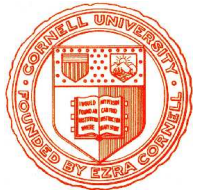
Kryptographie und Komplexität



Einheit 4.2

Primzahltests

1. Deterministische Primzahltests
2. Der Primzahltest von Solovay-Strassen
3. Der Milner-Rabin Test

Wozu Primzahltests?

- **RSA Schlüssel benötigen sehr große Primzahlen**
  - Große Primzahlen können & sollen nicht systematisch erzeugt werden
    - Es gibt keine geeigneten Formeln zur Generierungen von Primzahlen
    - Systematisch erzeugte Primzahlen können leicht reproduziert werden
  - Primzahlen sollten zufällig erzeugt werden
- **Primzahlen sollten zufällig erzeugt werden**
  - Erzeuge zufällige Bitfolge und teste ob zugehörige Zahl p Primzahl ist
    Wiederhole das Verfahren bis eine Primzahl gefunden wurde
  - Aufwand abhängig von Primzahldichte und Laufzeit von Primzahltests
- **Wie dicht liegen Primzahlen?**
  - Es gibt $n/ln\ n$ Primzahlen zwischen 1 und n (Gauß'sches Primzahltheorem)
  - Wahrscheinlichkeit, daß eine zufällige (ungerade) 512 Bit Zahl
    ein Primzahl ist, liegt bei $2/ln(2^{512}) = 1/177$
  - Man braucht 355 Primzahltests zur Erzeugung der Primzahlen p und q

Primzahltests

- **Deterministische Tests**
  - Liefern sichere Erkenntnis, ob eine Zahl n Primzahl ist
  - Stützen sich meist auf Teilbarkeitsuntersuchungen
  - Liefern oft eine (implizite) Faktorisierung von n, wenn n nicht prim
  - Sind für praktische Anwendungen erheblich zu langsam
    - Klassische Testverfahren haben exponentielle Laufzeit in $\|n\|$
    - AKS-Test (2002) ist polynomiell, aber mit sehr hohem Overhead
- **Probabilistische Tests**
  - Erlauben mit kleiner Wahrscheinlichkeit ϵ ein falsches Ergebnis
  - Können Wahrscheinlichkeit ϵ durch Iteration beliebig klein machen
  - Benötigen polynomielle Laufzeit ($\mathcal{O}(\|n\|^3)$) mit geringem Overhead
  - Werden mit großem Erfolg in der Praxis eingesetzt

Probedivision

Ältestes und einfachstes aller Testverfahren

- **Überprüfe alle möglichen Teiler von n**
  - Brute-Force Ansatz: Versuche n durch jedes $i \in \{2..n-1\}$ zu teilen
  - Entspricht der mathematischen Definition des Begriffs Primzahl
  - Hochgradig ineffizient
- **Viele Optimierungen möglich**
  - Alle Primzahlen größer als 2 sind ungerade
    Beschränke mögliche Teiler auf ungerade Zahlen
  - Jede zusammengesetzte Zahl n hat einen Teiler nicht größer als $\sqrt{n}$
    Beschränke Suche auf Zahlen bis $\sqrt{n}$
  - Alle zusammengesetzten Zahlen sind durch Primzahlen teilbar
    Beschränke Suche auf (erkannte) Primzahlen
    Sieb des Eratosthenes: In der Liste der Zahlen zwischen 2 und n streiche alle Vielfachen von (nicht gestrichenen) Zahlen zwischen 2 und $\sqrt{n}$
    Übrig bleibt Liste aller Primzahlen zwischen 2 und n
  - Effizent für kleine Zahlenbereiche, aber exponentiell in $\|n\|$

