

Kryptologie

Johannes Köbler



Institut für Informatik
Humboldt-Universität zu Berlin

WS 2022/23

- **Pseudozufallszahlen-Generatoren** (kurz **PZG**) f werden mit einem Startwert x – dem sogenannten **Keim** (engl. seed) – für die Erzeugung einer „zufälligen“ Bitfolge $f(x)$ gestartet
- Dabei wird die Eingabe x zufällig unter Gleichverteilung aus einer endlichen Menge X gewählt und die Ausgabe $f(x)$ sollte länger sein als x und möglichst zufällig aussehen
- Zudem sollte die Funktion f von einem deterministischen Algorithmus effizient berechenbar sein

Beispiel

- Beim **Linear-Kongruenz-Generator** wird der Keim x_0 zufällig aus der Menge $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$ gewählt
- Die Parameter a und b sind ebenfalls aus $\mathbb{Z}_n$

Algorithmus $\text{LinGen}_{n,l,a,b}(x_0)$

```
1 for  $i := 1$  to  $l$  do
2    $x_i := ax_{i-1} + b \bmod n$ 
3    $b_i := x_i \bmod 2$ 
4 output( $b_1 \dots b_l$ )
```

Beispiel

- Beim **Power-Generator** wird der Keim x_0 zufällig aus der Menge $\mathbb{Z}_n^*$ gewählt

Algorithmus $\text{PowerGen}_{n,l,e}(x_0)$

```
1 for  $i := 1$  to  $l$  do
2    $x_i := x_{i-1}^e \bmod n$ 
3    $b_i := x_i \bmod 2$ 
4 output( $b_1 \dots b_l$ )
```

Es gibt zwei interessante Spezialfälle des Powergenerators:

- **RSA-Generator (RsaGen)** mit $n = p \cdot q$ wobei p und q große Primzahlen sind und $\text{ggT}(e, \varphi(n)) = 1$ ist
- **Quadratischer-Reste-Generator (BBS)** mit $e = 2$ (siehe unten)

Wir betrachten ab jetzt nur noch den Fall, dass sowohl x als auch $f(x)$ Bitfolgen sind und die Länge der Ausgabe $f(x)$ nur von der Länge der Eingabe x abhängt

Definition

- Sei $\ell : \mathbb{N} \rightarrow \mathbb{N}$ eine Funktion mit $\ell(k) \geq k + 1$ für alle $k \geq 0$
- Ein $\ell(k)$ -Generator ist eine Funktion f auf $\{0, 1\}^*$, die Strings der Länge k auf Strings der Länge $\ell(k)$ abbildet und effizient berechenbar ist
- Seien $(\mathcal{X}_k)$ und $(\mathcal{Y}_k)$, $k \geq 0$, Familien von Zufallsvariablen mit Wertebereich $W(\mathcal{X}_k), W(\mathcal{Y}_k) \subseteq \{0, 1\}^{\ell(k)}$ und sei $\varepsilon : \mathbb{N} \rightarrow \mathbb{R}^+$ eine Funktion
- Ein ε -Unterscheider zwischen $(\mathcal{X}_k)$ und $(\mathcal{Y}_k)$ ist ein effizienter probabilistischer Algorithmus D mit

$$\Pr[D(\mathcal{X}_k) = 1] - \Pr[D(\mathcal{Y}_k) = 1] \geq \varepsilon(\ell(k))$$

- Hierbei ist $\Pr[D(\mathcal{X}_k) = 1]$ die Wahrscheinlichkeit, dass D bei einer zufällig gemäß $\mathcal{X}_k$ gewählten Eingabe akzeptiert (bzw. 1 ausgibt)

Sicherheit von Pseudozufallsgeneratoren

Definition (Fortsetzung)

