6 Zahlentheoretische Grundlagen

In diesem Abschnitt stellen wir die Hilfsmittel aus der Zahlentheorie bereit, die wir zum Verständnis der Public-Key Verfahren, die im nächsten Abschnitt vorgestellt werden, benötigen.

Satz 116 (Euler-Fermat) *Für alle $a \in \mathbb{Z}_m^*$ gilt*

$$a^{\varphi(m)} \equiv_m 1$$

Beweis: Sei $\mathbb{Z}_m^* = \{u_1, \ldots, u_{\varphi(m)}\}$ und $a \in \mathbb{Z}_m^*$ beliebig. Wegen $a \cdot u_i \not\equiv_m a \cdot u_j$ für $i \not\equiv_m j$ folgt $\mathbb{Z}_m^* = \{a \cdot u_1, \ldots, a \cdot u_{\varphi(m)}\}$. Dies impliziert $\prod_{i=1}^{\varphi(m)} u_i \equiv_m \prod_{i=1}^{\varphi(m)} a \cdot u_i \equiv_m a^{\varphi(m)} \cdot \prod_{i=1}^{\varphi(m)} u_i$. Also muss $a^{\varphi(m)} \equiv_m 1$ sein. ■

Korollar 117 (Satz von Fermat) *Ist p eine Primzahl und a eine natürliche Zahl, die nicht durch p teilbar ist, also $a \in \mathbb{Z}_p^*$, so ist $a^{p-1} - 1$ durch p teilbar:*

$$a^{p-1} \equiv_p 1.$$

6.1 Diskrete Logarithmen

Nehmen wir ein beliebiges Element a aus $\mathbb{Z}_m^*$ und betrachten die Folge $a^0 = 1$, $a^1 = a$, a^2, a^3, …, so wissen wir nach dem Satz von Euler-Fermat, dass spätestens für $e = \varphi(m)$ wieder $a^e = 1$ gilt.

Definition 118 (Ordnung)

Es sei $m \geq 1$ und $\mathrm{ggT}(a, m) = 1$. Die **Ordnung** von a modulo m ist

$$\mathrm{ord}_m(a) = \min\{e > 1 \mid a^e \equiv_m 1\}.$$

Für das folgende besonders interessant sind Elemente a aus $\mathbb{Z}_m^*$, die ganz $\mathbb{Z}_m^*$ aufspannen.

Definition 119 (Primitivwurzel/Erzeuger)

Eine Zahl g heißt **Primitivwurzel** modulo m (oder **Erzeuger** von $\mathbb{Z}_m^*$), falls $\mathbb{Z}_m^* = \{g^0, g^1, g^2, \ldots, g^{\varphi(m)-1}\}$ ist.

Ein Element $a \in \mathbb{Z}_m^*$ ist also genau dann ein Erzeuger, wenn $\mathrm{ord}_m(a) = \varphi(m)$ ist. Falls $\mathbb{Z}_m^*$ einen Erzeuger besitzt, wird $\mathbb{Z}_m^*$ auch **zyklisch** genannt. Da $\{a^0, a^1, a^2, \ldots, a^{\mathrm{ord}_m(a)-1}\}$ eine Untergruppe von $\mathbb{Z}_m^*$ ist, ist $\mathrm{ord}_m(a)$ für alle $a \in \mathbb{Z}_m^*$ ein Teiler von $\varphi(m)$, d. h. $\varphi(m) \equiv_{\mathrm{ord}_m(a)} 0$. Allgemeiner gilt (siehe Übungen)

$$a^i \equiv_m a^j \Leftrightarrow i \equiv_{\mathrm{ord}_m(a)} j.$$

Satz 120 (Gauß) *Genau für* $m \in \{1, 2, 4, p^k, 2p^k \mid 2 < p \text{ prim}\}$ *ist* $\mathbb{Z}_m^*$ *zyklisch. (Ohne Beweis)*

Wählen wir als Basis einen Erzeuger g von $\mathbb{Z}_m^*$, so ist die Exponentiation $e \mapsto g^e$ eine bijektive Abbildung von der Menge $\{0, 1, \ldots, \varphi(m) - 1\}$ auf $\mathbb{Z}_m^*$. Die zugehörige Umkehrabbildung spielt eine sehr wichtige Rolle in der Kryptographie.

Definition 121 **(Index/diskreter Logarithmus)**

Sei g ein Erzeuger von $\mathbb{Z}_m^*$ und $a \in \mathbb{Z}_m^*$. Dann heißt der eindeutig bestimmte Exponent $e \in \{0, 1, \ldots, \varphi(m) - 1\}$ mit

$$g^e \equiv_m a$$

Index oder **diskreter Logarithmus** modulo m von a zur Basis g (kurz: $e = \log_{m,\,g}(a)$).

Während die diskrete Exponentialfunktion $e \mapsto g^e$ durch **Wiederholtes Quadrieren und Multiplizieren** (siehe nächsten Abschnitt) effizient berechnet werden kann, sind bis heute keine effizienten Verfahren zur Berechnung des diskreten Logarithmus bekannt.

Beispiel 122 $m = 11$, $g = 2$.

e	0	1	2	3	4	5	6	7	8	9
2^e	1	2	4	8	5	10	9	7	3	6

a	1	2	3	4	5	6	7	8	9	10
$\log_{11,2}(a)$	0	1	8	2	4	9	7	3	6	5

◁

Die nächsten beiden Sätze benötigen wir zur Bestimmung der Anzahl aller Erzeuger von $\mathbb{Z}_p^*$, falls p prim ist.

