

Public-Key Kryptography mit Diskreten Logarithmen

Jan Schwarz Kristine Jetzke

11.01.2005

Gliederung

Das ElGamal Kryptosystem

Algorithmen zum Lösen von Diskreten Logarithmen

Untere Komplexitätsgrenze

Das Problem des Diskreten Logarithmus

Gegeben:

Eine multiplikative Gruppe (G, \cdot) , ein Element $\alpha \in G$ der Ordnung n , ein Element $\beta \in \langle \alpha \rangle$ mit $\langle \alpha \rangle = \{\alpha^i \mid 0 \leq i \leq n - 1\}$.

Gesucht:

a mit $0 \leq a \leq n - 1$, so dass gilt:

$$\alpha^a = \beta \Leftrightarrow a = \log_{\alpha} \beta .$$

Das ElGamal Public-Key Kryptosystem in \mathbb{Z}_p^*

Sei p eine Primzahl, so dass das Problem des Diskreten Logarithmus in (\mathbb{Z}_p^*, \cdot) schwer ist, und $\alpha \in \mathbb{Z}_p^*$ ein primitives Element.

Sei $\mathcal{P} = \mathbb{Z}_p^*$, $\mathcal{C} = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ und \mathcal{K} definiert durch:

$$\mathcal{K} = \{(p, \alpha, a, \beta) \mid \beta \equiv \alpha^a \pmod{p}\}.$$

Die Werte p , α und β sind die öffentliche Schlüssel, a ist der geheime Schlüssel.

Für $\mathcal{K} = (p, \alpha, a, \beta)$ und eine (geheime) zufällig gewählte Nummer $k \in \mathbb{Z}_{p-1}$ definiere:

$$e_{\mathcal{K}}(x, k) = (y_1, y_2)$$

mit

$$y_1 = \alpha^k \pmod{p} \text{ und } y_2 = x\beta^k \pmod{p}.$$

Für $y_1, y_2 \in \mathbb{Z}_p^*$ definiere:

$$d_{\mathcal{K}}(y_1, y_2) = y_2(y_1^a)^{-1} \pmod{p}.$$

Einfache Algorithmen

Berechne α^i solange, bis $\beta = \alpha^a$ gefunden ist.

⇒ Laufzeit: $O(n)$

Der Algorithmus von Shanks

SHANKS(G, n, α, β)

1. $m \leftarrow \lceil \sqrt{n} \rceil$
2. **for** $j \leftarrow 0$ **to** $m - 1$ **do** erzeuge α^{mj}
3. Sortiere die m Paare (j, α^{mj}) nach ihren zweiten Koordinaten.
Erhalte die Liste L1.
4. **for** $i \leftarrow 0$ **to** $m - 1$ **do** erzeuge $\beta\alpha^{-i}$
5. Sortiere die m Paare $(i, \beta\alpha^{-i})$ nach ihren zweiten Koordinaten.
Erhalte die Liste L2.
6. Finde ein Paar $(j, y) \in L1$ und ein Paar $(i, y) \in L2$.
7. $\log_{\alpha} \beta \leftarrow (mj + i) \bmod n$

Beispiel

Gesucht: $\log_3 525$ in $(\mathbb{Z}_{809}^*, \cdot) \Rightarrow \alpha = 3, n = 808, \beta = 525, m = 29$

