Bernd Borchert

Univ. Tübingen WS 13/14

Vorlesung

Kryptographie

Teil 8
(Prof. Reinhardt)

Faktorisierungsalgorithmen, DL Algorithmen

20.12.13

Pollard's $p-1$ Faktorisierungsalgorithmus Ges: n zusammengesetzt

Hoffnung: n enthält einen Primfaktor p mit $p-1$ ist B-glatt

d.h. $p-1$ besteht aus Primfaktoren $\leq B$

Wähle zufälliges $2 \leq a \leq n-1$ mit $ggT(a,n)=1$ (sonst schon Faktor gefunden)

Für jede Primzahl $q \leq B$ do

$l = \left\lfloor \frac{\ln n}{\ln q} \right\rfloor$, $a := a^{(q^l)}$, $d = ggT(a-1, n)$

~~If~~ $1 < d < n$ then output d $\qquad a^{\prod \frac{1}{[illegible]}} = a^{x \cdot \phi(p)} \equiv 1 \mod p$

falls $d = n$ versuche anderes a

Quadratische Faktorisierung: Idee: Konstruiere x, y mit $x^2 - y^2 \equiv 0 \mod n$

durch zufälliges Sammeln von Quadraten und Lösen eines linearen Gleichungssystems in den Exponenten

Quadratisches Sieb: Vorauswahl von kleinen Zahlen zu erhalten, die sich besser zu Quadraten zusammensetzen lassen

Eingabe: Zusammengesetztes n nicht Primzahlpotenz Gesucht: Faktor

Wähle Faktorbasis $S = \{p_1 = -1, p_2, \ldots, p_t\}$ mit p_j ist $j-1$-te Primzahl für die n quadratischer Rest [illegible] mod p_j ist

$m = \lfloor \sqrt{n} \rfloor$

Sammle Paare (a_i, b_i) für Werte x in der Reihenfolge $0, 1, -1, 2, -2, \ldots$

$a_i := x + m$ $\quad b_i := (x+m)^2 - n$ mit b_i p_t-glatt [illegible] bis $i = t+1$

repeat

Mit linearer Algebra über $\mathbb{Z}_2$ finde eine Teilmenge $T \subseteq \{1, 2, \ldots, t+1\}$

mit $\sum_{i \in T} v_i = 0$ mit $v_{ij} = e_{ij} \bmod 2$, $\quad b_i = \prod_{j=1}^{t} p_j^{e_{ij}}$

$x := \prod_{i \in T} a_i \mod n$ $\qquad y = \prod_{j=1}^{t} p_j^{l_j} \mod n$ mit $l_j = \left(\sum_{i \in T} e_{ij}\right)/2$

falls $x \equiv \pm y$ dann neues T berechnen, ggf. weitere Paare (a_i, b_i) berechnen

until $d := ggT(x-y, n) \gtrless 1$ [illegible] return d

Bsp.: $n = 24961$ $\quad S = \{-1, 2, 3, 5, 13, 23\}$ $\quad m = \lfloor \sqrt{24961} \rfloor = 158$

i	x	$q(x)$	Faktorisierung	a_i	v_i
1	0	-312	$-2^3 \cdot 3 \cdot 13$	157	$(1,1,1,0,1,0)$
2	1	3	3	158	$(0,0,1,0,0,0)$
3	-1	-625	-5^4	156	$(1,0,0,0,0,0)$
4	2	320	$2^6 \cdot 5$	159	$(0,0,0,1,0,0)$
5	-2	-936	$-2^3 \cdot 3^2 \cdot 13$	155	$(1,1,0,0,1,0)$
6	4	960	$2^6 \cdot 3 \cdot 5$	161	$(0,0,1,1,0,0)$
7	-6	-2166	$-2^4 \cdot 3^3 \cdot 5$	151	$(1,0,1,1,0,0)$

