Aplikace modulární aritmetiky
Algoritmus RSA

Matematické algoritmy (K611MA)

Jan Přikryl, Miroslav Vlček

Ústav aplikované matematiky
Fakulta dopravní ČVUT

6. přednáška K611MA
čtvrtek 15. listopadu 2007

Obsah přednášky

Čínská věta o zbytcích

Opakování z minula

Nechť $n_1, n_2, \ldots, m_k$ jsou navzájem nesoudělná přirozená čísla, $n_i \geq 2$ pro všechna $i = 1, \ldots, k$. Potom řešení soustavy rovnic

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ x &\equiv a_2 \pmod{n_2} \\ &\vdots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

existuje a je určeno jednoznačně v modulo $n = n_1 \cdot n_2 \cdot \ldots \cdot m_k$.

K čemu to vlastně všechno je

Krátká motivační vložka

Díky nesoudělnosti modulů $n_1 \cdot n_2 \cdot \ldots \cdot m_k$ existuje ve třídě operací modulo n_i ke každému $N_i = n/n_i$ jeho multiplikativní inverze M_i, tedy

$$M_i \cdot N_i \equiv 1 \pmod{n_i}$$

a platí

$$x = \sum_{i=1}^{k} a_i M_i N_i.$$

Problém nůše s vejci

Ilustrace použití CRT

V nůši je x vajec. Pokud z ní odebíráme vejce po dvou, třech a pěti najednou, v nůši nakonec zůstane 1, 2, respektive 4 vejce. Pokud odebíráme vejce po sedmi kusech, v nůši nakonec nezůstane vejce žádné.

Jaká je nejmenší hodnota x pro niž může uvedená situace nastat?

Problém nůše s vejci

Ilustrace použití CRT

Zbytkové třídy jsou $[1]_2$, $[2]_3$, $[4]_5$ a $[0]_7$.

Společný modul $n = 210$.

Spočteme si $N_1 = 210/2 = 105$ až $N_4 = 210/7 = 30$ a jejich odpovídající modulární inverze $M_1 = 1$ až $M_3 = 3$.

Potom $x = 1 \cdot 1 \cdot 105 + 1 \cdot 2 \cdot 70 + 4 \cdot 3 \cdot 42 = 749 \bmod 210 = 119$.

Obsah přednášky

Šifrování

Symetrické a asymetrické šifry

Existují dvě základní skupiny šifrovacích algoritmů:

- **Symetrické šifry** u nichž se ten samý klíč používá jak k šifrování, tak i k dešifrování zprávy. Odesílatel i příjemce musí mít k dispozici identické klíče. Příkladem je DES, 3DES, AES.
- **Asymetrické šifry** u nichž se šifruje jiným klíčem, než je klíč určený k dešifrování. Odesílatel po zašifrování již nemá možnost zprávu dešifrovat. Příkladem je RSA (PGP), GnuPG, ElGamal.

Symetrické šifry jsou při stejné délce šifrovacího klíče výrazně bezpečnější, než šifry asymetrické.

Diffie-Hellman výměna klíčů

Jak se dohodnout na klíči přes nezabezpečený kanál

Symetrické šifrování má základní problém: distribuci klíčů.

Diffie a Hellman, 1976

Alice a Bob se na klíči mohou dohodnout přes nezabezpečený komunikační kanál. Pouze je potřeba, aby operace, které Alice a Bob provádějí, nebyly výpočetně snadno invertovatelné.

Diffie-Hellman výměna klíčů

Veřejně známé prvočíslo p a $\alpha \in \{2, \ldots, p-2\}$. Oba jako klíč použijí $\alpha^{xy} \bmod p$ – Alice si vymyslí veliké $x \in \mathbb{N}$ a Bobovi pošle $\alpha^x \bmod p$, Bob pošle Alici $\alpha^y \bmod p$. Alice pak provede $(\alpha^y)^x \bmod p$, Bob obdobně.

Šifrování veřejným klíčem
Myšlenka RSA

RSA vychází z předpokladu, že faktorizace součinu prvočísel p a q je časově náročná – všichni proto mohou znát šifrovací klíč $n = p \cdot q$, ale nepomůže jim to ke zjištění dešifrovacího klíče, založeného na p a q.

