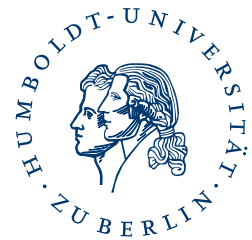
Einführung in die Kryptologie

Johannes Köbler



Institut für Informatik
Humboldt-Universität zu Berlin

SS 2020

- Für das Verständnis der Public-Key Verfahren benötigen wir noch einige Hilfsmittel aus der Zahlentheorie
- Nehmen wir ein beliebiges Element a einer endlichen Gruppe G und betrachten die Folge der Potenzen $a^0 = 1$, $a^1 = a$, a^2, a^3, ..., so stellt sich die Frage, ob es einen Exponenten $n \geq 1$ mit $a^n = 1$ gibt
- In den Übungen werden wir sehen, dass das so ist und bis dahin alle Potenzen paarweise verschieden sind

Definition. Sei G eine endliche Gruppe.

- Die ***Ordnung von*** G ist die Anzahl $\|G\|$ ihrer Elemente
- Die ***Ordnung eines Elements*** $a \in G$ ist $\mathrm{ord}_G(a) = \min\{n \geq 1 \mid a^n = 1\}$
- Ist $G = \mathbb{Z}_m^*$, so schreiben wir einfach $\mathrm{ord}_m(a)$ anstelle von $\mathrm{ord}_{\mathbb{Z}_m^*}(a)$
- Die von a in G erzeugte Untergruppe $\{a^0, \ldots, a^{\mathrm{ord}_G(a)-1}\}$ bezeichnen wir mit $\langle a \rangle_G$ oder mit $\langle a \rangle$, wenn G aus dem Kontext ersichtlich ist

Die Ordnung von Gruppenelementen

- In den Übungen werden wir sehen, dass für beliebige ganze Zahlen $i, j \in \mathbb{Z}$ folgende Äquivalenz gilt

$$a^i = a^j \;\Leftrightarrow\; i \equiv_{\mathrm{ord}(a)} j$$

- Da $\mathrm{ord}(a) = \|\langle a \rangle\|$ die Ordnung einer Untergruppe von G ist, muss $\mathrm{ord}(a)$ ein Teiler der Gruppenordnung $\|G\|$ sein (Satz von Lagrange)

Beispiel

a	1	2	3	4	5	6
$\langle a \rangle$	$\{1\}$	$\{1,2,4\}$	$\{1,3,2,6,4,5\}$	$\{1,4,2\}$	$\{1,5,4,6,2,3\}$	$\{1,6\}$
$\mathrm{ord}_G(a)$	1	3	6	3	6	2

- Die Tabelle zeigt für jedes Element a der Gruppe $G = \mathbb{Z}_7^*$ die von a erzeugte Untergruppe $\langle a \rangle$ sowie dessen Ordnung $\mathrm{ord}_G(a) = \|\langle a \rangle\|$ ◁

Die Ordnung von Gruppenelementen

Satz (Euler-Fermat)

In jeder Gruppe $(G, \cdot, 1)$ der Ordnung $\|G\| = m$ gilt $a^m = 1$ für alle $a \in G$

Beweis.

- Wir betrachten hier nur den Fall, dass G kommutativ ist, der allgemeine Fall wird in den Übungen bewiesen
- Sei also $G = \{b_1, \ldots, b_m\}$ abelsch und sei $a \in G$ beliebig
- Wegen $ab_i \neq ab_j$ für $i \neq j$ folgt $G = \{ab_1, \ldots, ab_m\}$
- Dies impliziert $\prod_{i=1}^{m} b_i = \prod_{i=1}^{m} ab_i = a^m \prod_{i=1}^{m} b_i$
- Also muss $a^m = 1$ sein □

Korollar (Kleiner Satz von Fermat)

Für jede Primzahl p und jede Zahl a mit $a \not\equiv_p 0$ gilt $a^{p-1} \equiv_p 1$

Diskrete Logarithmen

- Für ein beliebiges Gruppenelement $a \in G$ ist die Exponentiation $\exp_{G,a} : x \mapsto a^x$ zur Basis a eine Bijektion zwischen der Menge $\mathbb{Z}_{ord(a)} = \{0, 1, \ldots, ord(a) - 1\}$ und der Untergruppe $\langle a \rangle$
- Die zugehörige Umkehrabbildung spielt in der Kryptografie eine wichtige Rolle

