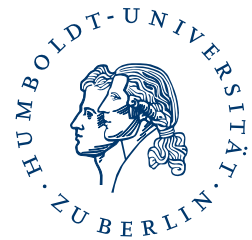
# Einführung in die Kryptologie

Johannes Köbler




Institut für Informatik
Humboldt-Universität zu Berlin

SS 2022

# Zahlentheoretische Grundlagen

Für das Verständnis der Public-Key Verfahren benötigen wir noch einige Hilfsmittel aus der Zahlentheorie

**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

- Sei $e \geq 1$ der kleinste Exponent mit $a^e = a^{e'}$ für ein $e' \in \{0, \ldots, e-1\}$
- Dann gilt $a^i \neq a^j$ für alle $0 \leq i < j < e$ und wegen $a^{e-e'} = a^e a^{-e'} = a^e a^{-e} = 1 = a^0$ muss $e - e' = e$, also $e' = 0$ sein
- Dies zeigt, dass $\mathrm{ord}_G(a) = e$ ist und die Menge $\{a^0, \ldots, a^{\mathrm{ord}_G(a)-1}\}$ tatsächlich eine Untergruppe von $G$ der Ordnung $\mathrm{ord}_G(a)$ bildet

# 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

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

| $a$ | 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 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 □

Angewandt auf die Gruppe $\mathbb{Z}_p^*$ erhalten wir den kleinen Satz von Fermat

**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 Exponentialfunktion $\exp_{G,a} : x \mapsto a^x$ mit der Basis $a$ eine Bijektion zwischen der Menge $\mathbb{Z}_{ord(a)} = \{0, 1, \ldots, ord(a) - 1\}$ und der von $a$ erzeugten Untergruppe $\langle a \rangle = \{a^0, \ldots, a^{\mathrm{ord}_G(a)-1}\}$
- Die zugehörige Umkehrabbildung spielt in der Kryptografie eine wichtige Rolle

**Definition.** Sei $G$ eine Gruppe und seien $a, b \in G$ mit $b \in \langle a \rangle$.

- Der eindeutig bestimmte Exponent $x \in \mathbb{Z}_{\mathrm{ord}(a)}$ mit $a^x = b$ heißt 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 $m \in \{1, 2, 4, p^n, 2p^n \mid p \geq 3 \text{ prim}, n \geq 1\}$ 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 folgt wegen

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

die Gleichheit

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

□

# Zyklische Gruppen

**Satz.** Sei $G$ eine Gruppe der Ordnung $|G| = m$.

- $G$ 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 $m' = m/g$
- Wegen $\mathrm{ggT}(n', m') = 1$ folgt dann mit dem Lemma von Euklid

$$a^{kn} = 1 \Leftrightarrow kn \equiv_m 0 \Leftrightarrow kn' \equiv_{m'} 0 \overset{Euklid}{\Leftrightarrow} k \equiv_{m'} 0$$

- Die Gleichung $x^n = 1$ hat also genau $g \leq n$ Lösungen $a^0, a^{m'}, \ldots, a^{(g-1)m'}$

# Zyklische Gruppen

**Satz.** Sei $G$ eine Gruppe der Ordnung $|G| = m$.

- $G$ 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 für $d \leq m$ die beiden Mengen
  - $Ord_d = \{a \in G \mid \text{ord}(a) = d\}$ und
  - $Sol_d = \{a \in G \mid a^d = 1\}$
- Wegen $Ord_d \subseteq Sol_d$ gilt nach Voraussetzung $|Ord_d| \leq |Sol_d| \leq d$
- Wir zeigen $|Sol_d| = d$ und $|Ord_d| = \varphi(d)$ für jeden Teiler $d$ von $m$
- Da für jedes $a \in Ord_d$ die Untergruppe $\langle a \rangle$ die Größe $d$ hat, folgt nach Euler-Fermat die Inklusion $\langle a \rangle \subseteq Sol_d$
- Wegen $d = |\langle a \rangle| \leq |Sol_d| \leq d$ folgt daher $Ord_d \subseteq Sol_d = \langle a \rangle$

# Zyklische Gruppen

## Beweis (Fortsetzung).

- Wegen $d = |\langle a \rangle| \leq |Sol_d| \leq d$ folgt daher $Ord_d \subseteq Sol_d = \langle a \rangle$
- Zudem gilt $\text{ord}(a^i) = d$ genau dann, wenn $\text{ggT}(i, d) = 1$ ist (siehe Üb.)
- Daher folgt $Ord_d = \{a^i \mid i \in \mathbb{Z}_d^*\}$ für jedes $a \in Ord_d$ und somit $|Ord_d| \in \{0, \varphi(d)\}$ für alle $d$
- Mit obigem Lemma folgt nun