<i>L1</i>	<i>L1 sortiert</i>																																																																	
<table style="width: 100%; border-collapse: collapse;"> <tr><td>(0, 1)</td><td>(1, 99)</td><td>(2, 93)</td><td>(3, 308)</td></tr> <tr><td>(4, 559)</td><td>(5, 329)</td><td>(6, 211)</td><td>(7, 664)</td></tr> <tr><td>(8, 207)</td><td>(9, 268)</td><td>(10, 644)</td><td>(11, 654)</td></tr> <tr><td>(12, 26)</td><td>(13, 147)</td><td>(14, 800)</td><td>(15, 727)</td></tr> <tr><td>(16, 781)</td><td>(17, 464)</td><td>(18, 632)</td><td>(19, 275)</td></tr> <tr><td>(20, 528)</td><td>(21, 496)</td><td>(22, 564)</td><td>(23, 15)</td></tr> <tr><td>(24, 676)</td><td>(25, 586)</td><td>(26, 575)</td><td>(27, 295)</td></tr> <tr><td>(28, 81)</td><td></td><td></td><td></td></tr> </table>	(0, 1)	(1, 99)	(2, 93)	(3, 308)	(4, 559)	(5, 329)	(6, 211)	(7, 664)	(8, 207)	(9, 268)	(10, 644)	(11, 654)	(12, 26)	(13, 147)	(14, 800)	(15, 727)	(16, 781)	(17, 464)	(18, 632)	(19, 275)	(20, 528)	(21, 496)	(22, 564)	(23, 15)	(24, 676)	(25, 586)	(26, 575)	(27, 295)	(28, 81)				\Rightarrow	<table style="width: 100%; border-collapse: collapse;"> <tr><td>(0, 1)</td><td>(23, 15)</td><td>(12, 26)</td><td>(28, 81)</td></tr> <tr><td>(2, 93)</td><td>(1, 99)</td><td>(13, 147)</td><td>(8, 207)</td></tr> <tr><td>(6, 211)</td><td>(9, 268)</td><td>(19, 275)</td><td>(27, 295)</td></tr> <tr><td>(3, 308)</td><td>(5, 329)</td><td>(17, 464)</td><td>(21, 496)</td></tr> <tr><td>(20, 528)</td><td>(4, 559)</td><td>(22, 564)</td><td>(26, 575)</td></tr> <tr><td>(25, 586)</td><td>(18, 632)</td><td>(10, 644)</td><td>(11, 654)</td></tr> <tr><td>(7, 664)</td><td>(24, 676)</td><td>(15, 727)</td><td>(16, 781)</td></tr> <tr><td>(14, 800)</td><td></td><td></td><td></td></tr> </table>	(0, 1)	(23, 15)	(12, 26)	(28, 81)	(2, 93)	(1, 99)	(13, 147)	(8, 207)	(6, 211)	(9, 268)	(19, 275)	(27, 295)	(3, 308)	(5, 329)	(17, 464)	(21, 496)	(20, 528)	(4, 559)	(22, 564)	(26, 575)	(25, 586)	(18, 632)	(10, 644)	(11, 654)	(7, 664)	(24, 676)	(15, 727)	(16, 781)	(14, 800)			
(0, 1)	(1, 99)	(2, 93)	(3, 308)																																																															
(4, 559)	(5, 329)	(6, 211)	(7, 664)																																																															
(8, 207)	(9, 268)	(10, 644)	(11, 654)																																																															
(12, 26)	(13, 147)	(14, 800)	(15, 727)																																																															
(16, 781)	(17, 464)	(18, 632)	(19, 275)																																																															
(20, 528)	(21, 496)	(22, 564)	(23, 15)																																																															
(24, 676)	(25, 586)	(26, 575)	(27, 295)																																																															
(28, 81)																																																																		
(0, 1)	(23, 15)	(12, 26)	(28, 81)																																																															
(2, 93)	(1, 99)	(13, 147)	(8, 207)																																																															
(6, 211)	(9, 268)	(19, 275)	(27, 295)																																																															
(3, 308)	(5, 329)	(17, 464)	(21, 496)																																																															
(20, 528)	(4, 559)	(22, 564)	(26, 575)																																																															
(25, 586)	(18, 632)	(10, 644)	(11, 654)																																																															
(7, 664)	(24, 676)	(15, 727)	(16, 781)																																																															
(14, 800)																																																																		