Definition

- Seien $a, b \in G$ mit $b \in \langle a \rangle$
- Dann heißt der eindeutig bestimmte Exponent $x \in \mathbb{Z}_{\text{ord}(a)}$ mit $a^x = b$ ***Index*** oder ***diskreter Logarithmus von b zur Basis a in G***, kurz

$$x = \log_{G,a}(b)$$

- Im Fall $G = \mathbb{Z}_m^*$ schreiben wir auch einfach $\log_{m,a}(b)$ anstelle von $\log_{\mathbb{Z}_m^*,a}(b)$

Diskrete Logarithmen

- Die Funktion $\exp_{m,a} : x \mapsto a^x$ ist effizient berechenbar (siehe unten)
- Dagegen sind bis heute keine effizienten Verfahren zur Berechnung von $\log_{m,a}(b)$ bekannt (falls a und m geeignet gewählt werden)

Beispiel

- Das Element $a = 2$ hat in der Gruppe $G = \mathbb{Z}_{11}^*$ die maximal mögliche Ordnung $\mathrm{ord}_{11}(2) = \|G\| = 10$
- Die folgenden Tabellen zeigen den Werteverlauf der Funktionen $\exp_{11,2}$ und $\log_{11,2}$

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

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

◁

Zyklische Gruppen

Für manche Anwendungen sind Elemente $a \in G$ nützlich, mit denen sich die gesamte Gruppe erzeugen lässt

Definition

- Sei G eine endliche Gruppe der Ordnung $\|G\| = m$
- Ein Element $g \in G$ mit $\mathrm{ord}_G(g) = m$ heißt ***Erzeuger*** von G
- G heißt ***zyklisch***, falls G mindestens einen Erzeuger besitzt

Ein Element $a \in G$ ist also genau dann ein Erzeuger, wenn die von a erzeugte Untergruppe $\langle a \rangle$ die gesamte Gruppe G umfasst

Satz (Gauß)

Genau für $m \in \{1, 2, 4, p^k, 2p^k \mid 2 < p \text{ prim}\}$ ist die Gruppe $\mathbb{Z}_m^*$ zyklisch (ohne Beweis)

Zyklische Gruppen

Lemma (Euler)

Für alle $m \geq 1$ gilt

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

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

Beweis.

- Für jeden Teiler $d \geq 1$ von m sei $T_d := \{a \in \mathbb{Z}_m \mid \mathrm{ggT}(a, m) = d\}$
- Dann folgen wegen

$$\varphi(m/d) = \|\{b \in \mathbb{Z}_{m/d} \mid \underbrace{\mathrm{ggT}(b, m/d) = 1}_{\Leftrightarrow\, \mathrm{ggT}(bd, m) = d}\}\| = \|T_d\|$$

die Gleichungen

$$\sum_{d|m} \varphi(d) = \sum_{d|m} \varphi(m/d) = \sum_{d|m} \|T_d\| = \|\mathbb{Z}_m\| = m$$

□

Zyklische Gruppen

Satz

Eine Gruppe G der Ordnung $\|G\| = m$ ist genau dann zyklisch, wenn jede Gleichung der Form $x^n = 1$ $(1 \leq n \leq m)$ höchstens n verschiedene Lösungen $a \in G$ hat. In diesem Fall hat G genau $\varphi(m)$ Erzeuger

Beweis ($\Rightarrow$).

- Falls G zyklisch und a ein Erzeuger von G ist, so ist $G = \{a^k \mid k \in \mathbb{Z}_m\}$
- Die Potenz a^k ist genau dann Lösung von $x^n = 1$, wenn $a^{kn} = 1$ ist
- Sei $g = \mathrm{ggT}(n, m)$ und seien $n' = n/g$ sowie $m' = m/g$
- Da $\mathrm{ggT}(n', m') = 1$ ist, existiert ein Inverses $(n')^{-1}$ von n' modulo m'
- Wegen

$$a^{kn} = 1 \Leftrightarrow kn \equiv_m 0 \Leftrightarrow kn' \equiv_{m'} 0 \Leftrightarrow kn'(n')^{-1} \equiv_{m'} 0 \Leftrightarrow k \equiv_{m'} 0$$

hat also $x^n = 1$ nur die $g \leq n$ Lösungen $a^k = a^{jm'}$, $j = 0, 1, \ldots, g-1$

Zyklische Gruppen

Satz

Eine Gruppe G der Ordnung $\|G\| = m$ ist genau dann zyklisch, wenn jede Gleichung der Form $x^n = 1$ $(1 \leq n \leq m)$ höchstens n verschiedene Lösungen $a \in G$ hat. In diesem Fall hat G genau $\varphi(m)$ Erzeuger

Beweis ($\Leftarrow$).

