_{j-2}) - (a u_{j-1} + b v_{j-1}) q_j$

$= a(u_{j-2} - u_{j-1} q_j) + b(v_{j-2} - v_{j-1} q_j)$

$= a u_j + b v_j$.

Ex. We continue the example

$a = 81 \quad b = 64$

Row	R	Q	U	V
-1	81	-	1	0
0	64	-	0	1
1	17	1	1	-1
2	13	3	-3	4
3	4	1	4	-5
4	1	3	-15	19

$\rightarrow \quad 1 = 81(-15) + 64(19)$

$\rightarrow \quad 64 \cdot 19 \equiv 1 \bmod 81$

19 is the inverse of 64 mod 81

It is possible that the final v_n is negative, giving a negative inverse. Simply use the congruence relation (add enough multiples of a) to v_n to get a positive inverse

III A result of Fermat

Thm. If p is a prime and p does not divide e, then

$$e^{p-1} \equiv 1 \bmod p$$

Proof. Let $S = \{1, \ldots, p-1\}$

$eS = \{e, 2e, \ldots (p-1)e\} \bmod p$

If $ei \equiv ej \bmod p$ then p divides $e(i-j)$

$\rightarrow$ p divides $i-j$

$\rightarrow$ $i \equiv j \bmod p$

So $eS \bmod p$ just rearranges the elements in S. So the product of all the elements in S equals the product of all the elements in $eS \bmod p$

$\rightarrow$ $(p-1)! = e^{p-1}(p-1)! \bmod p$

$\rightarrow$ p divides $(e^{p-1}-1)(p-1)!$ ~~mod~~

$\rightarrow$ p divides $e^{p-1}-1$

$\rightarrow$ $e^{p-1} \equiv 1 \bmod p$

Thm. Suppose $\gcd(e, p-1) = 1$ Then there is a d such that $ed \equiv 1 \bmod (p-1)$

Then $a^{ed} \equiv a^{1+k(p-1)} = a \bmod p$

Proof When $\gcd(a, p) = 1$,

$$a^{ed} = a^{1+k(p-1)} = a(a^{p-1})^k = a \cdot 1 = a \bmod p$$

If $\gcd(a, p) \neq 1$, then $\gcd(a, p) = p$ and p divides a

So $a^{ed} \bmod p = 0 \bmod p = a \bmod p$

These facts allow the existence of a cipher, the exponential cipher, a predecessor of the RSA. It goes as follows. Alice and Bob pick a prime p and e with $\gcd(e, p) = 1$ They compute d such that $ed \equiv 1 \bmod p$ Bob wants to send a message a to Alice. Bob computes $a^e \bmod p = b$ and sends b to Alice who computes $b^d = a^{ed} \bmod p = a$ and gets the message. This is a private key system. For if Eve knows e and p, she then knows $p-1$ and can find d from

the Euclidean algorithm for $a = p-1$ and $b = e$.

Ex Bob ~~wants~~ and Alice pick $p = 181$, $e = 97$. They compute $d = 28$ from the Euclidean algorithm. ie $ed \equiv 1 \bmod 180$

Bob is to send

B E T R A Y A L

to Alice. This converts to

1 4 19 17 0 24 0 11

Each of these is a in $a^{97} \bmod 181$. ie $19^{97} \bmod 181$ for T

Computing each gives

1 94 19 92 0 158 0 118

which is send to Alice who computes $b^d \bmod 181$ $(d = 28)$

This will give the original numbers which she converts to English. There is a lot of computation and a computer would be used. But it uses the repeated squaring algorithm discussed in the beginning of the computation section

KNAPSACK CIPHERS

KNAPSACK CIPHERS

KNAPSACK CIPHERS ARE OUR FIRST EXAMPLE OF PUBLIC KEY CRYPTOGRAPHY. They were the rage until the late 80's when they were broken. Some concepts are needed.

Given integers $a_1, \ldots, a_n$ choose some of them and take the sum of them. Can we find which ones were choosen? This can be hard but for special sequences, it can be solved by an algorithm.

Let $a_1, \ldots, a_n$ be positive integers such that $a_1 + \cdots + a_{i-1} < a_i$ for each $i > 1$. Such a sequence is called SUPER INCREASING

Ex 1, 3, 5, 12, 24 is superincreasing. Suppose $S = 30$ is the sum of some of them. Which ones?

Algorithm

Test	$24 \leq 30$	Yes keep 24 and compute $30-24=6$
Test	$11 \nleq 6$	No move on
Test	$5 \nleq 6$	Yes. Keep 5 and compute $6-5=1$
Test	$3 \nleq 1$	No move on
Test	$1 \leq 1$	Yes keep 1 and stop

Then $1+5+24=30$

KNAPSACK CIPHERS

Alice picks integers $a_1, \ldots, a_n$ which is a superincreasing sequence

Alice pick a prime $p > a_1 + \cdots + a_n$
Find e such that $\gcd(e,p)=1$
Since p is a prime, any $e<p$ works.

Use the Euclidean algorithm table to find d such that $ed \equiv 1 \bmod p$

Alice ~~picks some of the a_i's~~ ~~and~~ computes

$$b_i = e a_i \bmod p$$

She makes public

$b_1, \ldots b_n$ and p

Bob wants to send a message written in binary (all 0's and 1's) to Alice If is $e_1, \ldots, e_n$ is the message. ie the message is a binary string. Let $e_1, \ldots, e_n$ be the message. Bob computes $\sum e_i b_i = c \bmod p$ Note that computing c is just adding certain of the b_i together.

Bob sends c to Alice Alice finds d using the Euclidean algorithm on p and e Alice needs to find the value of each e_i as she is looking for the a_i that add to dc. For that she uses the super increasing sequence algorithm

4 G

Alice picks 1, 3, 5, 11, 24 and $p = 53$. She picks $e = 20$ and finds $d = 8$ from the Euclidean algorithm. for e and p Alice computes $h_i = e_i a_i \bmod p$ and this is her public key

20, 7, 47, 8, 3

(Thus $20 = 20 \cdot 1 \bmod p$

$7 = 20 \cdot 3 \bmod p$

$47 = 20 \cdot 5 \bmod p$

$8 = 20 \cdot 11 \bmod p$

$3 = 20 \cdot 24 \bmod p$

She makes 20, 7, 47, 8, 3 and $p = 53$ public.

Bob wants to send 11001

He computes $1 \cdot 20 + 1 \cdot 7 + 0 \cdot 47 + 0 \cdot 8 + 1 \cdot 3 = 30 \bmod 53$

He sends 30 to Alice. 30 is the cipherText.

Alice solves $30 \cdot d \bmod 53 = 28$ where d is the inverse of $e \bmod p$. using the Euclidean algorithm.

5

$p = 53 \qquad e = 20$

$53 = 20 \cdot 2 + 13$

$20 = 13 \cdot 1 + 7$

$13 = 7 \cdot 1 + 6$

$7 = 6 \cdot 1 + 1$

row	R	Q	u	v
-1	53	–	1	0
0	20	–	0	1
1	13	2	1	-2
2	7	1	-1	3
3	6	1	2	-5
4	1	1	-3	8

$d = 8$

Check $1 = 53(-3) + 20(+8) = 1$

6

Alice runs the super increasing sequence algorithm on 1, 3, 5, 11, 24 and 28

$24 \leq 28$	Keep 24	$28-24=4$
$11 \leq 4$	No, more on	
$5 \leq 4$	No, more on	
$3 \leq 4$	Yes keep 3,	$4-3=1$
$1 \leq 1$	Yes keep 1	

$24 + 3 + 1 = 28.$