- In diesem Fall heißen $(\mathcal{X}_k)$ und $(\mathcal{Y}_k)$ ε -unterscheidbar
- Ein $\ell(k)$ -Generator f heißt ε -unterscheidbar, falls $(f(\mathcal{U}_k))$ und $(\mathcal{U}_{\ell(k)})$ ε -unterscheidbar sind, wobei $\mathcal{U}_n$ auf $\{0, 1\}^n$ gleichverteilt ist
- Eine Funktion $\varepsilon: \mathbb{N} \rightarrow \mathbb{R}$ heißt vernachlässigbar, wenn für jedes Polynom p eine Zahl $n_0 \in \mathbb{N}$ existiert, so dass $\varepsilon(n) < 1/p(n)$ für alle $n \geq n_0$ gilt
- f heißt (kryptografisch) sicher, falls f nur für vernachlässigbare Funktionen $\varepsilon: \mathbb{N} \rightarrow \mathbb{R}^+$ ε -unterscheidbar ist

- Ein $\ell(k)$ -Generator f ist also genau dann sicher, wenn es für jeden Unterscheider D und jedes Polynom p nur endlich viele Werte k gibt mit

$$\Pr[D(f(\mathcal{U}_k)) = 1] - \Pr[D(\mathcal{U}_{\ell(k)}) = 1] \geq 1/p(\ell(k))$$

- Unterscheider fungieren also als Gegner von Pseudozufallsgeneratoren und werden üblicherweise durch probabilistische Schaltkreise polynomieller Größe modelliert

Beispiel

- Betrachte folgenden Unterscheider D für den $\ell(k)$ -Generator f mit $\ell(k) = k + 1$ und $f(x) = 1x$ für alle $x \in \{0, 1\}^*$:

1 **input** $y = y_1 \cdots y_{k+1} \in \{0, 1\}^{k+1}$
2 **output**(y_1)

- Dann gilt $\Pr[D(f(\mathcal{U}_k)) = 1] = 1$ und $\Pr[D(\mathcal{U}_{k+1}) = 1] = 1/2$ und somit

$$\Pr[D(f(\mathcal{U}_k)) = 1] - \Pr[D(\mathcal{U}_{k+1}) = 1] = 1/2$$

für alle k

- Folglich ist f $(1/2)$ -unterscheidbar
- Da die konstante Funktion $n \mapsto 1/2$ nicht vernachlässigbar ist, ist der Generator f nicht sicher



- Es ist nicht bekannt, ob kryptografisch sichere PZGen existieren
- Eine notwendige Bedingung hierfür ist $P \neq NP$, da $P = NP$ die Existenz eines effizienten Unterscheiders impliziert, welcher genau die Strings im Bild von f akzeptiert
- Ob diese Bedingung auch hinreichend ist, ist ebenfalls nicht bekannt
- Man kann jedoch zeigen, dass die Existenz von kryptografisch sicheren PZGen äquivalent zur Existenz von Einwegfunktionen ist
- Bei manchen Anwendungen ist es wichtig, dass kein effizienter Algorithmus das nächste Bit der Pseudozufallsfolge korrekt vorhersagen kann
- Es ist nicht schwer zu sehen, dass ein sicherer PZG diese Bedingung erfüllt

Definition

- Sei f ein $\ell(k)$ -Generator und sei $\varepsilon : \mathbb{N} \rightarrow \mathbb{R}$ eine Funktion
- Für $i \in \{1, \dots, \ell(k)\}$ bezeichne $f_i(x)$ das i -te Bit und für $i \in \{0, \dots, \ell(k)\}$ bezeichne $f_{[i]}(x)$ die Folge der ersten i Bits von $f(x)$
- Ein **next bit predictor (NBP)** N ist ein effizienter probabilistischer Algorithmus, der bei jeder Eingabe $(y, 1^n)$ mit $y \in \{0, 1\}^{i-1}$ für ein $i \in \{1, \dots, n\}$ ein Bit $N(y, 1^n)$ ausgibt
- N heißt **ε -next bit predictor (ε -NBP)** für f , falls für alle k gilt:

$$\Pr[N(f_{[\mathcal{I}-1]}(\mathcal{U}_k), 1^{\ell(k)}) = f_{\mathcal{I}}(\mathcal{U}_k)] \geq 1/2 + \varepsilon(\ell(k))$$

wobei die Zufallsvariable $\mathcal{I}$ auf der Menge $\{1, \dots, \ell(k)\}$ gleichverteilt ist