Satz 123 (Euler) *Sei* $m \geq 1$, *dann gilt*

$$\sum_{d|m} \varphi(d) = m,$$

wobei die Summe über alle Teiler $d \geq 1$ *von* m *läuft.*

Beweis: Mit $\varphi_d(m) = \|\{a \in \mathbb{Z}_m \mid \text{ggT}(a, m) = d\}\|$ ist $\sum_{d|m} \varphi_d(m) = m$. Wegen

$$\text{ggT}(a, m) = d \quad \Longleftrightarrow \quad \text{ggT}(a/d,\, m/d) = 1$$

ist $\varphi_d(m) = \varphi(m/d)$ und damit

$$\sum_{d|m} \varphi(d) = \sum_{d|m} \varphi(m/d) = \sum_{d|m} \varphi_d(m) = m. \qquad \blacksquare$$

Satz 124 (Lagrange) *Sei* p *eine Primzahl und sei* $\text{ggT}(a_n, p) = 1$. *Dann hat die Polynomkongruenz*

$$f(x) = a_n \cdot x^n + \cdots + a_1 \cdot x^1 + a_0 \cdot x^0 \quad \equiv_p \quad 0 \tag{6}$$

höchstens n *verschiedene Lösungen modulo* p.

Beweis: Durch Induktion über n (es wird nur die Körpereigenschaft von $\mathbb{Z}_p$ benutzt).

$n = 0$: klar.

$n - 1 \rightsquigarrow n$: Angenommen, die Polynomkongruenz (6) habe $n + 1$ verschiedene Lösungen $x_1, \ldots, x_{n+1}$ modulo p. Mit

$$\begin{aligned} & x^i - x_{n+1}^i \\ &= \underbrace{x^i + x_{n+1}x^{i-1} + \cdots + x_{n+1}^{i-1}x}_{xh_i(x)} \underbrace{-x_{n+1}x^{i-1} - \cdots - x_{n+1}^{i-1}x - x_{n+1}^i}_{-x_{n+1}h_i(x)} \\ &= (x - x_{n+1}) \underbrace{(x^{i-1} + x_{n+1}x^{i-2} + \cdots + x_{n+1}^{i-2}x + x_{n+1}^{i-1})}_{h_i(x)} \end{aligned}$$

folgt

$$\begin{aligned} g(x) &= f(x) - f(x_{n+1}) \\ &= a_n \cdot (x^n - x_{n+1}^n) + \cdots + a_1 \cdot (x - x_{n+1}) \\ &= (x - x_{n+1}) \underbrace{\sum_{i=1}^{n} a_i h_i(x)}_{h(x)} . \end{aligned}$$

Aus $g(x_j) = f(x_j) - f(x_{n+1}) \equiv_p 0$ und $x_j - x_{n+1} \not\equiv_p 0$ folgt $h(x_j) \equiv_p 0$ (für $j = 1, \ldots, n$), was im Widerspruch zur Induktionsvoraussetzung steht, da das Polynom $h(x)$ vom Grad $n - 1$ ist. ■

Nun können wir zeigen, dass $\mathbb{Z}_p^*$ für primes p zyklisch ist. Tatsächlich folgt mit demselben Beweis, dass die multiplikative Gruppe $\mathbb{F}_{p^n}^*$ des endlichen Körpers $\mathbb{F}_{p^n}$ genau $\varphi(p^n - 1)$ Erzeuger enthält.

Satz 125 (Gauß) *Ist p prim, so gibt es genau $\varphi(p - 1)$ Erzeuger von $\mathbb{Z}_p^*$.*

Beweis: Für jeden Teiler d von $\varphi(p) = p - 1$ sei

$$S_d = \{a \in \mathbb{Z}_p^* \mid \mathrm{ord}_p(a) = d\}.$$

Da die Menge S_{p-1} aus allen Erzeugern von $\mathbb{Z}_p^*$ gebildet wird, müssen wir $\|S_{p-1}\| = \varphi(p-1)$ zeigen. Zuerst überlegen wir uns, dass die Menge S_d für ein beliebiges $a \in S_d$ in der Menge $\{a^e \mid e \in \mathbb{Z}_d^*\}$ enthalten ist, woraus $\|S_d\| \leq \varphi(d)$ folgt.

Jedes $a \in S_d$ erfüllt die Kongruenz $x^d \equiv_p 1$, die nach dem Satz von Lagrange (Satz 124) höchstens d verschiedene Lösungen modulo p besitzt. Da neben a auch $a^2, \ldots, a^d$ Lösungen dieser Kongruenz sind, die alle modulo p verschieden sind, folgt $S_d \subseteq \{a, a^2, \ldots, a^d\}$. Nun folgt aus der Annahme, dass in S_d eine Potenz a^e mit $\mathrm{ggT}(e, d) = g > 1$ enthalten ist, dass $(a^e)^{d/g} \equiv_p (a^d)^{e/g} \equiv_p 1^{e/g} \equiv_p 1$ ist, was im Widerspruch zu $\mathrm{ord}_p(a^e) = d$ steht.

