Klausurtermin

Klausur Diskrete Mathematik I

- Do. 28.02.2008
- 3-stündig

Wiederholung

- Komplexität modularer Arithmetik
  - Addition: $\mathcal{O}(n)$
  - Multiplikation: $\mathcal{O}(n^2)$ bzw. $\mathcal{O}(n^{\log_2 3})$
  - Exponentiation: $\mathcal{O}(n^3)$ bzw. $\mathcal{O}(n^{1+\log_2 3})$
- Kleiner Satz von Fermat
  - Carmichael-Zahlen
  - Probabilistischer Primzahltest außer für Carmichael-Zahlen
- Chinesischer Restsatz

Chinesischer Restsatz

Spezieller CRT-Satz: Seien m,n $\in \mathbb{N}$ teilerfremd.

Dann existiert genau eine Lösung x mod mn des Gleichungssystems

$$\left| \begin{array}{rcl} x & = & a \bmod m \\ x & = & b \bmod n \end{array} \right| .$$

Existenz:

- EEA liefert r,s mit mr + ns = 1
  $\Rightarrow$ mr = 1 mod n und ns = 1 mod m
- Sei x = ans + bmr mod mn.
  $\Rightarrow$ x = a mod m und x = b mod n

Eindeutigkeit mod mn: Sei x‘ zweite Lösung.

- x = a = x‘ mod m und x = b = x‘ mod n
  $\Rightarrow$ m | x-x‘ und n | x-x‘
  $\Rightarrow$ mn | x-x‘ (wegen m,n teilerfremd)
  $\Rightarrow$ x = x‘ mod mn

Allgemeiner Chinesischer Restsatz

CRT-Satz: Seien $m_1, m_2, \ldots, m_n \in \mathbb{Z}$ teilerfremd.
Dann existiert genau eine Lösung x mod $m_1 * \ldots * m_n$ des Gleichungssystems

$$\left| \begin{array}{rcl} x & = & a_1 \bmod m_1 \\ x & = & a_2 \bmod m_2 \\ & \vdots & \\ x & = & a_n \bmod m_n \end{array} \right| .$$

Induktion über n:
IA: n=2: spezieller CRT-Satz
IS: n-1 $\rightarrow$ n:

- Induktionshypothese liefert eindeutige Lösung y mod $m_1 * \ldots * m_{n-1}$
- Berechnen mit speziellem CRT-Satz eindeutige Lösung x von

$$\left| \begin{array}{rcl} x & = & y \bmod m_1 * \ldots * m_{n-1} \\ x & = & a_n \bmod m_n \end{array} \right|$$

1. CRT-Isomorphismus

Satz: Sei $N=m_1*\ldots*m_n$ für teilerfremde m_i. Dann gilt:

$$\mathbb{Z}_N \cong \mathbb{Z}_{m_1} \times \ldots \times \mathbb{Z}_{m_n}.$$

- Betrachten f: $\mathbb{Z}_N \to \mathbb{Z}_{m_1} \times \ldots \times \mathbb{Z}_{m_n}$,
  $x \bmod N \mapsto (x \bmod m_1, \ldots, x \bmod m_n)$
- f ist bijektiv:
  - f ist injektiv nach CRT-Satz und
    $|\mathbb{Z}_N| = N = m_1*\ldots*m_n = |\mathbb{Z}_{m_1}| * \ldots * |\mathbb{Z}_{m_n}|$
- f ist Homomorphismus, denn
  - $f(x + y) = ((x+y \bmod N) \bmod m_1, \ldots, (x+y \bmod N) \bmod m_n)$
    $= (x+y \bmod m_1, \ldots, x+y \bmod m_n)$
    $= f(x) + f(y)$
- Damit ist f ein Isomorphismus.
- f ist effizient berechenbar, f^{-1} ebenfalls mit CRT-Konstruktion.

Korollar: Sei $N=m_1*\ldots*m_n$ für teilerfremde m_i. Dann gilt für alle $x,a \in \mathbb{Z}$:

$x=a \bmod N \Leftrightarrow x=a \bmod m_i$ für $1 \leq i \leq n$.