Der AKS Test

- **Polynomieller deterministischer Primzahltest**
  - Benannt nach Erfindern Agrawal, Kayal, Saxena
  - Lösung eines jahrzehntelang offenen Problems der Komplexitätstheorie
    "Liegt PRIMES in $\mathcal{P}$?"
  - Aus theoretischer Sicht extrem bedeutendes Ergebnis
- **Aufwendige mathematische Konstruktion**
  - Verwendet elementare zahlentheoretische Erkenntnisse
  - Originalalgorithmus von 2002 liegt in $\mathcal{O}(\|n\|^{12})$
    Verbesserte Analysen zeigen Komplexität $\mathcal{O}(\|n\|^{6})$
  - Große Faktoren und Konstanten machen Testverfahren unpraktikabel
  - Trotzdem geeignet als Basis für Entwicklung eines echten deterministischen Primzahltests mit (guter) polynomieller Laufzeit

Fermat Test

- **Prüft Bedingungen des Satzes von Fermat**
  - *Ist n Primzahl und $gcd(a,n)=1$, so folgt* $a^{n-1} \bmod n = 1$
  - Test identifiziert zusammengesetzte Zahlen ohne Faktorisierung
    - Wähle zufälliges $a<n$
    - Gilt $gcd(a,n)\neq 1$ oder $a^{n-1} \bmod n \neq 1$, so ist n keine Primzahl
- **Fermat-Test ist kein Entscheidungsverfahren**
  - Wenn der Test fehlschlägt, ist n nicht notwendigerweise eine Primzahl
    - Für $n=25$ und $a=7$ gilt $gcd(a,n)=1$ und $a^{n-1} \bmod n = 1$
  - Zusammengesetzte Zahlen, die für ein a den kleinen Satz von Fermat erfüllen, heißen (Fermat'sche) **Pseudoprimzahlen**
- **Wie oft irrt der Fermat-Test?**
  - Für wieviele $a<n$ gilt $a^{n-1} \bmod n = 1$, wenn n Pseudoprimzahl?
    - Für $n=25$ gilt dies für $a = 7, 18, 24$
  - Aber es gibt zusammengesetzte Zahlen, welche $a^{n-1} \bmod n = 1$ für alle $a<n$ mit $gcd(a,n)=1$ erfüllen (Carmichael Zahlen)

Carmichael Zahlen

- **Die ultimativen Pseudoprimzahlen**
  - Für alle $a<n$ mit $gcd(a,n)=1$ gilt $a^{n-1} \bmod n = 1$

    z.B. 561 (3·11·17), 1105 (5·13·17), 1729 (7·13·19), ...
- **Identifizierbar über äquivalente Kriterien**
  - Eine ungerade zusammengesetzte Zahl $n>2$ ist genau dann eine Carmichael Zahl, wenn sie keinen mehrfachen Primfaktor hat und für jeden Primteiler p von n die Zahl $p-1$ die Zahl $n-1$ teilt

    $\Rightarrow$: Sei n Carmichael Zahl mit Primteiler p und a Erzeuger von $\mathbb{Z}_p$ mit $gcd(a,n)=1$. Dann gilt $a^{n-1} \bmod n = 1$, also $a^{n-1} \bmod p = 1$. Damit ist $n-1$ ein Vielfaches von $p-1$, der Ordnung von a. Wäre p^2 Teiler von n, so wäre $(p-1)p$ Teiler von $\varphi(n)$ und in $\mathbb{Z}_n$ gäbe es Elemente der Ordnung p. Somit würde p auch $n-1$ teilen.

    $\Leftarrow$: Sei n ohne mehrfache Primfaktoren, $p-1$ Teiler von $n-1$ für Primteiler p von n und $gcd(a,n)=1$. Mit dem Satz von Fermat folgt $a^{p-1} \bmod p = 1$, folglich $a^{n-1} \bmod p = 1$ und damit $a^{n-1} \bmod n = 1$.
  - Konsequenz: Für große Carmichael Zahlen ist $\varphi(n) = \prod_{p|n}(p-1) \approx n$
  - Jede Carmichael Zahl hat mindestens drei (verschiedene) Primteiler