Da die Mengen S_d eine Partition von $\mathbb{Z}_p^*$ bilden, folgt mit Satz 123, dass

$$\sum_{d|(p-1)} \|S_d\| = p - 1 = \sum_{d|(p-1)} \varphi(d)$$

ist. Da aber, wie oben gezeigt, $\|S_d\| \leq \varphi(d)$ ist, muss $\|S_d\| = \varphi(d)$ und insbesondere $\|S_{p-1}\| = \varphi(p-1)$ gelten. ■

Satz 126 *Sei p prim. Dann ist $a \in \mathbb{Z}_p^*$ genau dann ein Erzeuger, wenn für jeden Primteiler q von $p-1$ gilt:*

$$a^{(p-1)/q} \not\equiv_n 1.$$

Beweis: Falls $a \in \mathbb{Z}_p^*$ ein Erzeuger ist, so gilt $a^e \not\equiv_p 1$ für alle Exponenten $e \in \{1, \ldots, p-2\}$ und somit auch für alle Exponenten e der Form $(p-1)/q$, q prim.

Ist dagegen $a \in \mathbb{Z}_p^*$ kein Erzeuger, so ist $\mathrm{ord}_p(a) < p-1$, und da $\mathrm{ord}_p(a)$ ein Teiler von $p-1$ ist, existiert eine Zahl $d \geq 2$ mit $d \cdot \mathrm{ord}_p(a) = p-1$. Sei q ein beliebiger Primteiler von d. Dann gilt

$$a^{(p-1)/q} \equiv_p a^{d\,\mathrm{ord}_p(a)/q} \equiv_p (a^{\mathrm{ord}_p(a)})^{d/q} \equiv_p 1.$$

■

Folgender probabilistische Algorithmus berechnet einen Erzeuger $a \in \mathbb{Z}_p^*$, falls alle Primteiler q von $p-1$ bekannt sind.

Algorithmus 127 COMPUTEGENERATOR$(p, q_1, \ldots, q_k)$

1 **Eingabe:** Primzahl p, Primteiler $q_1, \ldots, q_k$ von $p-1$
2 **repeat**
3 **rate zufällig** $a \in \{2, \ldots, p-1\}$
4 **until** $a^{(p-1)/q_i} \not\equiv_p 1$ für $i = 1, \ldots, k$
5 **Ausgabe:** a

Da $\varphi(n) \geq n/(2 \ln \ln n)$ für hinreichend große n ist, findet der Algorithmus in jedem Schleifendurchlauf mit Wahrscheinlichkeit $\varphi(p-1)/(p-2) \geq 1/(2 \ln \ln(p-1))$ einen Erzeuger. Die erwartete Anzahl der Schleifendurchläufe ist also $O(\ln \ln p)$.

6.2 Modulares Potenzieren

Modulare Potenzen $a^e \bmod m$ lassen sich durch **Wiederholtes Quadrieren und Multiplizieren** in $O(\log^2 a + \log e \log^2 m)$ Schritten berechnen (Zeitkomplexität $O(n^3)$, wobei n die Länge der Eingabe in Binärdarstellung ist).

Dazu sei $e = \sum_{i=0}^{r} e_i \cdot 2^i$ mit $r = \lfloor \mathrm{ld}\, e \rfloor$, d.h. die e_i bilden die Binärdarstellung von e. Dann können wir den Exponenten e sukzessive mittels $b_0 = e_0$ und $b_{i+1} = b_i + e_i 2^i$ für $i = 0, \ldots, r-1$ zu $e = b_r$ berechnen. Der folgende Algorithmus berechnet nach diesem Schema in der Variablen z die Potenzen $a^{b_i} \bmod m$ für $i = 0, \ldots, r$.

Algorithmus 128 MODPOT(a, e, m)

```
1  y ← a
2  z ← a^{e_0} mod m
3  for i ← 1 to r do
4    y ← y^2 mod m
5    z ← z · y^{e_i} mod m
6  end
7  return z
```

Beispiel 129 *Sei* $a = 1920$, $e = 19$ *und* $m = 2773$. *Dann erhalten wir mit obigem Algorithmus* $1920^{19} \bmod 2773 = 1868$*:*

i	e_i	$y = a^{2^i}$	$z = a^{b_i}$
0	1	1920	1920
1	1	1083	2383
2	0	2683	2383
3	0	2554	2383
4	1	820	**1868**

wobei $b_i = \sum_{j=0}^{i} e_j \cdot 2^j$. ◁

Alternativ können wir auch das **Horner-Schema** zur Berechnung von e benutzen: Sei $c_r = e_r = 1$ und sei $c_{i-1} = 2c_i + e_{i-1}$ für $i = r, \ldots, 1$, dann ist $e = c_0$. Dies führt auf folgenden Algorithmus, der in der Variablen z die Potenzen $a^{c_i} \bmod m$ für $i = r, \ldots, 0$ berechnet.