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

- Wegen $|Ord_d| \in \{0, \varphi(d)\}$ muss also $|Ord_d| = \varphi(d)$ für alle Teiler $d$ von $m$ und insbesondere $|Ord_m| = \varphi(m)$ sein □

## Für endliche Körper $\mathbb{F}_q$ erhalten wir eine interessante Folgerung:

- Da die Gleichung $x^n = 1$ in jedem Körper höchstens $n$ Lösungen hat (siehe Übungen), ist die Einheitengruppe $\mathbb{F}_q^*$ von $\mathbb{F}_q$ zyklisch und hat genau $\varphi(q-1)$ Erzeuger (insbesondere ist auch $\mathbb{Z}_p^*$, $p$ prim, 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 $\mathrm{ord}(a) < m$, so 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

Der folgende 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

**Algorithmus** 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 $i \in \{1, \ldots, k\}$
5 **output** $a$

## Eigenschaften von ComputeGenerator:

- 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

- Falls in einer Halbgruppe oder einem Ring das Produkt zweier Elemente effizient berechenbar ist, sind auch die Potenzen $a^e$ effizient berechenbar
- Hierzu sind maximal $2\lceil \log e \rceil$ Multiplikationen erforderlich
- Sei $(e_r \dots e_0)_2$ mit $r = \lfloor \log_2 e \rfloor$ die Binärdarstellung von $e = \sum_{i=0}^{r} e_i \cdot 2^i$
- Dann können wir den Exponenten $e$ sukzessive mittels $b_0 = e_0$ und $b_i = b_{i-1} + e_i 2^i = \sum_{j=0}^{i} e_j \cdot 2^j$ für $i = 1, \dots, r$ zu $b_r = e$ berechnen
- Alternativ lässt sich $e$ auch nach dem Horner-Schema berechnen:
  - Sei $c_r = e_r = 1$ und für $i = r-1, \dots, 0$ sei $c_i = 2c_{i+1} + e_i$
  - Dann ist $c_i = \sum_{j=i}^{r} e_j \cdot 2^{j-i}$, also $c_0 = \sum_{j=0}^{r} e_j \cdot 2^j = e$
- Pot berechnet die Potenzen $y_i = a^{b_i}$ und HornerPot die Potenzen $z_i = a^{c_i}$:

**Pot$(a, e, G)$**

```
1  x := a; y := a^{e_0}
2  for i := 1 to r do
3     x := x^2; y := y · x^{e_i}
4  return(y)
```

**HornerPot$(a, e, G)$**

```
1  z := a
2  for i := r − 1 downto 0 do
3     z := z^2 · a^{e_i}
4  return(z)
```

# Wiederholtes Quadrieren und Multiplizieren

## Beispiel

- Wir berechnen in in der Gruppe $\mathbb{Z}_m^*$ die Potenz $a^e$ für $a = 1920$, $e = 19$ und $m = 2773$:

| Pot(1920, 19, 2773) | | | | | HornerPot(1920, 19, 2773) | | | |
|---|---|---|---|---|---|---|---|---|
| $i$ | $e_i$ | $b_i$ | $x_i = x_{i-1}^2 = a^{2^i}$ | $y_i = y_{i-1} x_i^{e_i} = a^{b_i}$ | $i$ | $e_i$ | $c_i$ | $z_i = (z_{i+1})^2 a^{e_i} = a^{c_i}$ |
| 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 | $1016^2 \cdot 1920^1 = \mathbf{1868}$ |

