Exponentiation

Tanja Lange

Eindhoven University of Technology

2WF80: Introduction to Cryptology

How to compute modular exponentiation

Want to compute

$$c \equiv a^b \bmod n$$

for given $a \in \mathbb{Z}, b, n \in \mathbb{N}$.

How to compute modular exponentiation

Want to compute

$$c \equiv a^b \bmod n$$

for given $a \in \mathbb{Z}, b, n \in \mathbb{N}$.
By definition

$$c \equiv a^b = \underbrace{a \cdot a \cdot a \cdots a}_{b \text{ copies of } a} \bmod n$$

How not to compute modular exponentiation

Want to compute

$$c \equiv a^b \bmod n$$

for given $a \in \mathbb{Z}, b, n \in \mathbb{N}$.
By definition

$$c \equiv a^b = \underbrace{a \cdot a \cdot a \cdots a}_{b \text{ copies of } a} \bmod n$$

`IN:` $a \in \mathbb{Z}, b, n \in \mathbb{N}$.
`OUT:` $c \equiv a^b \bmod n$.

1. $c \leftarrow 1$
2. `for` $i = 0$ `to` $b - 1$ `do`
   $c \leftarrow c \cdot a$.
3. $c \leftarrow c \bmod n$.

How not to compute modular exponentiation

Want to compute

$$c \equiv a^b \bmod n$$

for given $a \in \mathbb{Z}, b, n \in \mathbb{N}$.
By definition

$$c \equiv a^b = \underbrace{a \cdot a \cdot a \cdots a}_{b \text{ copies of } a} \bmod n$$

IN: $a \in \mathbb{Z}, b, n \in \mathbb{N}$.
OUT: $c \equiv a^b \bmod n$.

1. $c \leftarrow 1$
2. for $i = 0$ to $b - 1$ do
   $c \leftarrow c \cdot a$.
3. $c \leftarrow c \bmod n$.

IN: $a \in \mathbb{Z}, b, n \in \mathbb{N}$.
OUT: $c \equiv a^b \bmod n$.

1. $c \leftarrow 1$
2. for $i = 0$ to $b - 1$ do
   $c \leftarrow c \cdot a \bmod n$.

How not to compute modular exponentiation

Want to compute

$$c \equiv a^b \bmod n$$

for given $a \in \mathbb{Z}, b, n \in \mathbb{N}$.
By definition

$$c \equiv a^b = \underbrace{a \cdot a \cdot a \cdots a}_{b \text{ copies of } a} \bmod n$$

`IN`: $a \in \mathbb{Z}, b, n \in \mathbb{N}$.
`OUT`: $c \equiv a^b \bmod n$.

1. $c \leftarrow 1$
2. `for` $i = 0$ `to` $b - 1$ `do`
   $c \leftarrow c \cdot a$.
3. $c \leftarrow c \bmod n$.

`IN`: $a \in \mathbb{Z}, b, n \in \mathbb{N}$.
`OUT`: $c \equiv a^b \bmod n$.

1. $c \leftarrow 1$
2. `for` $i = 0$ `to` $b - 1$ `do`
   $c \leftarrow c \cdot a \bmod n$.

To see the difference, compute $2^{42} \bmod 127$:

$$\begin{aligned} 2^{42} &= 4398046511104 \\ &\equiv 1 \bmod 127 \end{aligned}$$

Numbers grow with exponent.

How not to compute modular exponentiation

Want to compute

$$c \equiv a^b \bmod n$$

for given $a \in \mathbb{Z}, b, n \in \mathbb{N}$.
By definition

$$c \equiv a^b = \underbrace{a \cdot a \cdot a \cdots a}_{b \text{ copies of } a} \bmod n$$

IN: $a \in \mathbb{Z}, b, n \in \mathbb{N}$.
OUT: $c \equiv a^b \bmod n$.

1. $c \leftarrow 1$
2. for $i = 0$ to $b - 1$ do
   $c \leftarrow c \cdot a$.
3. $c \leftarrow c \bmod n$.

IN: $a \in \mathbb{Z}, b, n \in \mathbb{N}$.
OUT: $c \equiv a^b \bmod n$.

1. $c \leftarrow 1$
2. for $i = 0$ to $b - 1$ do
   $c \leftarrow c \cdot a \bmod n$.

To see the difference, compute $2^{42} \bmod 127$:

$$\begin{aligned} 2^{42} &= 4398046511104 \\ &\equiv 1 \bmod 127 \end{aligned}$$

Numbers grow with exponent.

We see $1, 2, 4, 8, 16, 32, 64$,
$128 \equiv 1 \bmod 127, 2, 4, 8, 16, 32, 64$
$1, 2, 4, 8, 16, \ldots 1$
No number larger than 128.

Right–to–Left Binary

IN: Non-zero positive integers a, b, n, with $b = (b_{\ell-1} \dots b_0)_2$.
OUT: $c \equiv a^b \bmod n$.

1. $c \leftarrow 1, t \leftarrow a$,
2. for $i = 0$ to $\ell - 1$ do
   2.1 if $b_i = 1$ then $c \leftarrow c \cdot t \bmod n$
   2.2 $t \leftarrow t^2 \bmod n$
3. return c

Example
$42 = (101010)_2 = 2^5 + 2^3 + 2^1$, so $\ell = 6$ is minimal
We see the following intermediate states of (c, t):
$(1, a)$ initialization
$(1, a^2)$ no 2^0 contribution
(a^2, a^4) has 2^1
(a^2, a^8) no 2^2 contribution
(a^{10}, a^{16}) has 2^3
(a^{10}, a^{32}) no 2^4 contribution
(a^{42}, a^{64}) has 2^5 We could have skipped computing a^{64}.

Left–to–Right Binary

IN: Non-zero positive integers a, b, n, with $b = (b_{\ell-1} \dots b_0)_2$.
OUT: $c \equiv a^b \bmod n$.

1. $c \leftarrow 1$
2. for $i = \ell - 1$ to 0 do
   - 2.1 $c \leftarrow c^2 \bmod n$
   - 2.2 if $b_i = 1$ then $c \leftarrow c \cdot a \bmod n$
3. return c

Example
$42 = (101010)_2 = 2^5 + 2^3 + 2^1$, so $\ell = 6$ is minimal
We see the following intermediate states of c:
1 initialization
a has 2^5
a^2 no 2^4 contribution
a^5 has 2^3
a^{10} no 2^2 contribution
a^{21} has 2^1
a^{42} no 2^0 contribution
Only 1 variable to update. Same number of squarings and multiplications.