Algorithmus 130 MODPOT$^*(a, e, m)$

```
1  z ← 1
2  for i ← r downto 0 do
3    z ← z^2 · a^{e_i} mod m
4  end
5  return z
```

Beispiel 131 *Mit diesem Algorithmus erhalten wir natürlich ebenfalls* $1920^{19} \bmod 2773 = 1868$*:*

i	e_i	$z = a^{c_i}$
4	1	1920
3	0	1083
2	0	2683
1	1	1016
0	1	**1868**

wobei $c_i = \sum_{j=i}^{4} e_j \cdot 2^{j-i}$. ◁

6.3 Quadratische Reste

In diesem Abschnitt beschäftigen wir uns mit dem Problem, Lösungen für eine quadratische Kongruenzgleichung

$$x^2 \equiv_m a \tag{7}$$

zu bestimmen. Zunächst gehen wir der Frage nach, wie sich feststellen lässt, ob überhaupt Lösungen existieren.

Definition 132 (**Quadratischer (Nicht-)Rest, Legendre-Symbol**)

Ein Element $a \in \mathbb{Z}_m^*$ heißt **quadratischer Rest** modulo m (kurz: $a \in \mathsf{QR}_m$), falls ein $x \in \mathbb{Z}_m^*$ existiert mit $x^2 \equiv_m a$. Andernfalls heißt a **quadratischer Nichtrest** (kurz: $a \in \mathsf{QNR}_m$).

Sei $p > 2$ eine Primzahl und $\mathrm{ggT}(a, p) = 1$. Dann heißt

$$\mathcal{L}(a,p) = \left(\frac{a}{p}\right) = \begin{cases} 1, & a \in \mathsf{QR}_p \\ -1, & a \in \mathsf{QNR}_p \end{cases}$$

das **Legendre-Symbol** von a modulo p.

Die Kongruenzgleichung (7) besitzt also für ein $a \in \mathbb{Z}_m^*$ genau dann eine Lösung, wenn $a \in \mathsf{QR}_m$ ist. Wie das folgende Lemma zeigt, kann die Lösbarkeit von (7) für primes m effizient entschieden werden. Am Ende dieses Abschnitts werden wir noch eine andere Methode zur effizienten Berechnung des Legendre-Symbols kennenlernen.

Lemma 133 *lem:QRei $a \in \mathbb{Z}_p^*$, $p > 2$ prim, und sei $k = \log_{p,\,g}(a)$ für einen beliebigen Erzeuger g von $\mathbb{Z}_p^*$. Dann sind die folgenden drei Bedingungen äquivalent:*

a) $a^{(p-1)/2} \equiv_p 1$,

b) k *ist gerade,*

c) $a \in \mathsf{QR}_p$.

Beweis:
$a \Rightarrow b$: Angenommen, $a \equiv_p g^k$ für ein ungerades $k = 2 \cdot j + 1$. Dann ist

$$a^{(p-1)/2} \equiv_p \underbrace{g^{j\cdot(p-1)}}_{=1}\underbrace{g^{(p-1)/2}}_{\neq 1} \equiv_p g^{(p-1)/2} \not\equiv_p 1.$$

$b \Rightarrow c$: Ist $a \equiv_p g^k$ für $k = 2j$ gerade, so folgt $a \equiv_p (g^j)^2$, also $a \in \mathsf{QR}_p$.

$c \Rightarrow a$: Sei $a \in \mathsf{QR}_p$, d. h. $b^2 \equiv_p a$ für ein $b \in \mathbb{Z}_p^*$. Dann folgt mit dem Satz von Fermat,

$$a^{(p-1)/2} \equiv_p b^{p-1} \equiv_p 1.$$

■

Satz 134 (Eulers Kriterium) *Für alle $a \in \mathbb{Z}_p^*$, $p > 2$ prim, gilt*

$$a^{(p-1)/2} \equiv_p \left(\frac{a}{p}\right).$$

Beweis: Nach obigem Lemma reicht es zu zeigen, dass $a^{(p-1)/2} \equiv_p \pm 1$ sein muss. Da jedoch die Kongruenz $x^2 \equiv_p 1$ nach dem Satz von Lagrange (Satz 124) nur die beiden Lösungen 1 und -1 hat, folgt dies aus der Tatsache, dass $a^{(p-1)/2}$ Lösung dieser Kongruenz ist. ■

Korollar 135 *Für alle $a, b \in \mathbb{Z}_p^*$, $p > 2$ prim, gilt*

a) $\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2} = \begin{cases} 1, & p \equiv_4 1, \\ -1, & p \equiv_4 3, \end{cases}$

b) $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{b}{p}\right)$.

Als weiteres Korollar aus Eulers Kriterium erhalten wir eine Methode, quadratische Kongruenzgleichungen im Fall $p \equiv_4 3$ zu lösen. Für beliebige Primzahlen p ist kein effizienter, deterministischer Algorithmus bekannt. Es gibt jedoch einen probabilistischen Algorithmus von Adleman, Manders und Miller (1977).

Korollar 136 *Sei $p > 2$ prim, dann besitzt die quadratische Kongruenzgleichung $x^2 \equiv_p a$ für jedes $a \in \mathsf{QR}_p$ genau zwei Lösungen. Im Fall $p \equiv_4 3$ sind dies $\pm a^k$ (für $k = (p+1)/4$) von denen genau eine ein quadratischer Rest ist.*

Beweis: Sei $a \in \mathsf{QR}_p$, d. h. es existiert ein $b \in \mathbb{Z}_p^*$ mit $b^2 \equiv_p a$. Mit b ist auch $-b$ eine Lösung von $x^2 \equiv_p a$, die von b verschieden ist (p ist ungerade). Nach Lagrange (Satz 124) existieren keine weitere Lösungen.