2. CRT-Isomorphismus

Satz: Sei $N=m_1*\ldots*m_n$ für teilerfremde m_i. Dann gilt:

$$\mathbb{Z}_N^* \cong \mathbb{Z}_{m_1}^* \times \ldots \times \mathbb{Z}_{m_n}^*.$$

- Selber Isomorphismus f wie zuvor.
- ggT(a,N)=1 $\Rightarrow$ ax+ny =1
  $\Rightarrow$ ax+$m_1*\ldots*m_n$y = 1
  $\Rightarrow$ ggT(a,m_i)=1 für $1 \leq i \leq n$
- Rückrichtung ggT(a,m_i)=1 für $1 \leq i \leq n \Rightarrow$ ggT(a,N)=1 folgt aus dem Satz zur Teilerfremdheit.
- $|\mathbb{Z}_N^*| = \phi(N) = \phi(m_1*...*m_n) = \phi(m_1)*\ldots*\phi(m_n) = |\mathbb{Z}_{m_1}^*|*\ldots*|\mathbb{Z}_{m_n}^*|$ für teilerfremde m_i (Übungsaufgabe)

Anzahl Nullstellen modularer Gleichungen

Satz: Sei $N=p_1^{e_1}*\ldots*p_k^{e_k}$, $p_i^{e_i}>2$, paarweise teilerfremd. Dann existieren 2^k Lösungen der Gleichung $x^2=1 \bmod N$ in $\mathbb{Z}_N^*$.

- $\mathbb{Z}_n^* \cong \mathbb{Z}_{p_1^{e_1}}^* \times \ldots \times \mathbb{Z}_{p_k^{e_k}}^*$
- $x^2=1 \bmod N \Leftrightarrow x^2=1 \bmod p_i^{e_i}$ für $1 \leq i \leq k$.
- $x=\pm 1$ sind Lösungen für jede Gleichung $x^2=1 \bmod p_i^{e_i}$
  - $-1= p_i^{e_i}-1 \neq 1 \bmod p_i^{e_i}$ wegen $p_i^{e_i} \geq 2$.
- D.h. $(x \bmod p_1^{e_1},\ldots,x \bmod p_k^{e_k}) \in \{-1,1\}^k$ sind Lösungen
  - 2^k verschiedene Vektoren
  - 2^k verschiedene Lösungen nach CRT-Satz

Korollar: Sei $N=pq$, p,q teilerfremd. Dann existieren vier Lösungen der Gleichung $x^2=1 \bmod N$ in $\mathbb{Z}_N^*$.

Faktorisieren mit nicht-trivialen Wurzeln

Satz: Sei N=pq mit teilerfremden p,q > 2. Sei $x \neq \pm 1$ eine Lösung von $x^2 = 1 \bmod N$. Dann kann die Faktorisierung von N in die Faktoren p,q in Zeit $\mathcal{O}(\log^2 N)$ bestimmt werden.

- $x^2 = 1 \bmod N$ besitzt die Wurzeln 1=(1,1), -1=(-1,1), (1,-1) und (-1,1).
- ObdA sei x=(1,-1), d.h. $x=1 \bmod p$, $x=-1 \bmod q$.
  $\Rightarrow x-1 = 0 \bmod p$ und $x-1 = -2 \bmod q$ ($x-1 \neq 0 \bmod q$ wegen $q \neq 2$)
  $\Rightarrow$ ggT(N,x-1)=p
- Analog gilt ggT(N, x+1)=q

Bemerkung: Alle modernen Faktorisierungsalgorithmen (Quadratisches Sieb, Zahlkörpersieb) suchen nach nicht-trivialen Wurzeln der 1 mod N.