- Für die Rückrichtung betrachten wir für jeden Teiler d von m die Menge $S_d = \{a \in G \mid \text{ord}(a) = d\}$ aller Elemente der Ordnung d in G
- Es ist klar, dass jedes Element $a \in S_d$ eine Lösung von $x^d = 1$ ist, d.h. es gilt $S_d \subseteq \{x \in G \mid x^d = 1\}$
- Also enthält S_d nach Voraussetzung nicht mehr als d Elemente
- Wir zeigen, dass die Größe von S_d sogar durch $\varphi(d)$ beschränkt ist
- Da jedes Element $a \in S_d$ eine Untergruppe $\langle a \rangle$ der Größe d erzeugt, folgt nach Euler-Fermat die Inklusion $\langle a \rangle \subseteq \{x \in G \mid x^d = 1\}$

Zyklische Gruppen

Beweis (Fortsetzung).

- Da $x^d = 1$ nicht mehr als d Lösungen hat, sind die beiden Mengen $\langle a \rangle$ und $\{x \in G \mid x^d = 1\}$ sogar gleich und es folgt $S_d \subseteq \langle a \rangle$ für alle $a \in S_d$
- Zudem gilt $\operatorname{ord}(a^i) = d$ genau dann, wenn $\operatorname{ggT}(i, d) = 1$ ist (siehe Üb.)
- Daher folgt $S_d = \{a^i \mid i \in \mathbb{Z}_d^*\}$ für jedes $a \in S_d$ und somit $\|S_d\| \leq \varphi(d)$
- Da die Mengen S_d die Gruppe G partitionieren, folgt $\sum_{d|m} \|S_d\| = m$
- Da nach obigem Lemma auch $\sum_{d|m} \varphi(d) = m$ ist, folgt

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

- Da aber S_d für jedes d entweder die Größe 0 oder $\varphi(d)$ hat, muss jedes S_d die Größe $\varphi(d)$ und insbesondere S_m die Größe $\varphi(m)$ haben □

- In einem Körper hat die Gleichung $x^n = 1$ höchstens n verschiedene Lösungen (siehe Übungen)
- Daher ist die Einheitengruppe $\mathbb{F}_q^*$ jedes endlichen Körpers $\mathbb{F}_q$ zyklisch und hat genau $\varphi(q-1)$ Erzeuger (insbesondere ist auch $\mathbb{Z}_p^*$ zyklisch)

Algorithmische Verifikation von Erzeugern

Sofern die Primfaktorzerlegung der Gruppenordnung m bekannt ist, lässt sich effizient überprüfen, ob ein Gruppenelement $a \in G$ ein Erzeuger ist

Satz

Ein Element a einer endlichen Gruppe G der Ordnung $\|G\| = m$ ist genau dann ein Erzeuger, wenn für jeden Primteiler p von m gilt:

$$a^{m/p} \neq 1$$

Beweis.

- Für einen Erzeuger a von G gilt $a^e \neq 1$ für alle $e \in \{1, \ldots, m-1\}$ und somit auch für alle Exponenten e der Form m/p, p prim
- Ist dagegen $\mathrm{ord}(a) < m$, ist $\mathrm{ord}(a)$ ein echter Teiler von m und daher existiert eine Zahl $d \geq 2$ mit $d \cdot \mathrm{ord}(a) = m$
- Folglich gilt für einen beliebigen Primteiler p von d

$$a^{m/p} = a^{d\,\mathrm{ord}(a)/p} = (a^{\mathrm{ord}(a)})^{d/p} = 1$$

□

Berechnung von Erzeugern

ComputeGenerator($G, p_1, \ldots, p_k$)

1 **input** zyklische Gruppe G und alle Primteiler $p_1, \ldots, p_k$ von $m = \|G\|$
2 **repeat**
3 **guess randomly** $a \in G$
4 **until** $a^{m/p_i} \neq 1$ für alle $i = 1, \ldots, k$
5 **output** a

- Der obige probabilistische Algorithmus berechnet einen Erzeuger a in einer zyklischen Gruppe G, falls sich die Elemente von G zufällig generieren lassen und alle Primteiler p von $m = \|G\|$ bekannt sind
- Da $\varphi(m) \geq m/(2 \ln \ln m)$ für hinreichend große m gilt, findet der Algorithmus in jedem Schleifendurchlauf mit Wahrscheinlichkeit $\varphi(m)/m \geq 1/(2 \ln \ln m)$ einen Erzeuger
- Die erwartete Anzahl der Schleifendurchläufe ist also $O(\ln \ln m)$