Sei nun $p \equiv_4 3$. Dann gilt

$$\left(\frac{-b}{p}\right) = \left(\frac{-1}{p}\right) \cdot \left(\frac{b}{p}\right) = -\left(\frac{b}{p}\right)$$

nach Korollar 135. Demnach ist genau eine der beiden Lösungen $\pm b$ ein quadratischer Rest. Schließlich liefert Eulers Kriterium für $k = (p+1)/4$

$$a^{2k-1} = a^{(p-1)/2} \equiv_p 1$$

Also folgt $(a^k)^2 \equiv_p a$. ■

Zum Schluss dieses Abschnitts erweitern wir das Legendre-Symbol zum Jacobi-Symbol und zeigen, wie auch dieses effizient berechnet werden kann.

Definition 137 (Jacobi-Symbol)

Das **Jacobi-Symbol** ist für alle ungeraden $m > 3$ und $a \in \mathbb{Z}_m^*$ durch

$$\mathcal{J}(a, m) = \left(\frac{a}{m}\right) = \left(\frac{a}{p_1}\right)^{e_1} \cdots \left(\frac{a}{p_r}\right)^{e_r}$$

definiert, wobei $p_1^{e_1} \cdots p_r^{e_r}$ die Primfaktorzerlegung von m ist.

Man beachte, dass im Gegensatz zum Legendre-Symbol die Eigenschaft $\left(\frac{a}{m}\right) = 1$ für ein $a \in \mathbb{Z}_m^*$ nicht unbedingt mit $a \in \mathsf{QR}_m$ gleichbedeutend ist. Interessanterweise ist das Jacobi-Symbol auch ohne Kenntnis der Primfaktorzerlegung des Moduls effizient berechenbar. Der Algorithmus basiert auf den folgenden beiden Sätzen, die wir ohne Beweis angeben.

Satz 138 (Quadratisches Reziprozitätsgesetz, Gauß 1796) *Es seien $m, n > 2$, ungerade und teilerfremd. Dann gilt*

$$\left(\frac{n}{m}\right) = (-1)^{(m-1)\cdot(n-1)/4}\left(\frac{m}{n}\right) = \begin{cases} -\left(\frac{m}{n}\right), & m \equiv_4 n \equiv_4 3, \\ \left(\frac{m}{n}\right), & \textit{sonst} \end{cases}$$

Satz 139 *Für ungerades m gilt*

$$\left(\frac{2}{m}\right) = (-1)^{\frac{m^2-1}{8}} = \begin{cases} 1, & m \equiv_8 \pm 11, \\ -1, & m \equiv_8 \pm 3. \end{cases}$$

Korollar 140 *Das Jacobi-Symbol ist effizient berechenbar.*

Beweis: Dies folgt, ähnlich wie beim Euklidschen Algorithmus, aus folgenden Gleichungen.

$$\left(\frac{a}{m}\right) = \begin{cases} 1, & \text{falls } a = 1 \\ \left(\frac{m \bmod a}{a}\right) \cdot (-1)^{(a-1)(m-1)/4}, & \text{falls } a \text{ ungerade} \\ \left(\frac{b}{m}\right), & \text{falls } a = 2^{2k} \cdot b,\ b \text{ ungerade} \\ \left(\frac{b}{m}\right) \cdot (-1)^{(m^2-1)/8}, & \text{falls } a = 2^{2k+1} \cdot b,\ b \text{ ungerade.} \end{cases}$$

■

Beispiel 141 *Das Jacobi-Symbol zu 73 modulo 83:*

$$\mathcal{J}(73, 83) = \left(\frac{73}{83}\right) = \left(\frac{83}{73}\right) = \left(\frac{2}{73}\right) \cdot \left(\frac{5}{75}\right) = \left(\frac{73}{5}\right) = \left(\frac{5}{3}\right) = -1. \qquad \triangleleft$$

6.4 Primzahlen

Sei $\pi : \mathbb{N} \to \mathbb{N}_0$ mit

$$\pi(n) = \|\{2 \leq p \leq n \mid p \in \mathbb{P}\}\|$$

die Anzahl der Primzahlen kleiner gleich n. Mit $\pi_{a,\,m}(n)$ bezeichnen wir die Anzahl der Primzahlen kleiner gleich n, die von der Form $p = m \cdot k + a$ für ein $k \in \mathbb{N}$ sind.

Satz 142 (Primzahlsatz, Hadamard, de la Vallée Poussin 1896)
Ist $\mathrm{ggT}(a, m) = 1$*, so gilt*[*]

$$\pi_{a,\,m}(n) \sim \frac{n}{\varphi(m) \cdot \ln n}$$

Insbesondere gilt also

$$\pi(n) \sim \frac{n}{\ln n}.$$

Eine bessere Abschätzung liefert die Funktion $Li(n) = \int_2^n (\ln x)^{-1} dx$, wie folgende Tabelle zeigt.

n	$\pi(n)$	$\pi(n) - n/\ln n$	$Li(n) - \pi(n)$
10	4	−0.3	2.2
100	25	3.3	5.1
1 000	168	23	10
10 000	1 229	143	17
10 100	1 240	144	18
10^6	78 498	6 116	130
10^9	50 847 534	2 592 592	1 701
10^{12}	37 607 912 018	1 416 705 193	38 263
10^{15}	29 844 570 422 669	891 604 962 452	1 052 619
10^{18}	24 739 954 287 740 860	612 483 070 893 536	21 949 555
10^{21}	21 127 269 486 018 731 928	446 579 871 578 168 707	597 394 254

