Modulo-Rechnen

Das RSA Verfahren

Das asymmetrische RSA-Verfahren ist eines der ältesten und derzeit das mit Abstand wichtigste und bekannteste asymmetrische Verschlüsselungsverfahren.

Benannt ist es nach seinen Entwicklern Ron Rivest und Adi Shamir habe ich bereits mehrfach erwähnt. Zusammen mit Leonard Adleman entwickelten die beiden das Verfahren, die Reihenfolge im Namen des Verfahrens ist absichtlich nicht alphabetisch, da Adleman die von Rivest und Shamir vorgeschlagenen Verfahren "nur" mathematisch auf Schwachstellen durchleuchtete - er sah seinen Beitrag darum als geringer an, als den seiner Mitstreiter und bestand darum darauf, als letztes genannt zu werden.

Um die Funktionsweise des RSA-Verfahrens und später des Diffie-Hellman Schlüsseltauschs verstehen zu können, benötigt man etwas Mathematik.

Modulo-Rechnen

Sicherlich kennst du bereits die Modulo-Operation: Sie liefert den Rest bei einer ganzzahligen Division zweier ganzer Zahlen zurück:

```
7 mod 6 = 1
121 mod 2 = 1
15 mod 4 = 3
```

Mit diesem Wissen kann man das Modulo-Rechnen einführen: Modulo-Rechnen ist das Rechnen mit ganzen Zahlen von 0 bis zu einer größten Zahl n. Zahlen, die größer oder gleich n sind, gibt es beim Modulo-Rechnen nicht.

Wenn man beim Modulo-rechnen "zählt", wird nach `n-1` wieder bei Null angefangen zu zählen.

Die Stundenanzeige einer Digitaluhr kann man als Modulo-24-Zähler betrachten. Eine solche Anzeige zählt die Stunden von 0 bis 23 und fängt anschließend wieder bei 0 an.

Von nun an bezeichnet `n` immer eine positive ganze Zahl. `a` und `b` sind ebenfalls ganze Zahlen, die stets zwischen `0` und `n-1` liegen.

Fix Me! Zahlenkreis als Veranschaulichung

Modulo-Addition und -Subtraktion

Modulo-Addition und die Modulo-Subtraktion, entsprechen dem herkömmlichen Abziehen und Zusammenzählen, mit einer kleinen Änderung: Ist bei der **Addition** das Ergebnis **größer oder gleich**

n, zieht man n vom Ergebnis ab.

Ist bei einer Subtraktion das **Ergebnis kleiner null**, dann **zählt man n** zum Ergebnis hinzu.

Als Ergebnis erhält man also stets eine Zahl zwischen `0` und `n-1`.

Damit man Modulo-Rechenarten vom "normalen" Rechnen unterscheiden kein schreibt man am Ende einer Gleichung mit Modulo-Rechenoperationen immer `(mod n)`.

Beispiele:

```
3+5=1 (mod 7)
2+2=4 (mod 13)
3-6=6 (mod 9)
3+6=0 (mod 9)
```

(A1)

Berechne die folgenden Modulo-Additionen/Modulo-Subtraktionen

```
7+4    = __ (mod 10)
7+4    = __ (mod 6)
9+11   = __ (mod 5)
7-9    = __ (mod 7)
124-87 = __ (mod 16)
```

Modulo-Multiplikation und -Division

Das Ergebnis einer Modulo Multiplikation erhält man am einfachsten, indem man zunächst die Zahlen multipliziert und anschließend der Rest bei Division durch n berechnet:

```
5·6 = 2 (mod 7) weil 5·6=30, 30 = 4·7 Rest 2
```

Man kann sich das Vorgehen auch folgendermaßen veranschaulichen: Man multipliziert die beiden Zahlen und zieht dann so lange n ab, bis man schließlich eine Zahl zwischen `0` und `(n-1)` erhält:

```
4·5 = 6 (mod 7)
4·5 = 20-7-7 = 6 (mod 7)
```

(A2)

Berechne die folgenden Modulo-Multiplikationen

```
7*4     = __ (mod 10)
7*4     = __ (mod 6)
9*11    = __ (mod 5)
7*9     = __ (mod 7)
3*(4+8) = __ (mod 16)
```