Probabilistische Entscheidungsverfahren

- **Der Fermat-Test irrt zu oft**
  - Carmichael Zahlen werden in $\varphi(n)$ von n Fällen (falsch) akzeptiert
  - Die Wahrscheinlichkeit, daß das Ergebnis 'n ist Primzahl' korrekt ist, geht für große Carmichael Zahlen gegen Null
  - Keine Chance auf zuverlässige Aussagen durch Iteration des Tests
- ***RP*-Algorithmus** ("random polynomial")
  - Polynomieller Entscheidungsalgorithmus, bei dem die Antwort *ja* immer korrekt ist und die Antwort *nein* mit Wahrscheinlichkeit $\epsilon > 0$ korrekt ist
  - $\frac{k}{-\log(1-\epsilon)}$ Iterationen senken Fehlerwahrscheinlichkeit auf 2^{-k}
- ***ZPP* Algorithmus** ("zero error probabilistic polynomial")

  Auch **Las Vegas Algorithmus** genannt
  - Entscheidungsalgorithmus, bei dem die Antwort immer korrekt ist
  - Laufzeit nur im Erwartungswert polynomiell (in seltenen Fällen länger)

Ein aussagekräftigeres Primzahlkriterium

- **Quadratischer Rest modulo n**
  - Zahl $a \in \mathbb{Z}_n^*$ für welche die Kongruenz $y^2 \equiv a \bmod n$ eine Lösung hat
  - z.B. sind 1,4,5,6,7,9,11,16,17 quadratische Reste modulo 19

y	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
y^2	1	4	9	16	6	17	11	7	5	5	7	11	17	6	16	9	4	1

  - Für Primzahlen hat $y^2 \equiv a \bmod n$ genau 2 Lösungen oder keine
  - Es gibt jeweils $(n-1)/2$ quadratische Reste modulo n
- **Eulers Kriterium für quadratische Reste**

  Für jede Primzahl $n>2$ ist a genau dann ein quadratischer Rest modulo n, wenn $a^{(n-1)/2} \equiv 1 \bmod n$

  $\Rightarrow$: Es gelte $y^2 \equiv a \bmod n$. Mit dem Satz von Fermat folgt

  $$a^{(n-1)/2} \equiv (y^2)^{(n-1)/2} \bmod n \equiv y^{n-1} \bmod n \equiv 1 \bmod n$$

  $\Leftarrow$: Es gelte $a^{(n-1)/2} \equiv 1 \bmod n$.

  Es gilt $a \equiv b^i \bmod n$ für einen Erzeuger b von $\mathbb{Z}_n^*$ und ein i.

  Es folgt $1 \equiv a^{(n-1)/2} \bmod n \equiv (b^i)^{(n-1)/2} \bmod n \equiv b^{i(n-1)/2} \bmod n$

  Da b die Ordnung $n-1$ hat, muß $n-1$ Teiler von $i(n-1)/2$ sein

  Also ist i gerade und $(b^{i/2})^2 \equiv a \bmod n$

Schnelle Bestimmung quadratischer Reste

- **Legendre Symbol für Primzahlen**
  - Für eine Primzahl $n>2$ und $a \in \mathbb{Z}_n^*$ ist

$$\left(\frac{a}{n}\right) = \begin{cases} 0 & \text{falls } a \equiv 0 \bmod n \\ 1 & \text{falls } a \text{ quadratischer Rest modulo } n \\ -1 & \text{falls } a \text{ quadratischer Nichtrest modulo } n \end{cases}$$