<i>L2</i>	<i>L2 sortiert</i>																																																																	
<table style="width: 100%; border-collapse: collapse;"> <tr><td>(0, 525)</td><td>(1, 175)</td><td>(2, 328)</td><td>(3, 379)</td></tr> <tr><td>(4, 396)</td><td>(5, 132)</td><td>(6, 44)</td><td>(7, 554)</td></tr> <tr><td>(8, 724)</td><td>(9, 511)</td><td>(10, 440)</td><td>(11, 686)</td></tr> <tr><td>(12, 768)</td><td>(13, 256)</td><td>(14, 355)</td><td>(15, 388)</td></tr> <tr><td>(16, 399)</td><td>(17, 133)</td><td>(18, 314)</td><td>(19, 644)</td></tr> <tr><td>(20, 754)</td><td>(21, 521)</td><td>(22, 713)</td><td>(23, 777)</td></tr> <tr><td>(24, 259)</td><td>(25, 356)</td><td>(26, 658)</td><td>(27, 489)</td></tr> <tr><td>(28, 163)</td><td></td><td></td><td></td></tr> </table>	(0, 525)	(1, 175)	(2, 328)	(3, 379)	(4, 396)	(5, 132)	(6, 44)	(7, 554)	(8, 724)	(9, 511)	(10, 440)	(11, 686)	(12, 768)	(13, 256)	(14, 355)	(15, 388)	(16, 399)	(17, 133)	(18, 314)	(19, 644)	(20, 754)	(21, 521)	(22, 713)	(23, 777)	(24, 259)	(25, 356)	(26, 658)	(27, 489)	(28, 163)				\Rightarrow	<table style="width: 100%; border-collapse: collapse;"> <tr><td>(6, 44)</td><td>(5, 132)</td><td>(17, 133)</td><td>(28, 163)</td></tr> <tr><td>(1, 175)</td><td>(13, 256)</td><td>(24, 259)</td><td>(18, 314)</td></tr> <tr><td>(2, 328)</td><td>(14, 355)</td><td>(25, 356)</td><td>(3, 379)</td></tr> <tr><td>(15, 388)</td><td>(4, 396)</td><td>(16, 399)</td><td>(10, 440)</td></tr> <tr><td>(27, 489)</td><td>(9, 511)</td><td>(21, 521)</td><td>(0, 525)</td></tr> <tr><td>(7, 554)</td><td>(26, 658)</td><td>(19, 644)</td><td>(11, 686)</td></tr> <tr><td>(22, 713)</td><td>(8, 724)</td><td>(20, 754)</td><td>(12, 768)</td></tr> <tr><td>(23, 777)</td><td></td><td></td><td></td></tr> </table>	(6, 44)	(5, 132)	(17, 133)	(28, 163)	(1, 175)	(13, 256)	(24, 259)	(18, 314)	(2, 328)	(14, 355)	(25, 356)	(3, 379)	(15, 388)	(4, 396)	(16, 399)	(10, 440)	(27, 489)	(9, 511)	(21, 521)	(0, 525)	(7, 554)	(26, 658)	(19, 644)	(11, 686)	(22, 713)	(8, 724)	(20, 754)	(12, 768)	(23, 777)			
(0, 525)	(1, 175)	(2, 328)	(3, 379)																																																															
(4, 396)	(5, 132)	(6, 44)	(7, 554)																																																															
(8, 724)	(9, 511)	(10, 440)	(11, 686)																																																															
(12, 768)	(13, 256)	(14, 355)	(15, 388)																																																															
(16, 399)	(17, 133)	(18, 314)	(19, 644)																																																															
(20, 754)	(21, 521)	(22, 713)	(23, 777)																																																															
(24, 259)	(25, 356)	(26, 658)	(27, 489)																																																															
(28, 163)																																																																		
(6, 44)	(5, 132)	(17, 133)	(28, 163)																																																															
(1, 175)	(13, 256)	(24, 259)	(18, 314)																																																															
(2, 328)	(14, 355)	(25, 356)	(3, 379)																																																															
(15, 388)	(4, 396)	(16, 399)	(10, 440)																																																															
(27, 489)	(9, 511)	(21, 521)	(0, 525)																																																															
(7, 554)	(26, 658)	(19, 644)	(11, 686)																																																															
(22, 713)	(8, 724)	(20, 754)	(12, 768)																																																															
(23, 777)																																																																		