Beispiel 143 *Die Anzahl der Primzahlen in einem Intervall* $[n, m]$ *ist demnach näherungsweise* $\frac{m}{\ln m} - \frac{n}{\ln n}$*. Für das Intervall* $[10\,000, 10\,100]$ *ergibt sich z. B. ein Näherungswert von* $9{,}674 \approx 10$*, während der tatsächliche Wert gleich 11 ist.* ◁

Beispiel 144 *Für die Anzahl* $\|\mathbb{P}_{100}\|$ *aller 100-stelligen Primzahlen (in Dezimaldarstellung) erhalten wir z. B. den Näherungswert*

$$\|\mathbb{P}_{100}\| \approx \frac{10^{100}}{100 \cdot \ln 10} - \frac{10^{99}}{99 \cdot \ln 10} \approx 3{,}91 \cdot 10^{97}.$$

Vergleicht man diese Zahl mit der Anzahl aller 100-stelligen Dezimalzahlen, so sehen wir, dass ungefähr jede 230-te 100-stellige Dezimalzahl prim ist. ◁

Der Beweis des Primzahlsatzes ist sehr aufwendig. Mit elementaren Mitteln lässt sich jedoch folgender Satz beweisen, der für die meisten Anwendungen vollkommen ausreicht.

Satz 145 (Tschebyscheff) *Für alle* $n > 200$ *gilt* $\pi(n) > \frac{2 \cdot n}{3 \cdot \ln n}$.
(Ohne Beweis)

[*] $f(n) \sim g(n)$ bedeutet $\lim_{n\to\infty} f(n)/g(n) = 1$.

Der Solovay-Strassen Test (SST)

Eulers Kriterium besagt, dass im Falle p prim für alle $a \in \mathbb{Z}_p^*$ die Kongruenz $a^{(p-1)/2} \equiv_p (\frac{a}{p})$ gilt. Der Solovay-Strassen-Test basiert auf der Tatsache, dass für zusammengesetztes ungerades n mindestens die Hälfte aller $a \in \mathbb{Z}_n^*$ diese Kongruenz nicht erfüllen. Für $n \geq 3$ sei

$$\mathcal{P}_n^{SST} = \{a \in \mathbb{Z}_n^* \mid \left(\tfrac{a}{n}\right) \equiv_n a^{(n-1)/2}\}.$$

Satz 146 *Eigenschaften von $\mathcal{P}_n^{SST}$:*

a) $n \in \mathbb{P} \quad \Rightarrow \quad \|\mathcal{P}_n^{SST}\| = \varphi(n)$,

b) n *zusammengesetzt und ungerade* $\quad \Rightarrow \quad \|\mathcal{P}_n^{SST}\| \leq \frac{\varphi(n)}{2}$.

Beweis: Teil a folgt unmittelbar aus Eulers Kriterium. Für Teil b sei n zusammengesetzt und ungerade. Es ist leicht zu sehen, dass die Menge $\mathcal{P}_n^{SST}$ eine Untergruppe von $\mathbb{Z}_n^*$ bildet. Da somit $\|\mathcal{P}_n^{SST}\|$ ein Teiler von $\varphi(n)$ ist, reicht es zu zeigen, dass mindestens ein Element $w \in \mathbb{Z}_n^* - \mathcal{P}_n^{SST}$ existiert.

1. Fall : n ist quadratfrei, d. h. $n = p_1 \cdots p_k$ für k verschiedene Primzahlen. Dann garantiert der Chinesische Restsatz die Existenz einer Zahl $a \in \mathbb{Z}_n^*$ mit $a \equiv_{p_1} s$ und $a \equiv_{p_i} 1$ für $i = 2, \ldots, k$, wobei s beliebig aus QNR_{p_1} gewählt ist. Nun gilt

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right) \cdot \left(\frac{a}{p_2}\right) \cdot \ldots \cdot \left(\frac{a}{p_k}\right) = \left(\frac{a}{p_1}\right) = -1,$$

aber $a^{(n-1)/2} \not\equiv_n -1$, da andernfalls $a^{(n-1)/2} \equiv_{p_2 \cdots p_k} -1$ folgen würde, was im Widerspruch zu $a^{(n-1)/2} \equiv_{p_2 \cdots p_k} 1$ steht.

2. Fall : n ist nicht quadratfrei, d. h. es gibt eine Primzahl p mit $p^2 | n$. Betrachte $a = 1 + {}^{n}\!/_{p} \in \mathbb{Z}_n^*$. Dann gilt

$$\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{q_i}\right)^{e_i}$$

wobei $\prod_{i=1}^{k} q_i^{e_i}$ die Primfaktorzerlegung von n ist. Wegen $a \equiv_{q_i} 1$ folgt $\left(\frac{a}{n}\right) = 1$. Andererseits muss aber $a^{(n-1)/2} \not\equiv_n 1$ gelten, da sonst

$$1 \equiv_n (1 + {}^{n}\!/_{p})^{(n-1)/2} = \sum_{i=0}^{(n-1)/2} \binom{{}^{(n-1)}\!/_{2}}{i} ({}^{n}\!/_{p})^i \equiv_n 1 + \tfrac{n-1}{2} \cdot \tfrac{n}{p},$$

da $({}^{n}\!/_{p})^i \equiv_n 0$ für $i \geq 2$. Also folgt $\frac{n-1}{2} \cdot \frac{n}{p} \equiv_n 0$, und damit wäre p ein Teiler von $(n-1)/2$, was ein Widerspruch ist. ■

