Lecture 4 Fast Power Algorithm

We want to compute $g^A \pmod N$ for large powers A as efficiently as possible.

Fast Power Algorithm

Step 1: Write A in binary: $A = A_0 + A_1 * 2 + A_2 * 2^2 + ... + A_r * 2^r$ where each $A_i \in \{0,1\}$.

Step 2: Compute the following sequence of squares:

$a_0 \equiv g \pmod N$
$a_1 \equiv a_0^2 \pmod N$
$a_2 \equiv a_1^2 \pmod N$
$\vdots$
$a_r \equiv a_{r-1}^2 \pmod N$

Step 3: The answer is:

$$g^A \equiv g^{A_0 + A_1 * 2 + ... + A_r * 2^r} = a^{A_0} * a^{A_1} * ... * a^{A_r}$$

Ex:
(1) Compute $3^{22} \pmod{20}$
First compute 22 in binary: $22 = 16 + 4 + 2$

Compute the sequence of squares:
$3 \equiv 3 \pmod{20}$
$3^2 \equiv 9 \pmod{20}$
$3^4 \equiv 81 \equiv 1 \pmod{20}$
$3^8 \equiv 1^2 \equiv 1 \pmod{20}$
$3^{16} \equiv 1^2 \equiv 1 \pmod{20}$

Answer:
$3^{22} \equiv 9 * 1 * 1 \equiv 9 \pmod{20}$

(2) Compute $4^{201} \pmod{900}$
First compute 201 in binary: $201 = 128 + 64 + 8 + 1$

Compute the following sequence:

$$4^1 \equiv 4 \pmod{900}$$

$$4^2 \equiv 16 \pmod{900}$$

$$4^4 \equiv 16^2 \equiv 256 \pmod{900}$$

$$4^8 \equiv 256^2 \equiv 736 \pmod{900}$$

$$4^{16} \equiv 736^2 \equiv 796 \pmod{900}$$

$$4^{32} \equiv 796^2 \equiv 16 \pmod{900}$$

$$4^{64} \equiv 16^2 \equiv 256 \pmod{900}$$

$$4^{128} \equiv 256^2 \equiv 736 \pmod{900}$$

Answer:

$$4^{201} \equiv 736 * 256 * 736 * 4 \equiv 604 \pmod{900}$$