Cryptography using Primes

Max Neunhöffer

University of St Andrews

20 June 2008

Computing with remainders

For $a, n \in \mathbb{Z}$ we can divide a with remainder by n:

$$a = q \cdot n + r \;, \qquad \text{with } q, r \in \mathbb{Z},$$

such that $0 \leq r < n$.

$\Longrightarrow$ q and r are uniquely determined.

We write

$$a \equiv b \pmod{n}$$

if a and b have the same remainder on division by n.

We say: "a is equal to b modulo n".

Same as: the difference $a - b$ is divisible by n.

Note in particular: $a \equiv r \pmod{n}$.

Powering

Note (Computation tricks)

If $a \equiv \hat{a} \pmod{n}$ and $b \equiv \hat{b} \pmod{n}$, then

$$a + b \equiv \hat{a} + \hat{b} \pmod{n}$$

and

$$a \cdot b \equiv \hat{a} \cdot \hat{b} \pmod{n}$$

and thus

$$a^k \equiv \hat{a}^k \pmod{n}.$$

Can we compute 123^{129} modulo 10 easily?

$$\begin{aligned}
123^{129} &\equiv 3^{129} \equiv 3^{1+128} \equiv 3 \cdot 3^{(2^7)} \pmod{10} \\
&\equiv 3 \cdot (((((((3^2)^2)^2)^2)^2)^2)^2 \pmod{10} \\
&\equiv 3 \cdot ((((((9^2)^2)^2)^2)^2)^2 \pmod{10} \\
&\equiv 3 \cdot ((((1^2)^2)^2)^2)^2 \equiv 3 \pmod{10}
\end{aligned}$$

Fermat and Euclid

Theorem (Little Theorem of Fermat)

Let $n = p \cdot q$ be the product of two primes p and q. Then

$$a^{(p-1)(q-1)} \equiv 1 \pmod{n}$$

for all integers a that are not divisible by p or q.

From this we get immediately:

For $k \equiv 1 \pmod{(p-1)(q-1)}$ we have

$$a^k \equiv a \pmod{n},$$

as $a^k \equiv a^{x\cdot(p-1)(q-1)+1} \equiv (a^{(p-1)(q-1)})^x \cdot a \equiv a \pmod{n}$.

Theorem (Euclidean Algorithm)

If $d, m \in \mathbb{Z}$ do not have a common prime divisor, then it is (efficiently) possible to determine an $e \in \mathbb{Z}$, such that $de \equiv 1 \pmod{m}$.

Preparations

The RSA (Rivest-Shamir-Adleman) cryptosystem:

If Alice wants to send a secret message to Bob:

- Bob chooses two primes p and q
- and computes $n = p \cdot q$ and $m = (p-1)(q-1)$.
- He then chooses d such that m and d do not have a common prime divisor
- and computes an e, such that $de \equiv 1 \pmod{m}$.
- He then publishes n and e
- and keeps secret p, q, m and d.

Encrypting and Decrypting

Public: n and e Secret: p, q, $m = (p-1)(q-1)$ and d

Alice can now encrypt a message:

- Encode the message as numbers a with $1 < a < n$.
- Compute encrypted message c by

$$c \equiv a^e \pmod{n} \qquad \text{with } 1 \leq c < n$$

- Send c to Bob.

Bob can then decrypt the message:

- Receiving c, he computes b by

$$b \equiv c^d \pmod{n} \qquad \text{with } 1 \leq b < n$$

- He gets back $b \equiv c^d \equiv (a^e)^d \equiv a^{de} \equiv a \pmod{n}$ since $de \equiv 1 \pmod{(p-1)(q-1)}$.

Factorisation

Public: n and e Secret: p, q, $m = (p-1)(q-1)$ and d

If one knows p and q, one can compute m and d.

If one knows $m = (p-1)(q-1)$, then also p and q.

Proof: $p + q = n + 1 - (pq - p - q + 1)$ and
$(X - p)(X - q) = X^2 - (p + q)X + pq$

Knowing n in principle determines p and q!

However, actually computing p and q from n is HARD.

Computing roots and discrete logarithm

Public: n and e Secret: p, q, $m = (p-1)(q-1)$ and d

Cracking the encryption is basically solving the equation

$$x^e \equiv c \pmod{n}$$

that is, computing e-th roots.

However, computing e-th roots is HARD.

- Assume $1 < z < n$ such that every a is a power of z modulo n (not always possible!).
- Compute $w \equiv z^e \pmod{n}$ with $1 < w < n$.
- Solve $c \equiv w^x$ ("discrete logarithm").
- Then $a \equiv z^x \pmod{n}$ since

$$a \equiv c^d \equiv w^{dx} \equiv (z^e)^{dx} \equiv (z^{ed})^x \equiv z^x \pmod{n}$$

However, solving discrete logarithms like $c = z^x$ is HARD.

Not proved!

There is

no efficient method known

for

integer factorisation

or

computing e-th roors

or

discrete logarithms!

However: It is also not proved, that there is none!.