Wiederholtes Quadrieren und Multiplizieren

Pot(a, e)

1 $x := a$; $y := a^{e_0}$
2 **for** $i := 1$ **to** r **do**
3 $\quad x := x^2$; $y := y \cdot x^{e_i}$
4 **return**(y)

HornerPot(a, e)

1 $z := a$
2 **for** $i := r - 1$ **downto** 0 **do**
3 $\quad z := z^2 \cdot a^{e_i}$
4 **return**(z)

- Falls in einer Halbgruppe oder einem Ring das Produkt zweier Elemente effizient berechenbar ist, sind auch Potenzen a^e effizient berechenbar
- Hierzu sind maximal $2\lceil \log e \rceil$ Multiplikationen erforderlich
- Sei $e = \sum_{i=0}^{r} e_i \cdot 2^i$ mit $r = \lfloor \log_2 e \rfloor$ die Binärdarstellung von e
- Dann können wir den Exponenten e sukzessive mittels $b_0 = e_0$ und $b_i = b_{i-.55} + e_i 2^i = \sum_{j=0}^{i} e_j \cdot 2^j$ für $i = 1, \ldots, r$ zu $b_r = e$ berechnen
- Der Algorithmus Pot berechnet so die Potenzen a^{b_i} für $i = 0, \ldots, r$
- Alternativ lässt sich e auch nach dem Horner-Schema berechnen
  - Sei $c_r = e_r = 1$ und sei $c_{i-1} = 2c_i + e_{i-1}$ für $i = r, \ldots, 1$
  - Dann ist $c_i = \sum_{j=i}^{r} e_j \cdot 2^{j-i}$, also $c_0 = \sum_{j=0}^{r} e_j \cdot 2^j = e$
- So geht der Alg. HornerPot vor, der a^{c_i} für $i = r, \ldots, 0$ berechnet

Wiederholtes Quadrieren und Multiplizieren

Beispiel

Wir berechnen die Potenz a^e für $a = 1920$ und $e = 19$ im Ring $\mathbb{Z}_{2773}$:

			Pot(1920, 19)					HornerPot(1920, 19)	
i	e_i	b_i	$x_i = x_{i-1}^2 = a^{2^i}$	$y_i = y_{i-1} x_i^{e_i} = a^{b_i}$	i	e_i	c_i	$z_i = (z_{i+1})^2 a^{e_i} = a^{c_i}$
0	1	1	1920	$1920^1 = 1920$	4	1	1	$1920^1 = 1920$
1	1	3	$1920^2 = 1083$	$1920 \cdot 1083^1 = 2383$	3	0	2	$1920^2 \cdot 1920^0 = 1083$
2	0	3	$1083^2 = 2683$	$2383 \cdot 2683^0 = 2383$	2	0	4	$1083^2 \cdot 1920^0 = 2683$
3	0	3	$2683^2 = 2554$	$2383 \cdot 2554^0 = 2383$	1	1	9	$2683^2 \cdot 1920^1 = 1016$
4	1	19	$2554^2 = 820$	$2383 \cdot 820^1 = \mathbf{1868}$	0	1	19	$1016^2 \cdot 1920^1 = \mathbf{1868}$

◁

Der Primzahlsatz

- Bezeichne $\mathcal{P}$ die Menge aller Primzahlen und sei π die Funktion, die jeder Teilmenge $A \subseteq \mathbb{N}$ die Anzahl $\pi(A) = \|A \cap \mathcal{P}\|$ der Primzahlen in der Menge A zuweist
- Zudem bezeichnen wir die Zahl $\pi([1, n])$ auch einfach mit $\pi(n)$ und für $c \in \mathbb{Z}_m$ sei $\pi_{c,m}(n) := \pi(\{p \in \mathcal{P} \mid p \equiv_m c\})$

Satz (Hadamard und de la Vallée Poussin, 1896)

Es gilt

$$\pi(n) \sim n/\ln n$$

und für $c \in \mathbb{Z}_m^*$ gilt

$$\pi_{c,m}(n) \sim n/(\varphi(m) \ln n)$$

Hierbei bedeutet $f(n) \sim g(n)$, dass die beiden Funktionen f und g asymptotisch äquivalent sind (d.h. es gilt $\lim_{n\to\infty} f(n)/g(n) = 1$)

