Some number theoretical background for RSA-cryptography and Shor's algorithm

need: **modular arithmetics**

$\hat{=}$ arithmetics in $\mathbb{Z}_N = (0, 1, \ldots, N-1)$

for $a, b \in \mathbb{Z}$:

$$\boxed{\begin{array}{l} a = b \quad (\mathrm{mod}\ N) \\ \text{iff} \quad a = b + k \cdot N \quad \text{for} \\ \text{some} \quad k \in \mathbb{Z} \end{array}}$$

e.g.:

$$5 = 0 \quad (\mathrm{mod}\ 5)$$

$$7 = 2 \quad (\mathrm{mod}\ 5)$$

$$-8 = 2 \quad (\mathrm{mod}\ 5)$$

$$143 = 53 \quad (\mathrm{mod}\ 5) \quad \text{etc.}$$

sometimes (particularly in comp. science)

mod N as operation:

$a \bmod N := b \in \mathbb{Z}_N$ s.t. $a = b + kN$

(= remainder of N divided by a)

$\rightarrow$ „$+$", „$-$", „$\cdot$" in $\mathbb{Z}_N$ as usual, up to "mod N"!

note:

- $a + b = a \bmod N + b \bmod N \pmod N$
- $a \cdot b = (a \bmod N) \cdot (b \bmod N) \pmod N$

Modular Exponentiation

efficiently "by squaring":

$$b^a = b^{\sum_{\ell=0}^{n-1} a_\ell 2^\ell} = \prod_{\ell=0}^{n} \left(b^{(2^\ell)}\right)^{a_\ell}$$

$\{0,1\}$

$b^2, b^4, b^8, b^{16} \dots \pmod N$ can be determined recursively:

$$b \xrightarrow[\cdot b \bmod N]{} b^2 \xrightarrow[\cdot b^2 \bmod N]{} b^4 \xrightarrow[b^4 \bmod N]{} b^8 \text{ etc.}$$

$\rightsquigarrow$ $\boxed{b^a \bmod N \text{ can be efficiently computed in poly}(\log N) \text{ time !}}$

Modular division in $\mathbb{Z}_N$:

Def.:

a and b co-prime $:\Leftrightarrow$ a and b don't have common divisors

$\Leftrightarrow \gcd(a, b) = 1$

greatest common divisor of a and b

Thm. (Bezout) :

$$\gcd(a,b) \stackrel{!}{=} \min\{ k a + l b > 0 ,\; k, l \in \mathbb{Z}\}$$

⌜proof e.g. in M. Schroeder, Number Theory in Science and Communication (Springer)⌟

For a and b co-prime this means:

$$1 \stackrel{!}{=} \gcd(a,b) = k a + l b$$

$\rightarrow \exists\, k : \quad k a = 1 \pmod{b}$

$\quad \llcorner = a^{-1}$!

$\rightarrow$ Thm.:

inverse a^{-1} of $a \bmod N$ exists iff a and N are co-prime.

$\rightarrow$ efficient computation of $a^{-1} \bmod N$ with extended Euclid's algorithm:

Euclid's algorithm repeatedly makes use of

$$\gcd(a,b) \overset{!}{=} \gcd(b, a \bmod b),$$

as in the following example for $\gcd(71,31)$:

$$71 = 2 \cdot 31 + 9 \qquad (1)$$

$$31 = 3 \cdot 9 + 4 \qquad (2)$$

$$9 = 2 \cdot 4 + 1 \qquad (3)$$

$$4 = 4 \cdot 1 + \underline{\underline{0}} \qquad (4)$$

$\rightarrow$ $\gcd(71,31) = 1$ and

$$1 \overset{(3)}{=} 9 - 2 \cdot 4$$

$$\overset{(2)}{=} 9 - 2(31 - 3 \cdot 9) = 7 \cdot 9 - 2 \cdot 31$$

$$\overset{(1)}{=} 7(71 - 2 \cdot 31) - 2 \cdot 31$$

i.e.

$$\underline{\underline{1}} = \overset{!}{\underline{\underline{-16}}} \cdot \underline{\underline{31}} + 7 \cdot \underline{\underline{71}}$$

hence $31^{-1} \overset{!}{=} -16 = 55 \pmod{71}$

Euler's φ-function

$\varphi(n) :=$ number of elements in

$\{1, 2, 3, \ldots, n-1, n\}$

that are co-prime to n

e.g.:

- $\varphi(1) = 1$
- $\varphi(10) = \#\{1, \cancel{2}, 3, \cancel{4}, \cancel{5}, \cancel{6}, 7, \cancel{8}, 9, \cancel{10}\} = \underline{\underline{4}}$

  $\#$ „number of elements in"
- $\varphi(11) = 10$

for p prime: $\varphi(p) = p-1$

(since all $k<p$ are co-prime to prime p!)