- **Jacobi Symbol für beliebige Zahlen**
  - Für $n = p_1^{e_1}..p_k^{e_k}$ und $a \in \mathbb{Z}_n^*$ ist $\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}$
- **Satz: $\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \bmod n$ für jede Primzahl $n>2$**
  - Für quadratische Reste gilt $\left(\frac{a}{n}\right) \equiv 1 \equiv a^{(n-1)/2} \bmod n$
  - Ist $a \equiv 0 \bmod n$, so ist auch $a^{(n-1)/2} \equiv 0 \bmod n$
  - Für quadratische Nichtreste gilt $a^{(n-1)/2} \not\equiv 0 \bmod n$ und $(a^{(n-1)/2})^2 = a^{n-1} \equiv 1 \bmod n$ also $\left(\frac{a}{n}\right) \equiv -1 \equiv a^{(n-1)/2} \bmod n$

Schnelle Auswertung des Jacobi Symbols

- **Rechengesetze für $\left(\frac{a}{n}\right)$ und ungerade $n>2$**
  1. Für ungerade a ist $\left(\frac{a}{n}\right) = \begin{cases} -\left(\frac{n}{a}\right) & \text{falls } a \equiv n \equiv 3 \bmod 4 \\ \left(\frac{n}{a}\right) & \text{sonst} \end{cases}$
  2. Für alle a ist $\left(\frac{a}{n}\right) = \left(\frac{a \bmod n}{n}\right)$
  3. Für $a = b \cdot 2^k$ ist $\left(\frac{a}{n}\right) = \left(\frac{b}{n}\right)\left(\frac{2}{n}\right)^k$
  4. $\left(\frac{2}{n}\right) = \begin{cases} 1 & \text{falls } n \equiv \pm 1 \bmod 8 \\ -1 & \text{falls } n \equiv \pm 3 \bmod 8 \end{cases}$ $\quad \left(\frac{1}{n}\right) = 1 \quad \left(\frac{0}{n}\right) = 0$

  Beweis durch Detailanalyse und Anwendung zahlentheoretischer Reziprozitätsgesetze
- **Beispielauswertung**

$$\left(\frac{7411}{9283}\right) = -\left(\frac{9283}{7411}\right) = -\left(\frac{1872}{7411}\right) = -\left(\frac{117}{7411}\right)\left(\frac{2}{7411}\right)^4 = -\left(\frac{117}{7411}\right) = -\left(\frac{7411}{117}\right)$$
$$= -\left(\frac{40}{117}\right) = -\left(\frac{5}{117}\right)\left(\frac{5}{117}\right)^3 = \left(\frac{5}{117}\right) = \left(\frac{117}{5}\right) = \left(\frac{2}{5}\right) = \mathbf{-1}$$

- **Rechenzeit für $\left(\frac{a}{n}\right)$ liegt in $\mathcal{O}(\|n\|^3)$**
  - Wie bei `egcd` sind $\|n\|$ Anwendungen der Regeln 1–3 erforderlich
  - Division durch 2^k ist ein Bitshift entsprechend der Nullen am Ende
  - Berechnung von $a \bmod n$ ist in $\mathcal{O}(\|n\|^2)$ Schritten möglich

Eulersche Pseudoprimzahlen

$$\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \bmod n \quad \text{für jede Primzahl } n>2$$

- **Eulers Gesetz liefert verfeinertes Testverfahren**
  - Wähle zufälliges $a<n$
  - Gilt $gcd(a,n)\neq 1$, so ist n keine Primzahl
  - Gilt $a^{(n-1)/2} \bmod n \neq \left(\frac{a}{n}\right)$, so ist n keine Primzahl
- **Test ist kein Entscheidungsverfahren**
  - Es gibt zusammengesetzte Zahlen, die für ein a diesen Satz erfüllen (Eulerschen Pseudoprimzahlen)
    z.B. ist $\left(\frac{20}{21}\right) = \left(\frac{5}{21}\right) = \left(\frac{1}{5}\right) = 1 \equiv 20^{10} \bmod n$
  - Aber für jede Pseudoprimzahl n gibt es maximal $n/2$ falsche Zeugen
  - Die Wahrscheinlichkeit, daß aus $\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \bmod n$ folgt, daß n tatsächlich eine Primzahl ist, liegt über $1/2$ (Rothe, Theorem 7.25)
  - Eulers Test ist als Grundlage für einen RP-Algorithmus geeignet

Der Solovay-Strassen Primzahltest

1. Ist n gerade dann ist n keine Primzahl
2. Ansonsten wähle $a \in \{1 \ldots n-1\}$ zufällig
3. Ist $\left(\frac{a}{n}\right) = 0$ dann ist n keine Primzahl
4. Ansonsten teste $\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \bmod n$

   Ist dies nicht der Fall, dann ist n keine Primzahl.