Obiger Satz führt unmittelbar auf folgenden effizienten probabilistischen Primzahltest, der nach Solovay und Strassen benannt ist:

Algorithmus 147 *SST*(n,k), n ungerade und $k \geq 1$

```
1  while k ≥ 1 do
2    k ← k − 1
3    wähle zufällig ein a ∈ {2, ..., n − 1}
4    if ( ggT(a, n) ≠ 1 oder a ∉ P_n^SST ) then
5      return „zusammengesetzt“
6    end
7  end
8  return „prim“
```

Korollar 148 *Der Solovay-Strassen-Test läuft in Polynomialzeit und für ungerade n gilt,*

$$n \text{ ist eine Primzahl } \Rightarrow Prob[SST(n,k) = \text{„prim“}] = 1,$$
$$n \text{ ist zusammengesetzt } \Rightarrow Prob[SST(n,k) = \text{„zusammengesetzt“}] > 1 - 2^{-k}.$$

Ist n eine Primzahl, so gibt der Solovay-Strassen-Test mit Wahrscheinlichkeit 1 „prim“ aus, andernfalls erzeugt er mit einer Wahrscheinlichkeit größer als $1 - 2^{-k}$ die Ausgabe „zusammengesetzt“. Da der Algorithmus (mit beliebig kleiner Wahrscheinlichkeit) eine falsche Ausgabe produzieren kann, handelt es sich um einen sogenannten **Monte-Carlo-Algorithmus** (mit einseitigem Fehler, da es nur im Fall n zusammengesetzt zu einer falschen Ausgabe kommen kann). Im Gegensatz hierzu gibt ein sogenannter **Las-Vegas-Algorithmus** nie eine falsche Antwort. Allerdings darf ein Las-Vegas-Algorithmus (mit kleiner Wahrscheinlichkeit) die Antwort verweigern, also ein „?“ ausgeben.

6.5 Pseudo-Primzahlen und der Fermat-Test (FT)

Wie wir im vorigen Abschnitt gesehen haben, geht man bei der Konstruktion eines probabilistischen Monte-Carlo Primzahltests üblicherweise so vor, dass man eine Folge von Teilmengen $\mathcal{P}_n \subseteq \mathbb{Z}_n^*$ wählt, die die folgenden drei Eigenschaften für alle $n \geq n_0$ erfüllen:

a) Für ein gegebenes $a \in \mathbb{Z}_n^*$ kann effizient, d. h. in Polynomialzeit getestet werden, ob $a \in \mathcal{P}_n$ ist.

b) Für $n \in \mathbb{P}$ ist $\mathcal{P}_n = \mathbb{Z}_n^*$.

c) Für zusammengesetztes n ist ein konstanter Anteil aller Elemente von $\mathbb{Z}_n^*$ nicht in $\mathcal{P}_n$ enthalten, d. h. $\|\mathcal{P}_n\| \leq (1-\varepsilon)\varphi(n)$ für eine Konstante $\varepsilon > 0$.

Typischerweise wählt man für $\mathcal{P}_n$ daher eine Eigenschaft, die für alle Elemente $a \in \mathbb{Z}_n^*$ gilt, falls n prim ist.

Weist eine zusammengesetzte Zahl n für die Basiszahl $a \in \mathbb{Z}_n^*$ beim Solovay-Strassen Test das gleiche Verhalten wie eine Primzahl auf; gilt also $a \in \mathcal{P}_n^{SST}$, so spricht man von einer *Euler-Pseudo-Primzahl zur Basis* a.

Beispiel 149 *Sei* $n = 91$ *und* $a = 10$*:* $\quad 10^{45} \bmod 91 = 90 \equiv_{91} -1$

$$\left(\frac{10}{91}\right) = \left(\frac{2}{91}\right) \cdot \left(\frac{5}{91}\right) = -\left(\frac{91}{5}\right) = -1 \qquad \triangleleft$$

Es liegt nahe, neben Eulers Kriterium auch den Satz von Fermat zur Konstruktion einer „Testmengensequenz" $\mathcal{P}_n^{FT} = \{a \in \mathbb{Z}_n^* \mid a^{n-1} \equiv_n 1\}$ zu verwenden. Analog zu den Euler-Pseudo-Primzahlen nennt man eine zusammengesetzte Zahl n, die den resultierenden Primzahltest bei Wahl von $a \in \mathbb{Z}_n^*$ besteht, d. h. es gilt

$$\mathrm{ggT}(a, n) = 1 \qquad \text{und} \qquad a^{n-1} \equiv_n 1,$$

eine *Fermat-Pseudo-Primzahl* oder einfach *Pseudo-Primzahl zur Basis* a. Es ist leicht zu sehen, dass jede Euler-Pseudo-Primzahl zur Basis a auch pseudo-prim zur Basis a ist.