Um eine **Modulo-Division** zu erhalten, muss man sich überlegen, ob es zu jeder Zahl a eine Zahl b gibt, für die gilt:

```
a·b=1 (mod n)
```

Dann ist b das **inverse Element** zu a, man schreibt für das inverse Element auch $\mathbf{a^{-1}}$. Eine Division ist mathematisch nämlich nichts anderes, als die Multiplikation mit dem inversen Element, auch bei unseren "normalen" Zahlen:

```
10/5 = 2
- das Invserse Element von 5 ist 1/5, weil 5·1/5 = 1
- die Division kann man damit auch als Multiplikation schreiben:
10·1/5 = 2
```

Beim Modulo-Rechnen ist das allerdings etwas komplizierter, weil es nicht zu jeder Zahl ein inverses Element gibt.

Beim Modulo-Rechnen besitzt eine Zahl a nur dann ein inverses Element, wenn a und n teilerfremd sind, also wenn sie außer 1 keinen gemeinsamen Teiler haben.

Beispiele:

- 5^{-1}(mod 7) existiert, da 5 und 7 teilerfremd sind. Es gilt: $5^{-1}=3$.
- 4^{-1}(mod 8) existiert nicht, da 4 und 8 nicht teilerfremd sind (4 teilt beide Zahlen).
- 7^{-1}(mod 10) existiert, da 10 und 7 teilerfremd sind. Es gilt: $7^{-1}=3$

Man kann relativ schnell herausfinden, ob es das inverse Element gibt, bei größeren Zahlen ist es mitunter aber etwas schwieriger dieses zu bestimmen, das gelingt jedoch mit einem angepassten euklidischen Algorithmus - für uns genügt es zunächst zu wissen, dass man das inverse Element, wenn es existiert auch bestimmen kann.

(A3)

Berechne die folgenden inversen Elemente, in der linken Spalte das Inverse `mod 15`, in der rechten Spalten das Inverse `mod 13`. Die Fragen, die du dir stellen musst, sind also beispielsweise:

- Mit welcher Zahl b muss man 4 malnehmen, damit `4 * b = 1 (mod 15)`
- Mit welcher Zahl b muss man 1 malnehmen, damit `1 * b = 1 (mod 13)`

mod 15		**mod 13**	
a	**a^{-1}**	**a**	**a^{-1}**
———————————————			
0		0	
1		1	
2		2	
3		3	
4		4	
5		5	
6		6	
7		7	
8		8	
9		9	
10		10	
11		11	
12		12	
13		13	
14		14	

Was fällt dir auf? Woran könnte das liegen?

Lösung

mod 15		**mod 13**	
a	**a^{-1}**	**a**	**a^{-1}**
———————————————————————————————————————			
0	NN	0	NN
1	1	1	1
2	8 (8*2 mod 15 =1)	2	7
3	NN	3	9 (27 mod 13 = 1)
4	4 (4*4 mod 15 =1)	4	10 (40 mod 13 = 1)
5	NN	5	8 (40 mod 13 =1)
6	NN	6	11 (66 mod 13 = 1)
7	13 (13*7 mod 15 = 91 mod 15 = 1)	7	2 (14 mod 13 =1)
8	2	8	5 (s.o.)
9	NN	9	3 (s.o.)
10	NN	10	4 (s.o.)
11	11 (11*11 mod 15 = 121 mod 15 = 1)	11	6 (s.o.)

12	NN	12	12
13	7 (s.o.)	13	NN
14	14 (196 mod 15 = 1)	14	1

Modulo-Exponentiation

Potenzen sind eine Abkürzung für mehrmaliges Multiplizieren - das funktioniert natürlich auch beim Modulo-Rechnen in gewohnter Weise:

```
3^4 = 3 · 3 · 3 · 3 (mod 7)
    = 2 · 3 · 3 (mod 7)
    = 6 · 3 (mod 7)
    = 4 (mod 7)
```

Man kann das auf bewährte Weise abkürzen, indem man zunächst die Potenz berechnet und anschließend den ganzzahligen Rest bei der Division bestimmt: $3^4=81$, $81\; mod\; 7 = 4$, also ist $3^4 = 4\; (mod\; 7)$.