- Es gilt also $1920^{19} \equiv_{2773} 1868$

◁

# Der Primzahlsatz

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

**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

  $$\int_a^b (\ln x)^{-1} dx \geq 9 * 10^{99}/\ln 10^{100} = 9 * 10^{97}/\ln 10 \approx 3{,}91 \cdot 10^{97}$$
- Vergleichen wir 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 für das Primzahlproblem geht man üblicherweise so vor, dass man eine Folge von Testmengen $T_n \subseteq \mathbb{Z}_n^*$ definiert, die für hinreichend großes $n$ (also für alle $n \geq n_0$ für eine feste Zahl $n_0$) folgende drei Bedingungen erfüllen:
  - E: Für gegebene Zahlen $n \in \mathbb{N}$ und $a \in \mathbb{Z}_n$ kann effizient, d. h. in Polynomialzeit $|n|^{O(1)} = O(\log n)$ getestet werden, ob $a \in T_n$ ist
  - P: Für primes $n$ ist $T_n = \mathbb{Z}_n^*$
  - Z: Für (ungerades) zusammengesetztes $n \notin \mathcal{P}$ ist ein konstanter Anteil von $\mathbb{Z}_n^*$ nicht in $T_n$ enthalten, d.h. für ein $\varepsilon < 1$ gilt
    $$|T_n| \leq \varepsilon\varphi(n)$$
- Typischerweise wählt man für $T_n$ also eine Eigenschaft, die alle $a \in \mathbb{Z}_n^*$ haben, sofern $n$ prim ist

# Pseudo-Primzahlen und der Fermat-Test

- Der zugehörige generische Primzahltest $\mathcal{GT}$ arbeitet dann wie folgt:

**Algorithmus** $\mathcal{GT}(n, k)$, $k \geq 1$

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

- Wegen Bedingung P gibt $\mathcal{GT}(n, k)$ für primes $n$ immer „prim“ aus
- Zudem lässt sich für zusammengesetztes $n \notin \mathcal{P}$ über den Parameter $k$ die maximale Fehlerwahrscheinlichkeit von $\mathcal{GT}(n, k)$ begrenzen
- Wegen Bedingung Z gilt nämlich $t_n := |T_n| \leq \varepsilon\varphi(n)$ für alle $n \notin \mathcal{P}$ und eine Konstante $\varepsilon < 1$, und daher folgt

$$\Pr[GT(n, k) = \text{„prim“}] = (t_n/(n-1))^k < (t_n/\varphi(n))^k \leq \varepsilon^k$$

- Da der Algorithmus falsche Ergebnisse liefern kann, handelt es sich um einen Monte-Carlo-Algorithmus (mit einseitigem Fehler, da falsche Ergebnisse nur bei zusammengesetzten Eingaben möglich sind)

# Pseudo-Primzahlen und der Fermat-Test

- Im Gegensatz hierzu gibt ein Las-Vegas-Algorithmus nie ein falsches Ergebnis aus (d.h. er lügt nicht)
- Allerdings darf dieser mit begrenzter Wahrscheinlichkeit die Antwort verweigern (also „?“ ausgeben)
- Es liegt nahe, den kleinen Satz von Fermat zur Konstruktion der Testmengen $T_n^{\mathrm{FT}} = \{a \in \mathbb{Z}_n^* \mid a^{n-1} \equiv_n 1\}$ zu benutzen

**Algorithmus** $\mathcal{FT}(n,k)$, $n \geq 3$ ungerade und $k \geq 1$

```
1  berechne die Binärdarstellung von 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 ≠ 1 then return(„zusammengesetzt“)
9  return(„prim“)
```

# Pseudo-Primzahlen und der Fermat-Test

- Der Fermat-Test berechnet also die Potenz $z_0(a) = a^{n-1}$ genau wie HornerPot ausgehend von $z_r(a) = a$ mittels $z_{i-1}(a) = z_i(a)^2 a^{e_{i-1}} \bmod n$
- Er erkennt $n$ als zusammengesetzt, falls $z_0(a) \neq 1$ ist
- Man nennt eine zusammengesetzte Zahl $n$, die den Fermat-Test bei Wahl der Basis $a \in \mathbb{Z}_n^*$ besteht (d. h. es gilt $a^{n-1} \equiv_n 1$) eine Fermat-Pseudo-Primzahl oder einfach Pseudo-Primzahl zur Basis $a$
- Man bezeichnet eine solche Basis auch als einen (falschen) Primzahlzeugen für die Zahl $n$
- Zum Beispiel ist die Zahl 91 pseudo-prim zur Basis 3, da $3^{90} \equiv_{91} 1$ ist
- Es gibt Zahlen (z. B. $n = 561$) die pseudo-prim zu jeder Basis $a \in \mathbb{Z}_n^*$ sind (sogenannte Carmichael-Zahlen)
- Carmichael-Zahlen kommen nur sehr selten vor (erst 1992 konnte die Existenz unendlich vieler Carmichael-Zahlen nachgewiesen werden)
- Für diese Zahlen ist die Bedingung Z nicht erfüllt, weshalb der Fermat-Test als Pseudo-Primzahltest bezeichnet wird