Beispiel

- Betrachte folgenden NBP N für den $\ell(k)$ -Generator f mit $\ell(k) = k + 1$ und $f(x) = 1x$ für alle $x \in \{0, 1\}^*$:

1 **input** $(y, 1^n)$ mit $y = y_1 \cdots y_{i-1} \in \{0, 1\}^{i-1}$ für ein $i \in \{1, \dots, n\}$
2 **output**(1)

- Dann gilt

$$\Pr[N(f_{[i-1]}(\mathcal{U}_k)) = f_i(\mathcal{U}_k)] = \begin{cases} 1, & i = 1 \\ 1/2, & i = 2, \dots, k+1 \end{cases}$$

- Somit gilt

$$\begin{aligned} \Pr[N(f_{[I-1]}(\mathcal{U}_k)) = f_I(\mathcal{U}_k)] &= \frac{1}{k+1} \sum_{i=1}^{k+1} \Pr[N(f_{[i-1]}(\mathcal{U}_k)) = f_i(\mathcal{U}_k)] \\ &= 1/2 + 1/(2k+2) = 1/2 + 1/2\ell(k) \end{aligned}$$

- Also ist N ein $(1/2\ell)$ -NBP für f

Satz. Sei f ein $\ell(k)$ -Generator und sei $\varepsilon : \mathbb{N} \rightarrow \mathbb{R}^+$ eine Funktion.

Falls es einen ε -NBP für f gibt, so ex. auch ein ε -Unterscheider für f

Beweis.

- Sei N ein ε -NBP für f und betrachte folgenden Unterscheider D

```

1 input  $y = y_1 \cdots y_n$ 
2   wähle zufällig  $i \in_R \{1, \dots, n\}$ 
3 output  $(N(y_1 \cdots y_{i-1}, 1^n) \oplus y_i \oplus 1)$ 

```

- D gibt also bei Eingabe $y = y_1 \cdots y_n$ genau dann 1 aus, wenn der Prediktor N bei Eingabe $(y_1 \cdots y_{i-1}, 1^n)$ das i -te Bit von y richtig rät, wobei i zufällig aus $\{1, \dots, n\}$ gewählt wird
- Daher gilt für alle $k \geq 0$:

$$\Pr[D(f(\mathcal{U}_k)) = 1] = \Pr[N(f_{[\mathcal{I}-1]}(\mathcal{U}_k), 1^{\ell(k)}) = f_{\mathcal{I}}(\mathcal{U}_k)] \geq 1/2 + \varepsilon(\ell(k)),$$

wobei $\mathcal{I}$ auf $\{1, \dots, \ell(k)\}$ gleichverteilt ist

Beweis (Fortsetzung)

- Andererseits gilt für auf $\{0, 1\}$ unabhängig und gleichverteilte Zufallsvariablen $\mathcal{B}_1, \dots, \mathcal{B}_{\ell(k)}$ und jeden NBP N :

$$\Pr[N(\mathcal{B}_1 \dots \mathcal{B}_{\ell(k)-1}, 1^{\ell(k)}) = \mathcal{B}_{\ell(k)}] = 1/2$$

- Folglich gilt wegen $\mathcal{U}_{\ell(k)} = \mathcal{B}_1 \dots \mathcal{B}_{\ell(k)}$

$$\Pr[D(\mathcal{U}_{\ell(k)}) = 1] = \Pr[N(\mathcal{B}_1, \dots, \mathcal{B}_{\ell(k)-1}, 1^{\ell(k)}) = \mathcal{B}_{\ell(k)}] = 1/2,$$

was folgende Ungleichung impliziert:

$$\underbrace{\Pr[D(f(\mathcal{U}_k)) = 1]}_{\geq 1/2 + \varepsilon(\ell(k))} - \underbrace{\Pr[D(\mathcal{U}_{\ell(k)}) = 1]}_{1/2} \geq \varepsilon(\ell(k))$$