RSA-Verfahren

Alice besitzt

- öffentliche Parameter: N=pq, p,q prim und $e \in \mathbb{Z}^*_{\phi(N)}$
- geheimen Parameter: $d \in \mathbb{Z}^*_{\phi(N)}$

Algorithmus Schlüsselgenerierung

Eingabe: 1^k (k ist Sicherheitsparameter)

1. p,q $\leftarrow$ Wähle solange zufällige k-Bitzahlen bis beide prim sind (Primzahltest)
2. N $\leftarrow$ p*q
3. ϕ(N) $\leftarrow$ (p-1)(q-1)
4. e $\leftarrow$ Wähle zufällige Zahl e‘ in $\mathbb{Z}_{\phi(N)}$ bis ggT(e‘,ϕ(N)=1
5. d $\leftarrow$ e^{-1} mod N

Ausgabe: (N,e,d)

Erwartete Laufzeit: $\mathcal{O}(k^3*k) + \mathcal{O}(k^2) + \mathcal{O}(k^2) + \mathcal{O}(k^2) + \mathcal{O}(k^2) = \mathcal{O}(k^4)$

polynomiell im Sicherheitsparameter k

Ver- und Entschlüsselung

- Bob verschlüsselt Message m als Ciphertext $c=m^e \bmod N$
  - Beachte: Bob verwendet öffentliche Parameter (N,e).
- Alice entschlüssel Ciphertext c zur Message $m=c^d=m^{ed} \bmod N$
- Laufzeit Ver-/Entschlüsselung: $\mathcal{O}(\log^3 N)$ mit Square&Multiply Algorithmus.

Satz: Sei N=pq und $e,d \in \mathbb{Z}^*_{\phi(N)}$ mit $ed=1 \bmod \phi(N)$. Für alle $m \in \mathbb{Z}_N$ gilt:

$$m^{ed} = m \bmod N$$

- CRT: $m^{ed}=m \bmod N \Leftrightarrow m^{ed}=m \bmod p$ und $m^{ed}=m \bmod q$
- Zeigen Gleichung mod p, analog mod q.
  - $ed=1 \bmod \phi(N)$, d.h. $ed = 1+k\phi(N)$
  - $m^{ed} = m^{1+k\phi(N)} = m * (m^{p-1})^{k(q-1)} = m \bmod p$ für $m \in \mathbb{Z}_p^*$
    (Kleiner Satz von Fermat)
  - $0^{ed} = 0 \bmod p$ für m=0.

Sicherheit von RSA

Satz: Wir definieren die folgenden Probleme:

1. **Faktorisierung von N**
2. **Berechnen von ϕ(N)**
3. **Berechnen von d aus (N,e)**

Sei A ein Algorithmus mit polynomieller Laufzeit $\mathcal{O}(\log^c N)$, $c \in \mathbb{N}$, für eines der Probleme.

Dann besitzen alle drei Probleme polynomielle Komplexität.

- 1. $\Rightarrow$ 2. $\Rightarrow$ 3. folgt aus Algorithmus Schlüsselgenerierung
- Bleibt zu zeigen: 3. $\Rightarrow$ 1.

Berechnen von d $\Rightarrow$ Faktorisieren von N

Beweisidee:

- Sei A Algorithmus mit A(N,e)=d.
- Sei $ed-1=k\phi(N)=k(p-1)(q-1)=2^r t$, t ungerade, $r \geq 2$

- Für beliebiges $a \in \mathbb{Z}_N^*$ gilt: $a^{2^r t}=1 \bmod N$.
  - 1.Fall: Quadratwurzeln $a^{2^{r-1}t} = \ldots = a^t = 1 \bmod N$ (triviale Quadratwurzeln)
  - 2.Fall: $a^{2^k t} = -1 \bmod N$ für $0 \leq k < r$. (triviale Quadratwurzel)
  - 3.Fall: $a^{2^k t} \neq \pm 1 \bmod N$ und $a^{2^{k+1} t}=1 \bmod N$ (nicht-triviale Quadratwurzel)
    - Nichttriviale Quadratwurzel der Eins gefunden.
    - ggT($a^{2^k t} \pm 1$, N) liefert Faktoren p,q von N.

- Man kann zeigen, dass der 3.Fall für zufällige Wahl von $a \in \mathbb{Z}_N^*$ mit Wahrscheinlichkeit $\geq$ ½ eintritt.
- D.h. man muss im Erwartungswert zwei zufällige a wählen, um die Faktorisierung zu bestimmen.