   Ansonsten akzeptiere n als Primzahl.

- **RP-Algorithmus für Zusammengesetztheit**
  - Das Resultat "n ist keine Primzahl" ist immer korrekt
  - Das Resultat "n ist Primzahl" ist in mindestens 50% der Fälle korrekt
- **Rechenzeit liegt in $\mathcal{O}(\|n\|)^3$**
  - Modulares Potenzieren liegt in $\mathcal{O}(\|n\|)^3$
  - Berechnung des Jacobi Symbols liegt in $\mathcal{O}(\|n\|)^3$

Der Miller-Rabin Test

- **Verbesserung des Fermat-Tests**
  - "Strong Pseudo-Prime Test" auf Basis des Satzes von Fermat
  - Umgeht das Problem der Carmichael Zahlen
  - Deutlich verringerte Fehlerwahrscheinlichkeit (weniger als 1/4)
  - Besser als Solovay-Strassen, da Operationen einfacher sind und weniger Iterationen erforderlich werden
- **Miller-Rabin Testbedingung**
  - Fermat: Ist n Primzahl und $gcd(a,n)=1$, dann ist $a^{n-1} \bmod n = 1$
  - $n-1$ ist gerade, also $n-1 = 2^s \cdot d$ für ein ungerades d
  - Wegen $(a^d)^{2^s} \bmod n = 1$ ist $k = \text{order}(a^d) = 2^l$ für ein $l \leq s$ (k teilt 2^s)
  - Ist $l=0$, so gilt $a^d \bmod n = 1$
    Ansonsten hat $(a^d)^{k/2}$ die Ordnung 2 und $(a^d)^{2^{l-1}} \equiv -1 \bmod n$

Sei $n>2$ Primzahl, $s = max\{r \mid 2^r \text{ teilt } n-1\}$, $d = (n-1)/2^s$.
Für alle a mit $gcd(a,n) = 1$ gilt entweder $a^d \bmod n = 1$
oder es gibt ein $r<s$ mit $a^{2^r \cdot d} \equiv -1 \bmod n$.

Wie viele falsche Zeugen gibt es?

Für jede ungerade zusammengesetzte Zahl $n > 2$ gibt es höchstens $(n-1)/4$ Zahlen mit $gcd(a,n) = 1$, die falsche Zeugen für die Primalität von n sind

- **Falsche Zeugen für Primalität von n**
  - Zahlen $a<n$ mit $gcd(a,n) = 1$, für die entweder $a^d \bmod n = 1$ oder $a^{2^r \cdot d} \equiv -1 \bmod n$ für ein $r<s$ gilt
  - Wenn es keinen falschen Zeugen gibt ist der Satz beweisen
- **Es gibt falsche Zeugen mit $a^{2^r \cdot d} \equiv -1 \bmod n$ für ein $r<s$**
  - Aus $a^d \bmod n = 1$ folgt $(-a)^{2d} \bmod n = 1$, also $(-a)^{2^0 \cdot d} \equiv -1 \bmod n$
  - Sei k das größte r mit $a^{2^r \cdot d} \equiv -1 \bmod n$ für ein a und $m = 2^k \cdot d$
- **Definiere Untergruppen $M \subseteq L \subseteq K \subseteq J \subseteq \mathbb{Z}_n^*$ von $\mathbb{Z}_n^*$**
  - $J = \{a \mid gcd(a,n) = 1 \wedge a^{n-1} \bmod n = 1\}$
  - $K = \{a \mid gcd(a,n) = 1 \wedge \forall p|n.\, a^m \equiv \pm 1 \bmod p^{e(p)}\}$ $\quad (n = \prod_{p|n} p^{e(p)})$
  - $L = \{a \mid gcd(a,n) = 1 \wedge a^m \equiv \pm 1 \bmod n\}$
  - $M = \{a \mid gcd(a,n) = 1 \wedge a^m \equiv 1 \bmod n\}$