V praxi je klíč $n = p \cdot q \in \{0,1\}^{1024}$ až $\{0,1\}^{4096}$.

Poslední faktorizovaný RSA klíč je RSA-640 ($n = \{0,1\}^{640}$, 193 dekadických číslic) za 5 měsíců na 80 AMD Opteron 2,2 GHz CPU v roce 2005.

Ale pozor: V květnu 2007 padlo $M_{1039} = 2^{1039} - 1$ za 11 měsíců v laboratořích EPFL, Uni Bonn a NTT.

Algoritmy šifrování veřejným klíčem

Prerekvizity

Většina algoritmů (a RSA rozhodně) staví na několika algebraických postupech:

- Modulární mocnění
- Modulární inverze $a^{-1} \cdot a \equiv 1 \pmod{n}$
- Čínská věta o zbytcích
- Eulerova funkce

Modulární mocnění

Výpočet $c \equiv b^r \pmod{n}$

Neefektivně lze opakovaným násobením a redukcí:

$$c = b \left[b \left[\ldots \left[b^2 \bmod n \right] \ldots \right] \bmod n \right] \bmod n$$

Binární umocňování *(Exponentiaion by repeated squaring)*

Efektivní algoritmus pro $b, r \in \mathbb{N}$ je následující

1. Nechť $r = \sum_{j=0} k a_j 2^j, a_j \in \{0, 1\}$
2. Inicializujeme $c = 1 + a_0 \cdot (b - 1)$ a $b_0 = b$
3. Opakujeme pro $j = 1 \ldots k$:
   i. Spočteme $b_j = b_{j-1}^2 \bmod n$
   ii. Pokud je $a_j > 0$, přepíšeme $c \leftarrow c \cdot b_j \bmod n$
4. Výsledkem je $c \equiv b^r \pmod{n}$

Modulární mocnění
Příklad

Spočtěte $c = 3^{17} \bmod 7$

Nejprve rozložíme $r = 17 = 10001_{\mathrm{b}}$. Je $a_0 = 1$ a proto prvotní hodnota $c = b = 3$ a $b_0 = 3$. Potom

$$\begin{aligned} b_1 &= 3^2 \bmod 7 = 9 \bmod 7 = 2, a_1 = 0, \\ b_2 &= 2^2 \bmod 7 = 4 \bmod 7 = 4, a_2 = 0, \\ b_3 &= 4^2 \bmod 7 = 16 \bmod 7 = 2, a_3 = 0, \\ b_4 &= 16^2 \bmod 7 = 256 \bmod 7 = 4, a_4 = 1. \end{aligned}$$

Nyní přepočteme $c = 3 \cdot 4 \bmod 7 = 12 \bmod 7 = 5$. Další binární cifry už v r nejsou, výsledkem je proto $c = 5$.

Kontrola: $3^{17} = 129140163 \bmod 7 = 5$.

Eulerova funkce $\phi(n)$

Rozšíření Malé Fermatovy věty

Leonhard Euler generalizoval Malou Fermatovu větu na

$$a^{\phi(n)} \equiv 1 \pmod{n},$$

kde $\phi(n)$ je již zmíněná **Eulerova funkce**.

Pro nás je důležité, že pro prvočísla p a q platí

$$\phi(p \cdot q) = (p-1) \cdot (q-1) = \phi(p) \cdot \phi(q).$$

a tedy

$$\left(a^{p-1}\right)^{q-1} \equiv \left(a^{q-1}\right)^{p-1} \equiv 1 \pmod{p \cdot q}.$$

Algoritmus RSA

Generování veřejného a soukromého klíče

Přípravná fáze:

1. Zvolíme nepříliš si blízká prvočísla p a q.
2. Spočteme **modul** šifrovací a dešifrovací transformace, $n = p \cdot q$.
3. Vypočteme Eulerovu funkci pro n, $\phi(n) = (p-1)(q-1)$.
4. Zvolíme **šifrovací exponent** e takový, že $1 < e < \phi(n)$ a $\gcd(e, n) = 1$.
5. Dopočteme **dešifrovací exponent** d tak, aby d bylo multiplikativní inverzí k e modulo $\phi(n)$, $d \cdot e \equiv 1 \pmod{\phi(n)}$.