# Der Miller-Rabin-Test

- Es ist leicht zu sehen, dass der Fermat-Test die Bedingung Z nur bei Eingabe einer Carmichael-Zahl verletzt und sie bei allen anderen Eingaben für $\varepsilon = 1/2$ erfüllt
- Der Fermat-Test lässt sich wie folgt zu einem „echten“ Monte-Carlo-Primzahltest, dem Miller-Rabin-Test (kurz MRT), erweitern
- Der Miller-Rabin-Test berechnet wie der Fermat-Test die Potenzen $z_r(a) = a$ und $z_{i-1}(a) = z_i(a)^2 a^{e_i} \bmod n$ für $i = r, \ldots, 1$
- Dabei überprüft er bei jeder Quadrierung, ob $z_i(a) \not\equiv_n \pm 1$ gilt und das Ergebnis $z_i(a)^2 \equiv_n 1$ ist
- Ist dies der Fall, so muss $n$ zusammengesetzt sein, da $z_i(a)$ eine nicht-triviale Lösung der Kongruenz $x^2 \equiv_n 1$ in $\mathbb{Z}_n^*$ ist

# Der Miller-Rabin-Test

- Der Miller-Rabin-Test akzeptiert also nur die Elemente der Testmenge

$$T_n^{\mathrm{MRT}} = \left\{ a \in \mathbb{Z}_n^* \,\middle|\, \begin{array}{l} z_i(a)^2 \equiv_n 1 \Rightarrow z_i(a) \equiv_n \pm 1 \text{ für} \\ \text{alle } i = 1, \ldots, r \text{ und } z_0(a) = 1 \end{array} \right\}$$

  als Primzahlzeugen (diese werden als starke Primzahlzeugen für die Zahl $n$ bezeichnet)
- Ist $n$ zusammengesetzt und $a \in T_n^{\mathrm{MRT}}$, so wird $n$ als starke Pseudo-Primzahl zur Basis $a$ bezeichnet
- Es gibt nur eine Zahl $n < 2,5 \cdot 10^{10}$, die stark pseudo-prim zu den Basen 2, 3, 5 und 7 ist: $n = 3\,215\,031\,751 = 151 \cdot 751 \cdot 28\,351$

# Der Miller-Rabin-Test

**Algorithmus** $\mathcal{MRT}(n,k)$, $n \geq 3$ ungerade und $k \geq 1$

berechne die Binärdarstellung $n-1 = \sum_{i=0}^{r} e_i \cdot 2^i$ mit $e_r = 1$
**for** $j := 1$ **to** $k$ **do**
  **guess randomly** $a \in \{1, \ldots, n-1\}$
  $z := a$
  **for** $i := r-1$ **downto** $0$ **do**
    $y := z$
    $z := z^2 \bmod n$
    **if** $z = 1 \wedge y \not\equiv_n \pm 1$ **then**
      **return**(„zusammengesetzt“)
    **if** $e_i = 1$ **then**
      $z := z \cdot a \bmod n$
  **if** $z \neq 1$ **then**
    **return**(„zusammengesetzt“)
**return**(„prim“)

# Der Miller-Rabin-Test

## Beispiel

- Für $n = 221 = 13 \cdot 17$ und $a = 137$, $a' = 18$ sowie $a'' = 174$ berechnet der Fermat-Test folgende Werte $z_i = z_i(a)$, $z_i' = z_i(a')$ und $z_i'' = z_i(a'')$:

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

- Dagegen berechnet der Miller-Rabin-Test nur die fett gedruckten Werte
- Bei Wahl der Basis $a = 137$ erkennen also beide Tests die Zahl $n = 221$ als zusammengesetzt, bei Wahl von $a' = 18$ gelingt dies nur dem MRT und bei Wahl von $a'' = 174$ gelingt dies keinem der beiden Tests ◁

# Der Miller-Rabin-Test

- Es ist klar, dass die Mengen $T_n^{\mathrm{MRT}}$ die Bedingungen E und P erfüllen
- Wir zeigen nun, dass jede ungerade zusammengesetzte Zahl $n > 2$ höchstens $\varphi(n)/2$ starke Primzahlzeugen hat, d.h. der Miller-Rabin-Test erfüllt auch die Bedingung Z
- Sei $n - 1 = 2^\ell u$ mit $u$ ungerade, d.h. $e_0 = \cdots = e_{\ell-1} = 0$, $e_\ell = 1$ und $c_\ell = \sum_{i=\ell}^{r} e_i \cdot 2^{i-\ell} = u$
- Zudem sei $U_n = \{a \in \mathbb{Z}_n^* \mid z_m(a) \equiv_n \pm 1\}$, wobei $m$ der kleinste Index $i \geq 0$ ist, für den ein $a \in \mathbb{Z}_n^*$ mit $z_i(a) \equiv_n -1$ existiert
- Dann ist $m \leq \ell$, da $z_\ell(-1) \equiv_n (-1)^u = -1$ ist