Der Primzahlsatz

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

- Wie obige Tabelle zeigt, liefert die Funktion $Li(n) = \int_2^n (\ln x)^{-1} dx$ im Vergleich zu $n/\ln n$ eine deutlich bessere Abschätzung von $\pi(n)$
- Verwenden wir die Abschätzung $\pi(n) \approx \int_2^n (\ln x)^{-1} dx$, so ergibt sich für die Anzahl $\pi([a,b])$ der Primzahlen im Intervall $[a,b]$ der Näherungswert

$$\pi([a,b]) \approx \int_a^b (\ln x)^{-1} dx \geq (b-a)/\ln b$$

Der Primzahlsatz

Beispiel

- Für das Intervall $[a,b] = [10\,000,10\,100]$ ergibt sich z. B. der Wert
$$\pi([a,b]) \approx \int_a^b (\ln x)^{-1} dx \geq 100/\ln 10\,100 \approx 10{,}85$$
während der exakte Wert $\pi(10\,100) - \pi(10\,000) = 11$ ist
- Für die Anzahl aller 100-stelligen Primzahlen (in Dezimaldarstellung) im Intervall $[a,b] = [10^{99}, 10^{100}]$ erhalten wir den Näherungswert
$$\int_a^b (\ln x)^{-1} dx \geq 9 * 10^{99}/\ln 10^{100} = 9 * 10^{97}/\ln 10 \approx 3{,}91 \cdot 10^{97}$$
- Vergleicht man diese Zahl mit der Anzahl $10^{100} - 10^{99} = 9 \cdot 10^{99}$ aller 100-stelligen Dezimalzahlen, so sehen wir, dass ungefähr jede $900/3{,}91 \approx 230$-te 100-stellige Dezimalzahl prim ist
- Für die Anzahl aller 1000-stelligen Primzahlen im Intervall $[a,b] = [10^{999}, 10^{1000}]$ erhalten wir dagegen den Näherungswert
$$\int_a^b (\ln x)^{-1} dx \geq 9 * 10^{999}/\ln 10^{1000} = 9 * 10^{996}/\ln 10 \approx 3{,}91 \cdot 10^{996}$$
- Hier sehen wir, dass ungefähr jede $9000/3{,}91 \approx 2303$-te der $9 \cdot 10^{999}$ 1000-stelligen Dezimalzahlen prim ist ◁

Pseudo-Primzahlen und der Fermat-Test

- Bei der Konstruktion eines probabilistischen Monte-Carlo Algorithmus geht man üblicherweise so vor, dass man eine Folge von Teilmengen $\mathcal{A}_n \subseteq \mathbb{Z}_n^*$ definiert, die für hinreichend großes n (also für $n \geq n_0$) folgende drei Bedingungen erfüllen:

  T1: Für gegebene Zahlen $a, n \in \mathbb{N}$ kann effizient, d. h. in Polynomialzeit getestet werden, ob $a \in \mathcal{A}_n$ ist

  T2: Für primes n ist $\mathcal{A}_n = \mathbb{Z}_n^*$

  T3: Für (ungerades) zusammengesetztes n ist ein konstanter Anteil von $\mathbb{Z}_n^*$ nicht in $\mathcal{A}_n$ enthalten, d. h. für ein $\varepsilon > 0$ gilt

  $$\|\mathcal{A}_n\| \leq (1 - \varepsilon)\varphi(n)$$

- Typischerweise wählt man für $\mathcal{A}_n$ also eine Eigenschaft, die alle $a \in \mathbb{Z}_n^*$ erfüllen, wenn n prim ist

Pseudo-Primzahlen und der Fermat-Test

- Der zugehörige generische Primzahltest GT arbeitet dann wie folgt:

$GT(n,k),\ k \geq 1$

```
1  for j := 1 to k do
2    guess randomly a ∈ {1, ..., n − 1}
3    if a ∉ A_n then return(zusammengesetzt)
4  return(prim)
```