Multiplication-theorem:

For a co-prime b: $\varphi(a \cdot b) \overset{!}{=} \varphi(a)\,\varphi(b)$

$\rightarrow$ p, q prime $\leadsto$ $\varphi(p \cdot q) = \varphi(p)\,\varphi(q)$

$$= (p-1)(q-1)$$

⌜ e.g.: $\varphi(15) = \varphi(3 \cdot 5) = 2 \cdot 4 = 8$ ⌟

Euler's theorem:

For a and n co-prime:

$$a^{\varphi(n)} = 1 \pmod{n}$$

(for a proof see e.g. Schroeder's book)

for the special case that n is prime

also known as

Fermat's Little Thm.

For p prime: $a^{p-1} = 1 \pmod{p}$

Examples:

- $2^6 = 1 \pmod 7$ ✓

  $(64 = 1 + 9 \cdot 7 \quad :-)$

- 7 divides $999\,999$!

Euler/Fermat: $10^{7-1} \overset{!}{=} 1 \pmod 7$

$\Longleftrightarrow \quad 10^6 - 1 = 0 \pmod 7$

Euler's theorem is basis for

RSA - public-key-encryption

Adleman
Shamir
Rivest (1977)

Alice $\rightleftarrows$ (public channels) Bob: $M \in \mathbb{Z}_N$

How can Bob transfer message M <u>secretly</u> to Alice via <u>public</u> channels?
(= www)

RSA solve the problem as follows:

Alice:

- draws randomly two primes p and q (of about 100 digits each)
- computes $n = p \cdot q$

  and $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1)$
- draws e co-prime to $\varphi(n)$
- publishes "public key" (e, n)

Bob:

- encodes message M with (e, n) to

$$E(M) = M^e \mod n$$

and sends $E(M)$ to Alice

Alice

- recieves E from Bob and decodes with

$$D(E) := E^{d} \bmod n$$

where $d = e^{-1} \bmod \varphi(n)$

Note: Lacking prime factors p and q of n, a potential eavesdropper Eve can't compute $\varphi(n)$ as $\varphi(n = pq) = \varphi(p) \cdot \varphi(q) = (p-1)(q-1)$ and thus also can't compute d in order to decode E!

Why does Alice's decoding work?

- d as inverse of $e \mod \varphi(n)$ satisfies
  $de = 1 + k\varphi(n)$;
- assume that M and n are co-prime
  (e.g. when $M < p$ and $M < q$)

$$\rightarrow \quad \underline{E(M)^d} = (M^e)^d = M^{1+k\varphi(n)}$$
$$= M \Big(\underbrace{M^{\varphi(n)}}_{\substack{\shortparallel \\ 1 \bmod n}}\Big)^k = \underline{M} \pmod n$$

Euler $\longrightarrow$ ✓

Remk.:

RSA - encryption is considered to be safe as much as factoring n into p and q is seen to be a hard problem ! (as it seems to be the case for classical computers)

Since Shor's algorithm running on a quantum computer easily finds factors p and q of n, quantum computing poses a real threat to private communication via www!

Actually, Shor's algorithm primarily finds the order r of $b \bmod N$;

$= $ least integer b s.t.

$b^r = 1 \bmod N$

this enables efficient integer factorization, but also immediate $\longrightarrow$

RSA - decryption by eavesdropper

Eve:

- determines order r of encrypted message E mod n by use of Shor's algorithm
- computes $\tilde{d}$ as inverse of e mod r $(\rightsquigarrow \tilde{d} e = 1 + kr)$
- decodes by
$$\tilde{D}(E) := E^{\tilde{d}} \bmod n$$

this works because of the following

Fact: If e and $\varphi(n)$ co-prime,(*) then order r of M^e mod n also order of M mod n!

(*) true for e and $\varphi(n)$ of RSA-key (e,n)!

then :

$$\tilde{D}(E(m)) = \tilde{D}(m^e) = (m^e)^{\tilde{d}}$$

$$= m^{\tilde{d}e} = m^{1+hr}$$

$$= m \underbrace{(m^r)^k}_{\substack{\shortparallel \\ 1 \pmod{n}}} = m \pmod{n}$$

✓