Die Umkehrung gilt jedoch nicht. Zum Beispiel ist die Zahl 91 zwar pseudo-prim aber nicht euler-pseudo-prim zur Basis 3. Es gibt sogar Zahlen n (z. B. $n = 561$) die pseudo-prim zu jeder Basis $a \in \mathbb{Z}_n^*$ sind (sogenannte *Carmichael-Zahlen*). Für diese Zahlen ist Bedingung c in obiger Aufzählung nicht erfüllt, so dass der Satz von Fermat nur auf einen so genannten Pseudo-Primzahltest führt. Wie im Beweis zu Satz 150 on the facing page lässt sich zeigen, dass Bedingung c in obiger Aufzählung für $\varepsilon = 1/2$ erfüllt ist, sofern man nur zusammengesetzte Zahlen betrachtet, die keine Carmichael-Zahl sind. Carmichael-Zahlen kommen allerdings nur sehr selten vor (erst 1992 konnte die Existenz unendlich vieler Carmichael-Zahlen nachgewiesen werden).

Der Fermat-Pseudoprimzahltest kann zu einem Monte-Carlo Primzahltest (dem sogenannten Miller-Rabin Test) wie folgt erweitert werden:

> Gilt für das zufällig gewählte $a \in \{2, \ldots, n-1\}$ sowohl $\mathrm{ggT}(a, n) = 1$ als auch $a^{n-1} \equiv_n 1$, so gib nicht sofort „prim" aus, sondern berechne der Reihe nach die Zahlen $b_i = a^{(n-1)/2^i} \bmod n$ (für $0 \leq i \leq m$) bis entweder $(n-1)/2^m$ ungerade, oder $b_m \neq 1$ ist. Gib nur „prim" aus, falls $b_m = -1$ ist.

Mit etwas zahlentheoretischem Aufwand kann gezeigt werden, dass der Miller-Rabin Test obige Bedingung c (sogar mit $\varepsilon = 3/4$) erfüllt. Die durch den Miller-Rabin Test definierten Pseudo-Primzahlen werden *starke Pseudo-Primzahlen zur Basis* a genannt. Starke Pseudo-Primzahlen zur Basis a sind immer auch Euler-Pseudo-Primzahlen zur Basis a. Ist $n \equiv_4 3$, so gilt hiervon auch die Umkehrung.

Es gibt nur eine Zahl $n < 2,5 \cdot 10^{10}$, die stark pseudo-prim zu den Basen 2, 3, 5 und 7 ist: $n = 3\,215\,031\,751 = 151 \cdot 751 \cdot 28\,315$. Unter Verwendung der verallgemeinerten

Riemannschen Hypothese kann man sogar zeigen, dass es keine Zahl n gibt, die stark pseudo-prim zu allen Basen a mit $a < 2 \cdot (\ln n)^2$ ist. Unter dieser Hypothese kann der Miller-Rabin Test daher zu einem deterministischen Polynomialzeit-Algorithmus derandomisiert werden (mit der Folge, dass das Primzahlproblem in P lösbar ist). Erst 2002 fanden Agrawal, Kayal und Saxena einen Algorithmus, der das Primzahlproblem auch ohne diese Voraussetzung in P löst.

Der Miller-Rabin Test (MRT)

Für $n \geq 3$ sei $n - 1 = 2^m u$, u ungerade, und

$$\begin{aligned}\mathcal{P}_n^{MRT} &= \{a \in \mathbb{Z}_n^* \mid a^{n-1} \equiv_n 1 \text{ und } \forall i = 1, \ldots, m : a^{2^i u} \equiv_n 1 \rightarrow a^{2^{i-1} u} \equiv_n \pm 1\} \\ &= \{a \in \mathbb{Z}_n^* \mid a^u \equiv_n 1 \text{ oder } \exists i < m : a^{2^i u} \equiv_n -1\}.\end{aligned}$$

Satz 150 *Eigenschaften von $\mathcal{P}_n^{MRT}$:*

a) $n \in \mathbb{P} \quad \Rightarrow \quad \|\mathcal{P}_n^{MRT}\| = \varphi(n)$,

b) n *zusammengesetzt und ungerade* $\quad \Rightarrow \quad \|\mathcal{P}_n^{MRT}\| \leq \varphi(n)/4$.

(Beweis siehe Übungen.)

Algorithmus 151 *MRT*(n, k), n ungerade und $k \geq 1$

```
1  sei ∑_{i=0}^{r} e_i · 2^i, e_r = 1, die Binärdarstellung von n − 1
2  for j ← 1 to k do
3    wähle zufällig ein a ∈ {1, ..., n − 1}
4    b ← a
5    for i ← r − 1 downto 0 do
6      c ← b
7      b ← b^2 mod m
8      if (b ≡_n 1 ∧ c ≢_n ±1) then return „zusammengesetzt“
9      if (e_i = 1) then b ← b · a mod m
10   end
11   if (b ≢_n 1) then return „zusammengesetzt“
12 end
13 return „prim“
```

Korollar 152 *Der Miller-Rabin-Test läuft in Polynomialzeit und für ungerade n gilt,*

$$\begin{aligned}n \text{ ist eine Primzahl} &\Rightarrow Prob[\textit{MRT}(n,k) = \text{„prim“}] = 1, \\ n \text{ ist zusammengesetzt} &\Rightarrow Prob[\textit{MRT}(n,k) = \text{„zusammengesetzt“}] > 1 - 4^{-k}.\end{aligned}$$