- Hierbei steuert der Parameter k die maximale Fehlerwahrscheinlichkeit von $GT(n,k)$
- Gilt nämlich $\|\mathcal{A}_n\| \leq (1-\varepsilon)\varphi(n)$ für zusammengesetztes n und eine Konstante $\varepsilon > 0$, so gibt $GT(n,k)$ für zusammengesetztes n mit Wk

$$p = (a_n/(n-1))^k < (a_n/\varphi(n))^k \leq (1-\varepsilon)^k$$

„prim“ aus, wobei $a_n = \|\mathcal{A}_n\|$ ist

- Für primes n gibt $GT(n,k)$ dagegen immer „prim“ aus

Pseudo-Primzahlen und der Fermat-Test

- Da der Algorithmus falsche Ergebnisse liefern kann, handelt es sich um einen ***Monte-Carlo-Algorithmus*** (mit einseitigem Fehler, da nur im Fall n zusammengesetzt falsche Ergebnisse möglich sind)
- Im Gegensatz hierzu gibt ein ***Las-Vegas-Algorithmus*** nie eine falsche Antwort
- Allerdings darf ein Las-Vegas-Algorithmus (mit kleiner Wahrscheinlichkeit) die Antwort schuldig bleiben, also „?“ ausgeben
- Es liegt nahe, den Satz von Fermat zur Konstruktion von „Testmengen“

  $$\mathcal{A}_n^{FT} = \{a \in \mathbb{Z}_n^* \mid a^{n-1} \equiv_n 1\}$$

  zu verwenden
- Dies führt auf folgenden Fermat-Test (FT)

Pseudo-Primzahlen und der Fermat-Test

$FT(n,k)$, $n \geq 3$ ungerade und $k \geq 1$

```
1  berechne die Binärdarstellung n − 1 = Σ_{i=0}^{r} e_i · 2^i mit e_r = 1
2  for j := 1 to k do
3    guess randomly a ∈ {1, ..., n − 1}
4    z := a
5    for i := r − 1 downto 0 do
6      z := z^2 mod n
7      if e_i = 1 then z := z · a mod n
8    if z ≢_n 1 then return(zusammengesetzt)
9  return(prim)
```

- Der Fermat-Test berechnet also die Potenz $z_0 = a^{n-1}$ genau wie HornerPot ausgehend von $z_r = a$ mittels $z_{i-1} = z_i^2 a^{e_i - 1} \bmod n$
- Er erkennt n als zusammengesetzt, falls $z_0 \neq 1$ ist

Pseudo-Primzahlen und der Fermat-Test

- Man nennt eine zusammengesetzte Zahl n, die den Fermat-Test bei Wahl von $a \in \mathbb{Z}_n^*$ besteht (d. h. es gilt $a^{n-1} \equiv_n 1$) eine ***Fermat-Pseudo-Primzahl*** oder einfach ***Pseudo-Primzahl zur Basis*** a
- Man sagt auch, a ist ein ***(falscher) Primzahlzeuge*** für n
- Zum Beispiel ist die Zahl 91 pseudo-prim zur Basis 3
- Es gibt sogar Zahlen (z. B. $n = 561$) die pseudo-prim zu jeder Basis $a \in \mathbb{Z}_n^*$ sind (sogenannte ***Carmichael-Zahlen***)
- Für diese Zahlen ist Bedingung T3 in obiger Aufzählung nicht erfüllt, weshalb der Fermat-Test als ***Pseudo-Primzahltest*** bezeichnet wird
- Es ist leicht zu sehen, dass Bedingung T3 für jede zusammengesetzte Zahl, die keine Carmichael-Zahl ist, mit $\varepsilon = 1/2$ erfüllt ist
- Carmichael-Zahlen kommen nur sehr selten vor (erst 1992 konnte die Existenz unendlich vieler Carmichael-Zahlen nachgewiesen werden)

Der Miller-Rabin Test

- Der Fermat-Test kann zu einem Monte-Carlo Primzahltest (dem ***Miller-Rabin Test***, kurz ***MRT***) erweitert werden
- Wie wir gesehen haben, berechnet der Fermat-Test $z_0 = a^{n-1}$ ausgehend von $z_r = a$ mittels $z_{i-1} = z_i^2 a^{e_i - 1} \bmod n$
- Er erkennt n als zusammengesetzt, falls $z_0 \neq 1$ ist
- Der Miller-Rabin Test überprüft nun zusätzlich bei jeder Quadrierung, ob $z_i^2 \equiv_n 1$ und $z_i \not\equiv_n \pm 1$ ist
- Ist dies der Fall, so muss n zusammengesetzt sein, da z_i eine nicht-triviale Lösung der Kongruenz $x^2 \equiv_n 1$ in $\mathbb{Z}_n^*$ ist
- Die MRT-Testmenge ist also

$$\mathcal{A}_n^{MRT} = \{a \in \mathcal{A}_n^{FT} \mid \forall i = r, \ldots, 1 : z_i^2 \equiv_n 1 \Rightarrow z_i \equiv_n \pm 1\}$$