Beweis: es gibt maximal $(n-1)/4$ falsche Zeugen

$$J = \{a \mid gcd(a,n) = 1 \wedge a^{n-1} \bmod n = 1\}$$
$$K = \{a \mid gcd(a,n) = 1 \wedge \forall p|n.\, a^m \equiv \pm 1 \bmod p^{e(p)}\}$$
$$L = \{a \mid gcd(a,n) = 1 \wedge a^m \equiv \pm 1 \bmod n\}$$
$$M = \{a \mid gcd(a,n) = 1 \wedge a^m \equiv 1 \bmod n\}$$

- **Der Index von L in $\mathbb{Z}_n^*$ ist mindestens 4**
  - Der Index von M in K ist eine Zweierpotenz, da $a^2 \in M$ für alle $a \in K$
  - Der Index von L in K muß 2^j für ein j sein
  - Für $j \geq 2$ ist der Index von L in $\mathbb{Z}_n^*$ mindestens 4.
  - Für $j=1$ hat n zwei $(= 2^j)$ Primteiler und ist keine Carmichael Zahl.

    Dann ist J echte Untergruppe von $\mathbb{Z}_n^*$ mit Index 2 oder mehr.

    Damit ist der Index von L in $\mathbb{Z}_n^*$ mindestens $2*2 = 4$.
  - Ist $j=0$, so hat $n=p^e$ einen Primteiler und J hat genau $p-1$ Elemente.
  - Der Index von J in $\mathbb{Z}_n^*$ ist somit p^{e-1} und mindestens 4, falls $n \neq 9$.
  - Für $n=9$ gibt es nur einen einzigen falschen Zeugen $(a = 1)$

Der Miller-Rabin Algorithmus

1. Ist n gerade dann ist n keine Primzahl
2. Ansonsten zerlege $n-1$ in $2^s \cdot d$ für ein ungerades d
3. Wähle $a \in \{1 \ldots n-1\}$ zufällig
4. Ist $gcd(n, a) \neq 1$ dann ist n keine Primzahl
5. Setze $b := a^d \bmod n$
6. Ist $b \bmod n = 1$ dann ist n eine Primzahl
7. Wiederhole für $i = 1..s$
   - Ist $b \equiv -1 \bmod n$ dann ist n eine Primzahl
   - Ansonsten Setze $b := b^2 \bmod n$
8. Ansonsten ist n keine Primzahl

- **Fehlerwahrscheinlichkeit unter 25%**
- **Rechenzeit liegt in $\mathcal{O}(\|n\|)^3$**
  - Modulares Potenzieren liegt in $\mathcal{O}(\|n\|)^3$
  - Maximal $s \leq \|n\|$ Quadrierungen im Schritt 7 (je $\mathcal{O}(\|n\|)^2$

Fazit

- **Deterministische Primzahltests**
  - Sind leicht zu programmieren
  - Liefern meist Faktorisierungen
  - Sind nur geeignet für "kleine" Zahlen (z.B. bis 10^9)
  - Die Tests der AKS-Familie sind noch nicht für die Praxis geeignet
- **Probabilistische Primzahltests**
  - Sind in deutlich effizienter
  - Können Zahlen von mehreren tausend Bit überprüfen
  - Der Solovay-Strassen test brachte den Durchbruch
  - In der Praxis wird meist der Milner-Rabin Test oder ECPP eingesetzt