

# ***Computer Algebra***

Sebastian Iwanowski  
FH Wedel

## 3. Modulare Arithmetik

### 3.3. Primzahltest mit Hilfe modularer Arithmetik

#### **Referenzen zum Nacharbeiten:**

Köpf 4.6, Kaplan 5 (nur zur Vertiefung, wird hier nicht behandelt)

Seminararbeit 3, Kapitel 4 (Thomas Stuht)

Originalarbeit AKS (nur zur Vertiefung, wird hier nicht behandelt)

Reportage von Prof. Bornemann (DMV 2002)

# Computer Algebra 3

## Entschlüsselung von RSA-Codes: Faktorisierung ganzer Zahlen

- Probedivision  
mit Test, ob Division überhaupt erfolgversprechend ist  
exponentielle Laufzeit  
Details: Vortrag Stuhl  
Kapitel 4
- Primzahltest  
Rabin-Miller-Test zum Bestimmen, ob Zahl wahrscheinlich Primzahl ist  
Details: Köpf 4.6
- Spezielle Faktorisierungsverfahren für große Faktoren  
sprengt den Rahmen dieser Vorlesung  
Details: Kaplan 5

Asymptotisch effiziente Algorithmen zur Faktorisierung sind nicht bekannt!

# Computer Algebra 3

## Rabin-Miller-Test

### Kleiner Satz von Fermat:

$p$  Primzahl  $\Rightarrow \forall a \in \mathbb{Z}_p: a^p \equiv a \pmod{p}$

$\exists a \in \mathbb{Z}_p: a^p \not\equiv a \pmod{p} \Rightarrow p$  zerlegbar

Solch ein  $a$  heißt *Fermatscher Zeuge*  
für die Zerlegbarkeit von  $p$

### Die Umkehrung des Satzes von Fermat gilt nicht:

$\exists p \in \mathbb{N}: p$  zerlegbar  $\wedge (\forall a \in \mathbb{Z}_p: a^p \equiv a \pmod{p})$

Solch ein  $p$  heißt  
Carmichaelzahl

### Charakterisierung der Carmichaelzahlen:

$p$  Carmichaelzahl  $\Leftrightarrow p = p_1 \cdot p_2 \cdot \dots \cdot p_k$ , alle  $p_i$  paarweise verschiedene Primzahlen  
und  $p_i - 1 \mid p - 1$  für alle  $p_i$

# Computer Algebra 3

## Rabin-Miller-Test

**p (ungerade) heißt strenge Pseudoprimzahl zur Basis a:**

- i)  $p-1 = 2^t \cdot u$ , u ungerade      ii)  $\text{ggT}(p,a) = 1$       Solch ein a heißt  
*Rabin-Miller-Zeuge*  
iii)  $a^u \equiv 1 \pmod{p}$  oder  $\exists s \in \{0, 1, \dots, t-1\}: a^{2^s \cdot u} \equiv p-1 \pmod{p}$       für die Zerlegbarkeit von p

**Satz:** Jede Carmichaelzahl hat einen Rabin-Miller-Zeugen,  
d.h. der Rabin-Miller-Test erkennt Carmichaelzahlen als nicht prim  
für gewisse a.

**Die Umkehrung des Satzes von Fermat gilt nicht:**

$\exists p \in \mathbb{N}: p \text{ zerlegbar} \wedge (\forall a \in \mathbb{Z}_p: a^p \equiv a \pmod{p})$       Solch ein p heißt  
Carmichaelzahl

**Charakterisierung der Carmichaelzahlen:**

p Carmichaelzahl  $\Leftrightarrow p = p_1 \cdot p_2 \cdot \dots \cdot p_k$ , alle  $p_i$  paarweise verschiedene Primzahlen  
und  $p_i - 1 \mid p - 1$  für alle  $p_i$

# Computer Algebra 3

## Rabin-Miller-Test

**p (ungerade) heißt strenge Pseudoprimzahl zur Basis a:**

- i)  $p-1 = 2^t \cdot u$ , u ungerade      ii)  $\text{ggT}(p,a) = 1$   
iii)  $a^u \equiv 1 \pmod{p}$  oder  $\exists s \in \{0, 1, \dots, t-1\}: a^{2^s \cdot u} \equiv p-1 \pmod{p}$

**Satz:** Wird a zufällig gewählt, dann ist eine strenge Pseudoprimzahl zur Basis a mit Wahrscheinlichkeit  $> \frac{3}{4}$  eine echte Primzahl

**Korollar:** Durch mehrfache Wahl von zufälligen Basen a kann für jede Zahl p durch den Pseudoprimzahltest mit beliebiger Wahrscheinlichkeit  $< 1$  bestimmt werden, ob p Primzahl ist oder nicht

**Anmerkung:** Für kleine zusammengesetzte Zahlen n reichen die ersten p Primzahlen als Rabin-Miller-Zeugen aus:

$\leq p$	$\leq n$
2	2047
3	1373653
5	25326001
7	3215031751
11	2152302898747

# Computer Algebra 3

## AKS-Test

**Kleiner Satz von Fermat für Polynome:**

$$p \text{ Primzahl} \Leftrightarrow \forall a \in \mathbb{Z}_p \forall x \in \mathbb{Z}_p: (x+a)^p \equiv x^p + a \pmod{p}$$

**Idee:**

**Finde ein  $r$ , das logarithmisch in  $p$  ist, mit**

$$p \text{ Primzahl} \Leftrightarrow \forall a \in \mathbb{Z}_p \forall x \in \mathbb{Z}_p: (x+a)^p \equiv x^p + a \pmod{x^r-1, p}$$

↑  
Polynomdivision in  $\mathbb{Z}_p$

# Computer Algebra 3

## AKS-Test

### AKS-Algorithmus:

Eingabe: Zahl  $p \in \mathbb{N}$

Ausgabe: **true** für “p ist Primzahl”, **false** für “p ist zusammengesetzt”

1. Wenn  $p$  Primzahlpotenz ( $p = a^b$  für eine Primzahl  $a$ ), return **false**
2. Suche das kleinste  $r$ , sodass in  $Z_r$  gilt:  $o(p) > (\log p)^2$  ( $o_r(p)$  ist die Ordnung modulo  $r$ )
3. Wenn es ein  $a \leq r$  gibt mit  $\text{ggT}(a,p) > 1$ , return **false**
4. Wenn  $p \leq r$ , return **true**
5. For  $a := 1$  to  $\sqrt{\varphi(r) \cdot \log(p)}$  do  
    Wenn  $(x+a)^p \not\equiv x^p + a \pmod{x^r-1, p}$ , return **false**
6. return **true**

**Satz (Korrektheit):**  $p$  ist Primzahl  $\Leftrightarrow$  AKS-Algorithmus gibt **true** aus

**Satz (Laufzeit):** Der AKS-Algorithmus stoppt nach  $O((\log p)^{11})$  Operationen.