- Es ist klar, dass diese Testmengen die Bedingungen T1 und T2 erfüllen
- Weiter unten werden wir zeigen, dass auch Bedingung T3 für $\varepsilon = 1/2$ erfüllt ist (dies gilt sogar für $\varepsilon = 3/4$)

Der Miller-Rabin Test

$MRT(n, k)$, $n \geq 3$ ungerade und $k \geq 1$

```
1  berechne die Binärdarstellung n − 1 = Σ_{i=0}^{r} e_i · 2^i mit e_r = 1
2  for j := 1 to k do
3     guess randomly a ∈ {1, ..., n − 1}
4     z := a
5     for i := r − 1 downto 0 do
6        y := z
7        z := z^2 mod n
8        if z ≡_n 1 ∧ y ≢_n ±1 then return(zusammengesetzt)
9        if e_i = 1 then z := z · a mod n
10    if z ≢_n 1 then return(zusammengesetzt)
11 return(prim)
```

Der Miller-Rabin Test

Beispiel

- Bei Eingabe $n = 221 = 13 \cdot 17$ berechnet der Miller-Rabin Test für $a = 137$, $a' = 18$ und $a'' = 174$ die folgenden Werte z_i, z'_i bzw. z''_i (die dünn gedruckten Werte werden nur vom Fermat-Test berechnet, da der Miller-Rabin Test vorher abbricht)

i	e_i	$z_i = (z_{i+1})^2 a^{e_i}$	z_i^2	$z'_i = (z'_{i+1})^2 (a')^{e_i}$	$(z'_i)^2$	$z''_i = (z''_{i+1})^2 (a'')^{e_i}$	$(z''_i)^2$
7	1	**137**	**205**	**18**	**103**	**174**	**220**
6	1	$\mathbf{205 \cdot 137 = 18}$	**103**	$\mathbf{103 \cdot 18 = 86}$	**103**	$\mathbf{220 \cdot 174 = 47}$	**220**
5	0	**103**	**1**	**103**	**1**	**220**	**1**
4	1	$1 \cdot 137 = 137$	205	$1 \cdot 18 = 18$	103	$\mathbf{1 \cdot 174 = 174}$	**220**
3	1	$205 \cdot 137 = 18$	103	$103 \cdot 18 = 86$	103	$\mathbf{220 \cdot 174 = 47}$	**220**
2	1	$103 \cdot 137 = 188$	205	$103 \cdot 18 = 86$	103	$\mathbf{220 \cdot 174 = 47}$	**220**
1	0	205	35	103	1	**220**	**1**
0	0	35		1		**1**	

- Bei Wahl von $a = 137$ erkennen also beide Tests die Zahl $n = 221$ als zusammengesetzt, bei Wahl von $a' = 18$ tut dies nur der Miller-Rabin Test und bei Wahl von $a'' = 174$ keiner von beiden ◁

Der Miller-Rabin Test

- Die Zahlen $a \in \mathcal{A}_n^{\mathrm{MRT}}$ werden ***starke Primzahlzeugen*** für n genannt
- Falls n zusammengesetzt ist, sagt man auch, n ist eine ***starke Pseudo-Primzahl zur Basis*** a
- Es gibt nur eine Zahl $n < 2,5 \cdot 10^{10}$, die stark pseudo-prim zu den Basen 2, 3, 5 und 7 ist: $n = 3\,215\,031\,751 = 151 \cdot 751 \cdot 28\,351$
- Wir zeigen nun, dass jede ungerade zusammengesetzte Zahl $n > 2$ höchstens $\varphi(n)/2$ starke Primzahlzeugen hat
- Sei $n - 1 = 2^m u$ mit u ungerade, d.h. m ist der kleinste Index i mit $e_i = 1$ bzw. $m - 1$ der größte Index i, so dass $e_0 = \cdots = e_i = 0$ ist
- Zudem wählen wir ℓ als den kleinsten Index $i \geq 0$, so dass ein $a \in \mathbb{Z}_n^*$ existiert mit $z_i \equiv_n a^{c_i} \equiv_n -1$, d.h. für alle $i < \ell$ gilt $z_i \not\equiv_n -1$
- Da z_m für $a = n - 1$ den Wert $z_m \equiv_n (-1)^u \equiv_n -1$ hat, ist $\ell \leq m$
- Sei nun $U_n = \{a \in \mathbb{Z}_n^* \mid a^{2^j u} \equiv_n \pm 1\}$, wobei $j = m - \ell$ ist