Diskreter Logarithmus

Eine Umkehrung des Potenzierens ist der Logarithmus. Beim Modulo-Rechnen stellt man sich die folgende Frage (a, b und n sind gegeben):

Für welche Zahl x gilt $a^x=b\;(mod\; n)$?

Man sucht also die Hochzahl der Potenz. Diese Zahl existiert nicht immer und ist mitunter nicht einfach zu bestimmen.

Modulo-Wurzelziehen

Beim Modulo-Wurzelziehen wird zu gegebenem a, b und n eine Zahl x gesucht, für die gilt:

$x^a=b\; (mod\; n)$.

x heißt dann die a-te Wurzel von $b \;(mod\; n)$.

Um die Frage zu beantworten, wann eine Modulo-Wurzel existiert, benötigen wir die sogenannte φ-Funktion, die angibt, wie viele natürliche Zahlen, die größer als 0 und kleiner als eine Zahl n sind, zu n teilerfremd sind.

Zum Beispiel ist

- φ(3)=2, da 1 und 2 teilerfremd zu 3 sind.
- φ(6)=2, da 1 und 5 teilerfremd zu 6 sind.
- φ(7)=6, da 1, 2, 3, 4, 5 und 6 teilerfremd zu 7 sind.

Wenn **n eine Primzahl** ist, ist φ(n)=n-1

Wenn **n das Produkt zweier Primzahlen** p und q **ist**, ist φ(n) = φ(p·q) = (p-1)·(q-1)

Mathematische Details

Es gilt der folgende

Satz: Sind a und n natürliche Zahlen (a<n), dann gilt: $a^{1 (mod\; φ (n))} = a (mod\; n)$.

Mit anderen Worten: $a^b=a (mod\; n)$ gilt dann, wenn $b-1$ ein Vielfaches von φ(n) ist.

Beispiel:

- $3^3=3 (mod\;6)$, da $3=1 (mod\; (φ(6))$
- Hier ist: $a=3$, $b=3$, $n=6$, $b-1=2$ und $φ(6) = 2$. $b-1=2$ ist ein Vielfaches $φ(6) = 2$

Aus diesem Satz folgt (wenn wir beide Seiten der Gleichung mit einer Zahl i potenzieren) ein weiterer

Satz: Sind a, i und n natürliche Zahlen ($a<n$), dann gilt: $a^{i (mod\; φ(n))} = a^{i} (mod\; n)$.

Nun kann man die die a-te Modulo-Wurzel einer Zahl berechnen. Gegeben sind a, b und n, gesucht ist die Zahl x für die gilt

$x^a=b\;(mod\; n)$

Wenn a und φ(n) teilerfremd sind, kann man mit dem euklidischen Algorithmus die Zahl $c=a^{-1} (mod\; φ(n))$ berechnen. Mit dieser Zahl potenziert man beide Seiten der Gleichung von oben und erhält:

$(x^a)^{c(mod\; φ(n))} = b^{c(mod φ (n))}(mod\; n)$

Durch die Anwendung des zweiten Satzes von obenkönnen wir die rechte Seite vereinfachen:

$(x^a)^{c(mod\; φ(n))} = b^c (mod\; n)$.

Da $a·c = 1 (mod φ(n))$, erhalten wir auf der linken Seite:

$x^{1(mod \;φ(n))}=b^c (mod\; n)$.

Also ist x nach dem ersten Satz von oben:

x=b^c (mod\; n)

Unter Beachtung einiger mathematischer Sätze kann man die die a-te Modulo-Wurzel einer Zahl folgendermaßen berechnen: Gegeben sind a, b und n, gesucht ist die Zahl x für die gilt

$x^a=b\;(mod\; n)$.

Wenn man φ(n) kennt und a und φ(n) teilerfremd sind gilt $x=b^c\;(mod\;n)$.
Für c gilt dabei $c=a^{-1} (mod\; φ(n))$.

From:
https://www.info-bw.de/ -

Permanent link:
https://www.info-bw.de/faecher:informatik:oberstufe:kryptographie:rsamathe:start

Last update: **29.04.2025 06:29**