# Der Miller-Rabin-Test

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

**Behauptung 1.** $U_n = \{a \in \mathbb{Z}_n^* \mid z_m(a) \equiv_n \pm 1\}$ 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 $z_m(ab) \equiv_n (ab)^{c_m} = a^{c_m} b^{c_m} \equiv_n z_m(a) z_m(b) \equiv_n \pm 1$ $\square$

# Der Miller-Rabin-Test

## Behauptung 2. $T_n^{\mathrm{MRT}} \subseteq U_n$

- Für ein beliebiges $a \in T_n^{\mathrm{MRT}}$ gilt
  - $z_0(a) = 1$ und
  - $z_{i+1}(a)^2 \equiv_n 1 \Rightarrow z_{i+1}(a) \equiv_n \pm 1$ für $i = 0, \ldots, r-1$
- Da $z_i(a) \equiv_n z_{i+1}(a)^2$ für $i = 0, \ldots, \ell - 1$ ist (wegen $e_i = 0$), folgt

$$z_i(a) \equiv_n 1 \Rightarrow z_{i+1}(a) \equiv_n \pm 1 \text{ für } i = 0, \ldots, \ell - 1 \qquad (*)$$

- Da zudem $z_i(a) \not\equiv_n -1$ für $i < m$ ist (nach Definition von $m$), folgt

$$z_i(a) \equiv_n 1 \Rightarrow z_{i+1}(a) \equiv_n 1 \text{ für } i = 0, \ldots, m-1 \qquad (**)$$

- Somit folgt

$$z_0(a) \equiv_n 1 \overset{(**)}{\Rightarrow} z_1(a) \equiv_n 1 \overset{(**)}{\Rightarrow} \cdots \overset{(**)}{\Rightarrow} z_{m-1}(a) \equiv_n 1 \overset{(*)}{\Rightarrow} z_m(a) \equiv_n \pm 1$$

also ist $a \in U_n$ □

**Behauptung 3.** Für zusammengesetztes $n \equiv_2 1$ gilt $U_n \subsetneq \mathbb{Z}_n^*$

- 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 $a = p^{k-1} + 1 \in \mathbb{Z}_n^* \setminus U_n$
- Andernfalls können wir $n$ in zwei Faktoren $n = n_1 n_2$ mit $n_1, n_2 > 2$ und $\mathrm{ggT}(n_1, n_2) = 1$ zerlegen
- Zudem gibt es nach Definition von $m$ ein $b \in \mathbb{Z}_n^*$ mit $z_m(b) \equiv_n -1$
- Dann ist die eindeutige Lösung $a \in \mathbb{Z}_n^*$ von

  $$x \equiv_{n_1} b,$$
  $$x \equiv_{n_2} 1$$

  nicht in $U_n$ enthalten:
  - $z_m(a) \equiv_{n_1} a^{2^{\ell-m}u} \equiv_{n_1} b^{2^{\ell-m}u} \equiv_{n_1} z_m(b) \equiv_{n_1} -1 \;\Rightarrow\; z_m(a) \not\equiv_n 1$
  - und $z_m(a) \equiv_{n_2} a^{2^{\ell-m}u} \equiv_{n_2} 1^{2^{\ell-m}u} = 1 \;\Rightarrow\; z_m(a) \not\equiv_n -1$ □

# 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 $n$,

$$|T_n^{\mathrm{MRT}}| \leq |U_n| \leq \varphi(n)/2$$

- Damit haben wir bewiesen, dass der Miller-Rabin-Test die Bedingung Z für $\varepsilon = 1/2$ erfüllt (tatsächlich erfüllt er sie sogar für $\varepsilon = 1/4$)
- Unter Verwendung der verallgemeinerten Riemannschen Hypothese kann man sogar zeigen, dass es keine Zahl $n$ gibt, die stark pseudo-prim zu allen Basen $a < 2 \cdot (\ln n)^2$ ist
- Unter dieser Hypothese kann der Miller-Rabin-Test daher zu einem deterministischen Polynomialzeit-Algorithmus derandomisiert werden
- Erst 2002 fanden Agrawal, Kayal und Saxena einen Algorithmus, der das Primzahlproblem auch ohne diese Voraussetzung in P entscheidet