Grundlagen der theoretischen Informatik

Vorlesung 12:
Primzahltest und Kryptographie

Die folgende Operationen sind in P:
Addition, Subtraktion, Multiplikation, Division.

Exponentiation ist nicht in P.
2^n: die Länge der Eingabe ist $\sim \log_2 n$.
Länge des Outputs $\sim n$. $\Rightarrow$ Die Anzahl der Ziffern ist exponentiell.

Binäre Modulo-Exponentiation

Algorithmus

Input: $a, b, m \in \mathbb{Z}^+$, $m > 1$
Output: $a^b \pmod m$
Ablauf:
(1) Sei $K = \lfloor \log_2 b \rfloor$
(2) Man bestimmt die Binärdarstellung von b. Seien die Positionen, in der eine 1 steht: $0 \leq i_1 < i_2 < \cdots < i_t \leq K$.
(3) Man bestimmt durch sukzessives Quadrieren und Restbildung $a^2 \pmod m$, $a^{2^2} \pmod m$, $a^{2^3} \pmod m$, ..., $a^{2^K} \pmod m$.
(4) Man multipliziert $a^{i_1} a^{i_2} \ldots a^{i_K}$ modulo m.

Binäre Modulo-Exponentiation

Beispiel:

$22^{103} \equiv ?\ (mod\ 31)$ $\qquad$ $103 = 64 + 32 + 4 + 2 + 1$

$22^{2} \equiv 484 \equiv 19\ (mod\ 31)$

$22^{4} \equiv 19^{2} \equiv 361 \equiv 20\ (mod\ 31)$

$22^{8} \equiv 20^{2} \equiv 400 \equiv -3\ (mod\ 31)$

$22^{16} \equiv (-3)^{2} \equiv 9\ (mod\ 31)$

$22^{32} \equiv 9^{2} \equiv 81 \equiv 19\ (mod\ 31)$

$22^{64} \equiv 19^{2} \equiv 361 \equiv 20\ (mod\ 31)$

$22^{103} \equiv 22^{64} \cdot 22^{32} \cdot 22^{4} \cdot 22^{2} \cdot 22^{1} \equiv 20 \cdot 19 \cdot 20 \cdot 19 \cdot 22 \equiv 13\ (mod\ 31)$

Primzahltest-Verfahren

Die folgende Verfahren sind nicht Polynomial.

Probedivision

Bestimmt ob n eine Primzahl oder eine zusammengesetzte Zahl ist.
Überprüfe nacheinander, ob n mit 2, 3, 4, ..., $\sqrt{n}$ teilbar ist.
Wenn es mit keiner dieser Zahlen teilbar ist, dann ist es eine Primzahl, sonst eine zusammengesetzte Zahl.

Sieb des Eratosthenes

Findet die Primzahlen zwischen 2 und n.
Man schreibt alle ganze Zahlen von 2 bis n auf. Solange es noch unmarkierte Zahlen gibt, wiederhole:
(1) Markiere die kleinste unmarkierte Zahl als Primzahl.
(2) Markiere alle Vielfachen dieser Zahl als zusammengesetzt.

Fermatscher Primzahltest

Algorithmus

Input: $m, a \in \mathbb{Z}^+$, $m > 5$, $1 < a < m - 1$
Output: Indikation darüber ob m eine Primzahl ist.
Ablauf: Falls $ggT(a, m) = 1$, dann is m keine Primzahl. Falls $ggT(a, m) \neq 1$, dann überprüfe, ob $a^{m-1} \equiv 1(\mod m)$ gilt. Wenn ja, ist das Ergebnis: „m kann eine Primzahl sein". Sonst ist das Ergebnis: „m ist keine Primzahl".

Man probiert diesen Algorithmus für viele (z.B. 100) a Zahlen. Falls er keine a findet für die das Output „nein" ist, ist m wahrscheinlich eine Primzahl.

$\varphi(m) = m - 1$ falls m eine Primzahl ist.

Satz (Euler-Fermat)

Seien $a, m \in \mathbb{Z}$, $m > 1$, $ggT(a, m) = 1$. Dann gilt:
$a^{\varphi(m)} \equiv 1(\mod m)$.

$a^{m-1} \not\equiv 1(\mod m) \Rightarrow m$ ist keine Primzahl.

Wenn der Fermatscher Primzahltest sagt, dass m keine Primzahl ist, dann ist es wirklich keine Primzahl. Aber falls das Output immer „m kann eine Primzahl sein" ist, dann ist m nicht notwendig eine Primzahl.

Carmichael-Zahl

n ist eine Carmichael-Zahl falls es keine Primzahl ist und $a^{n-1} \equiv 1(\mod n)$ gilt für alle $ggT(a, n) = 1$.

Beispiele: 561 und 1105.

Kryptographie

RSA-Algorithmus: Vorbereitung

(1) Seien p und q zwei große Primzahlen. Sei $m = p \cdot q$, dann $\varphi(m) = (p-1)(q-1)$.
(2) Sei e eine Zahl mit $ggT(e, \varphi(m)) = 1$.
(3) Die Lösung von $ex \equiv 1(\mod \varphi(m))$ sei d.
(4) Der öffentliche Schlüssel ist (m, e), der private Schlüssel: $(p, q, \varphi(m), d)$.

RSA-Algorithmus: Verschlüsselung

Input: Zu verschlüsselnde Nachricht $x(\mod m)$.
Output: Verschlüsselte Nachricht y.
Ablauf: $y \equiv x^e(\mod m)$.

RSA-Algorithmus: Entschlüsselung

Input: Verschlüsselte Nachricht $y \equiv x^e (\mod m)$.
Output: Unverschlüsselte Nachricht x.
Ablauf: Sei $x \equiv (x^e)^d (\mod m)$.

Beispiel:
Hänsel möchtet einen Nachricht von Gretel bekommen. Hänsel wählt die volgende Zahlen:
Seien $p = 7$, $q = 11$. $m = p \cdot q = 77$, $\varphi(77) = 60$.
Sei $e = 13$.
Die Lösung von $13x \equiv 1 \ (mod\ 60)$ ist $d = 37$.
Er sendet die Zahlen $m = 77$, $e = 13$ nach Gretel. Die Hexe kann diese überhören aber sie kann diese Zahlen nicht benutzen um die Code zu brechen. (Diese Zahlen sind zu klein, man benutzt typischerweise mindestens eine 600-stellige m.)
Gretel hat Nachricht $x = 5$. Sie kalkuliert $5^{13} \equiv 26 \ (mod\ 77)$ und sendet 26. Die Hexe kann auch dieser überhören.
Dann kalkuliert Hänsel $26^{37} \equiv 5 \ (mod\ 77)$.