Veřejný klíč pro zašifrování zprávy je (n, e), **soukromý klíč** pro dešifrování je (n, d).

Algoritmus RSA
Jak to funguje

Princip přenosu zprávy X je primitivní:

Šifrování

Po lince přenášíme šifrovaný text c, jenž vznikne jako

$$c = X^e \bmod n.$$

Dešifrování

Příjemce si z přijatého šifrovaného textu spočítá původní zprávu jako

$$X = c^d \bmod n.$$

Trik celého postupu spočívá v tom, že z pouhé znalosti (n, e) nelze v rozumném čase určit d.

Algoritmus RSA
Důkaz (1)

Obdržíme dešifrováním opravdu původní text?

Při dešifrování $c \equiv X^e \pmod{n}$ máme

$$c^d \equiv (X^e)^d \equiv X^{ed} \pmod{n}.$$

Z definice RSA víme, že

$$ed \equiv 1 \pmod{\phi(n)} \Rightarrow \exists g \in \mathbb{Z} : ed = 1 + g\phi(n),$$

a navíc pro $n = p \cdot q$, kde p a q jsou prvočísla, ještě platí (zkuste to dokázat), že

$$ed \equiv 1 \pmod{\phi(n)} \equiv 1 \pmod{\phi(p)} \equiv 1 \pmod{\phi(q)}.$$

Algoritmus RSA
Důkaz (2)

Pro $p \nmid X$ je podle Malé Fermatovy věty $X^{p-1} \equiv 1 \pmod{p}$ a tedy

$$X^{ed} = X^{1+h(p-1)} = X^{h(p-1)}X = \left(X^{(p-1)}\right)^h X \equiv 1^h X \equiv X \pmod{p}.$$

Pro $p|X$ je

$$X^{ed} \equiv 0^{ed} \pmod{p} \equiv X \pmod{p}$$

To samé platí pro q a tedy

$$\begin{aligned} X^{ed} &\equiv X \pmod{p} \\ X^{ed} &\equiv X \pmod{q} \end{aligned}$$

a podle Čínské věty o zbytcích i $X^{ed} \equiv X \pmod{pq}$.

RSA pomocí CRT

Urychlení dešifrování (1)

Jak modul n, tak i dešifrovací exponent d jsou hodně velká čísla, a proces dešifrování C trvá dlouho. K urychlení dešifrovací transformace se používá následující rozklad na výpočet s menšími moduly:

$$\begin{aligned} c^d &\equiv X_p \pmod{p} \\ c^d &\equiv X_q \pmod{q} \end{aligned}$$

a z toho

$$\begin{aligned} X_p &\equiv c^d \pmod{p} \\ X_q &\equiv c^d \pmod{q} \end{aligned}$$

RSA pomocí CRT
Urychlení dešifrování (2)

Hodnotu d v c^d můžeme navíc v obou rovnicích většinou snížit opět pomocí Malé Fermatovy věty vyjádřením $d = d_p + h(p-1)$ respektive $d = d_q + k(q-1)$, z čehož

$$\begin{aligned} X_p &\equiv c^{d_p} \pmod{p} \\ X_q &\equiv c^{d_q} \pmod{q} \end{aligned}$$

a

$$X = [X_p M_p q + X_q M_q p] \bmod n,$$

kde $M_p = q^{-1} \bmod p$ a $M_q = p^{-1} \bmod q$.

Algoritmus RSA

Prolomení při nevhodné volbě p a q

Pokud zvolím p a q nevhodně (blízko sebe, příliš malá, atd.), útočník využije znalosti (n, e):

1. Faktorizuje n na p a q.
2. Vypočte Eulerovu funkci pro n, $\phi(n) = (p-1)(q-1)$.
3. Dopočte **dešifrovací exponent** d tak, aby $d \cdot e \equiv 1 \pmod{\phi(n)}$.

Můj **soukromý klíč** pro dešifrování (n, d) v ten okamžik zná i útočník a může moje zprávy dešifrovat.