$$\Rightarrow \log_{\alpha} \beta = ((29 * 10) + 19) \text{ mod } 808 = 309$$

Laufzeitbetrachtung

▶ Laufzeit:

- ▶ In 2. und 4. werden je m Elemente erzeugt, also brauchen beide Schritte je $O(m)$ Zeit.
- ▶ Die Sortierungen in 3. und 5. brauchen je $O(m \log m)$ Zeit.
- ▶ In 6. werden je zwei der m Elemente der beiden Listen verglichen, dieser Schritt braucht also $O(m)$ Zeit.

⇒ Laufzeit: $O(\sqrt{n})$

▶ Speicher:

- ▶ Es werden zwei Listen mit je m Elementen gespeichert.

⇒ Speicher: $O(\sqrt{n})$

Der Pollard Rho Algorithmus

- ▶ Der Pollard Rho Diskrete Logarithmen Algorithmus erstellt eine Folge $(x_1, a_1, b_1), (x_2, a_2, b_2), \dots$ mit $x_i = \alpha^{a_i} \beta^{b_i}$ durch Anwenden einer Funktion f und sucht nach einer Kollision $x_i = x_{2i}$ um dann $\log_{\alpha} \beta$ zu berechnen.
- ▶ Laufzeit: $O(\sqrt{n})$.
- ▶ Vorteil gegenüber Shanks: Speicherbedarf $O(1)$

Der Pohlig Hellmann Algorithmus

- ▶ Der Pohlig-Hellmann Algorithmus geht davon aus, dass die Primfaktorzerlegung von n , $n = \prod_{i=1}^k p_i^{c_i}$, bekannt ist.
- ▶ Er berechnet $a \bmod p_i^{c_i}$ für jedes i . Mit Hilfe des Chinesischen Restsatz kann daraus dann $a \bmod n$ berechnet werden.
- ▶ Laufzeit: $O(c_j \sqrt{p_j})$

Die Index Calculus Methode

- ▶ Die Index Calculus Methode lässt sich nur in \mathbb{Z}_p^* mit p Primzahl und α primitives Element modulo p anwenden.
- ▶ Sie benutzt eine Menge $\mathcal{B} = \{p_1, p_2, \dots, p_B\}$ „kleiner“ Primzahlen.
- ▶ Funktionsweise:
 1. Die Logarithmen der B Primzahlen in \mathcal{B} werden bestimmt.
 2. Mit Hilfe der nun bekannten diskreten Logarithmen der B Primzahlen wird dann $\log_\alpha \beta$ berechnet.
- ▶ Vorberechnung: $O(e^{(1+o(1))\sqrt{\ln p \ln \ln p}})$
Laufzeit: $O(e^{(1/2+o(1))\sqrt{\ln p \ln \ln p}})$

Diskreter Logarithmus Problem in $(\mathbb{Z}_n, +)$

- ▶ Reduktion des Problems auf die additive Gruppe $(\mathbb{Z}_n, +)$ mit Hilfe eines Isomorphismus $\phi : (\mathbb{Z}_n, \cdot) \rightarrow (\mathbb{Z}_n, +)$
- ▶ Das Diskrete Logarithmus Problem lässt sich wie folgt übertragen: Finde $a \in \mathbb{Z}_n$ mit $\alpha a \equiv \beta \pmod{n}$
- ▶ Schnell zu berechnen, da $a = \beta \alpha^{-1} \pmod{n}$
- ▶ Es existiert keine effiziente Methode zur Bestimmung des Isomorphismus

Ein generischer Algorithmus

- ▶ Kodierung von $(\mathbb{Z}_n, +)$ ist eine injektive Abbildung
 $\sigma : \mathbb{Z}_n \rightarrow S$
- ▶ Eingabe: $\sigma_1 = \sigma(1)$ und $\sigma_2 = \sigma(a)$, Ausgabe: a
- ▶ Erzeuge m paarweise verschiedenen Paare $(c_i, d_i) \in \mathbb{Z}_n \times \mathbb{Z}_n$
- ▶ Orakel erzeugt Kodierung $\sigma_i = \sigma((c_i + d_i a) \bmod n)$
- ▶ Falls $\sigma_i = \sigma_j$ berechne $a = (c_i - c_j)(d_j - d_i)^{-1} \bmod n$

Untere Komplexitätsgrenze

- ▶ Erfolgswahrscheinlichkeit $\leq \frac{\binom{m}{2}+1}{n}$
 - ▶ Wenn der Algorithmus a auf jeden Fall korrekt berechnet, ist die Erfolgswahrscheinlichkeit 1.
$$1 \leq \frac{\binom{m}{2}+1}{n} \Leftrightarrow n \leq \frac{m(m+1)}{2} + 1$$
 - ▶ $n = O(m^2)$, also $m = \Omega(\sqrt{n})$
- \Rightarrow Untere Komplexitätsgrenze ist $\Omega(\sqrt{n})$

Quellen



Douglas R. Stinson.

Cryptography: Theory and Practice, 2nd Edition.

Chapman & Hall/CRC, 2002.



Johannes Buchmann.

Einführung in die Kryptographie, 3., erweiterte Auflage.

Springer Verlag, 2003.



Wikipedia.

<http://www.wikipedia.de>.