$v_2 + v_3 + v_5 \equiv 0 \quad T = \{1,2,5\}$

$x = a_0 a_2 a_3 \bmod n \cdot 936$

$l_1 = 1, l_2 = 3, l_3 = 2, l_4 = 0, l_5 = 1, l_6 = 0$

$y = -2^3 \cdot 3^2 \cdot 13 \bmod n = 24025 = -936$

$v_3 + v_6 + v_7 \equiv 0 \quad T = \{3,6,7\}$

$x = a_3 \cdot a_6 \cdot a_7 \equiv 23405$

$l_1 = 1, l_2 = 5, l_3 = 2, l_4 = 3, l_5 = 0, l_6 = 0$

$y = -2^5 \cdot 3^2 \cdot 5^3 \bmod n = 13922$

$ggT(x - y, n) = ggT(9483, 24961) = 101$

$\frac{24961}{101} = 229$

Ist es möglich $\phi(n)$ zu berechnen ohne n zu faktorisieren? Nein

$n = r \cdot q \qquad \phi(n) = (r-1)(q-1)$

$q = \frac{n}{r} \qquad r^2 - (n - \phi(n) + 1) r + n = 0$

$n - \phi(n) + 1 \pm \sqrt{(n - \phi(n) + 1)^2 - 4n}$ / 2

Bsp. $n = 55 \quad \phi(n) = 40 \quad r^2 - (\underbrace{55 - 40 + 1}_{= 16}) r + 55$

$16 \pm \sqrt{256 - 4 \cdot 55}$ / 2 $= 8 \pm 3$

Ist es möglich den geheimen Exponenten d ohne Faktorisierung von n zu berechnen? Nein

Kann zum Schlüssel (e, n) kann der geheime Schlüssel (d, n) berechnet werden.

wähle $1 \le w \le n-1$ zufällig mit $ggT(w, n) \equiv 1$

Schreibe $e \cdot d - 1 = 2^s \cdot r$, mit r ungerade

$v = w^r \bmod n$ if $v \equiv 1$ mod n then "Fehlschlag" $\Rightarrow$ neues w

while $V \not\equiv 1 \bmod n$ do

$v_0 = V, \quad V = V^2 \bmod n$

if $v_0 \equiv -1 \bmod n$ then "Fehlschlag" else return $ggT(v_0 + 1, n)$

Ist es möglich einzelne Bits der geheimen Nachricht zu berechnen ohne die gesamte geheime Nachricht berechnen zu können? Nein

Anm. es ist möglich effizient $half(y) = \begin{cases} 0 & \text{falls } 0 \le x < n/2 \\ 1 & \text{sonst} \end{cases}$, mit x ist die geheime Nachricht zur Chiffre y

dann kann x mit folgendem Algorithmus berechnet werden:

```
for i:=0 to ⌊log2 n⌋ do
    yi := half(y)
    y := y * e_K(2) mod n
lo := 0, hi := n
for i := 0 to ⌊log2 n⌋ do
    mid := (hi + lo)/2
    if yi = 1 then lo := mid else hi := mid
x = ⌊hi⌋
```

$n = 1457$ $e = 779$ $y = 722$

$e_K(2) = 2^{779} \bmod 1457 = 946$

i	0	1	2	3	4	5	6	7	8	9	10
y_i	1	0	1	0	1	1	1	1	1	0	0

$x = \lfloor 999.55 \rfloor = 999$

Diskreter Logarithmus: Geg $n, y, a < n$, Gesucht x mit $a^x \bmod n = y$

Bsp $n=7$, $a=3$

$3^1 = 3$, $3^2 = 2$, $3^3 = 6$, $3^4 = 4$, $3^5 = 5$, $3^6 = 1$

Einfachste Methode: berechne $a, a^2, a^3 \ldots$ bis y gefunden

Aufwand: $O(n)$

Baby-step-Giant-step (Schnorr's Algorithmus)

$m = \lceil \sqrt{n} \rceil$ Berechne und sortiere alle Paare mit $i \le m$ $(i, a^{i \cdot m} \bmod n)$ nach der zweiten Komponente

For $j := 1$ to m suche $y \cdot a^{-j} \bmod n$ in der Liste

Falls $(i, y \cdot a^{-j} = a^{i \cdot m} \bmod n)$ gefunden $\Rightarrow y = a^{i \cdot m - j}$

$\Rightarrow$ Ausgabe $x = i \cdot m - j$

Zeit und Platzaufwand $O(\sqrt{n})$