Der Miller-Rabin Test

i	e_i	c_i	$a^{c_i} \equiv_n z_i$	
r	1	1	a	
$\vdots$				
m	1	u	a^u	$\equiv_n -1$ ist möglich
$m-1$	0	$2u$	a^{2u}	
$\vdots$				
$\ell = m-j$	0	$2^j u$	$a^{2^j u}$	
$\ell - 1$	0	$2^{j+1} u$	$a^{2^{j+1} u}$	$\not\equiv_n -1$
$\vdots$				
0	0	$2^m u$	$a^{2^m u}$	

Behauptung. U_n ist eine Untergruppe von $\mathbb{Z}_n^*$

- Es genügt zu zeigen, dass U_n unter Multiplikation abgeschlossen ist
- Für $a, b \in U_n$ gilt

$$(ab)^{2^j u} = a^{2^j u} b^{2^j u} \equiv_n (\pm 1)(\pm 1) = \pm 1$$

□

Der Miller-Rabin Test

Behauptung. $\mathcal{A}_n^{\mathrm{MRT}} \subseteq U_n$

- Sei $a \in \mathcal{A}_n^{\mathrm{MRT}}$. Dann gilt für die zugehörige Folge $z_r, \ldots, z_0$:
  - $z_0 \equiv_n a^{n-1} \equiv_n a^{2^m u} \equiv_n 1$ und
  - für alle $i = r, \ldots, 1$ mit $z_i^2 \equiv_n 1$ gilt $z_i \equiv_n \pm 1$
- Wegen $z_{m-i} \equiv_n a^{2^i u}$ für $i = 0, \ldots, m$ folgt also

$$\forall i \in [m] : a^{2^i u} \equiv_n 1 \Rightarrow a^{2^{i-1} u} \equiv_n \pm 1 \qquad (*)$$

- Zudem folgt aus der Definition von $\ell\ (= m - j)$,

$$\forall i \in \{j+1, \ldots, m\} : a^{2^i u} \not\equiv_n -1 \qquad (**)$$

- Insgesamt erhalten wir also aus (*) und (**) die Implikationen

$$a^{2^m u} \equiv_n 1 \overset{(*,**)}{\Rightarrow} a^{2^{m-1} u} \equiv_n 1 \overset{(*,**)}{\Rightarrow} \ldots \overset{(*,**)}{\Rightarrow} a^{2^{j+1} u} \equiv_n 1 \overset{(*)}{\Rightarrow} a^{2^j u} \equiv_n \pm 1$$

und somit folgt $a \in U_n$ □

Der Miller-Rabin Test

Behauptung. Für ungerades zusammengesetztes n ist U_n eine echte Untergruppe von $\mathbb{Z}_n^*$

- Falls $n = p^k$ eine Primzahlpotenz mit $p > 2$ und $k \geq 2$ ist, gilt $(p^{k-1}+1)^{p^k-1} \not\equiv_{p^k} \pm 1$ (siehe Übungen) und somit $a = p^{k-1}+1 \notin U_n$
- Andernfalls können wir n in teilerfremde Faktoren $n = n_1 n_2$ mit $n_1, n_2 > 2$ zerlegen
- Zudem existiert nach Definition von j eine Zahl $b \in \mathbb{Z}_n^*$ mit $b^{2^j u} \equiv_n -1$
- Sei $a \in \mathbb{Z}_n^*$ die eindeutige Lösung von

$$x \equiv_{n_1} b,$$

$$x \equiv_{n_2} 1$$

- Dann ist a wegen
  - $a^{2^j u} \equiv_{n_1} b^{2^j u} \equiv_{n_1} -1 \Rightarrow a^{2^j u} \not\equiv_n 1$ und
  - $a^{2^j u} \equiv_{n_2} 1^{2^j u} = 1 \Rightarrow a^{2^j u} \not\equiv_n -1$

  nicht in U_n enthalten □

Der Miller-Rabin Test

- Da U_n als echte Untergruppe von $\mathbb{Z}_n^*$ höchstens halb so groß wie $\mathbb{Z}_n^*$ sein kann, folgt also für ungerades zusammengesetztes n,

$$\left\|\mathcal{A}_n^{\mathrm{MRT}}\right\| \leq \|U_n\| \leq \varphi(n)/2.$$

- Damit ist gezeigt, dass der Miller-Rabin Test die Bedingung T3 für $\varepsilon = 1/2$ erfüllt
- 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