- Also ist D ein ε -Unterscheider für f



Definition

Ein probabilistischer Algorithmus P heißt ε -previous bit predictor (ε -PBP) für einen $\ell(k)$ -Generator f , falls für alle k gilt,

$$\Pr[P(f_{\mathcal{I}+1}(\mathcal{U}_k) \cdots f_{\ell(k)}(\mathcal{U}_k), 1^{\ell(k)}) = f_{\mathcal{I}}(\mathcal{U}_k)] \geq 1/2 + \varepsilon(\ell(k))$$

wobei $\mathcal{I}$ auf $\{1, \dots, \ell(k)\}$ gleichverteilt ist

Vollkommen analog zu obigem Satz lässt sich der folgende Satz beweisen

Satz

Falls es einen ε -PBP für f gibt, so ex. auch ein ε -Unterscheider für f

Interessanterweise lässt sich aus einem Unterscheider auch ein NBP (bzw. ein PBP) gewinnen

Satz

Falls es einen ε -Unterscheider für f gibt, so ex. auch ein (ε/ℓ) -NBP für f

Beweis.

- Sei D ein ε -Unterscheider für f , d.h. für alle $k \geq 0$ gilt

$$\Pr[D(f(\mathcal{U}_k)) = 1] - \Pr[D(\mathcal{U}_{\ell(k)}) = 1] \geq \varepsilon(\ell(k))$$

- Sei $\mathcal{B}_i$ das i -te Bit von $\mathcal{U}_{\ell(k)}$ und für $i = 1, \dots, \ell(k) + 1$ sei $\mathcal{H}_i$ die ZV

$$\mathcal{H}_i = f_1(\mathcal{U}_k) \cdots f_{i-1}(\mathcal{U}_k) \mathcal{B}_i \cdots \mathcal{B}_{\ell(k)} = f_{[i-1]}(\mathcal{U}_k) \mathcal{B}_i \cdots \mathcal{B}_{\ell(k)}$$

- Da $\mathcal{H}_{i+1}$ aus $\mathcal{H}_i$ entsteht, indem das Zufallsbit $\mathcal{B}_i$ durch das Pseudozufallsbit $f_i(\mathcal{U}_k)$ ersetzt wird, interpolieren $\mathcal{H}_1, \dots, \mathcal{H}_{\ell(k)+1}$ den Übergang von $\mathcal{H}_1 = \mathcal{B}_1 \cdots \mathcal{B}_{\ell(k)} = \mathcal{U}_{\ell(k)}$ zu $\mathcal{H}_{\ell(k)+1} = f_1(\mathcal{U}_k) \cdots f_{\ell(k)}(\mathcal{U}_k) = f(\mathcal{U}_k)$

Beweis (Fortsetzung)

- D ist also ein ε -Unterscheider zwischen $f(\mathcal{U}_k) = \mathcal{H}_{\ell(k)+1}$ und $\mathcal{U}_{\ell(k)} = \mathcal{H}_1$
- Tatsächlich ist D auch ein (ε/ℓ) -Unterscheider zwischen $\mathcal{H}_{\mathcal{I}+1}$ und $\mathcal{H}_{\mathcal{I}}$:

$$\begin{aligned}
 & \Pr[D(\mathcal{H}_{\mathcal{I}+1}) = 1] - \Pr[D(\mathcal{H}_{\mathcal{I}}) = 1] \\
 &= \sum_{i=1}^{\ell(k)} \underbrace{\Pr[\mathcal{I} = i]}_{1/\ell(k)} (\Pr[D(\mathcal{H}_{i+1}) = 1] - \Pr[D(\mathcal{H}_i) = 1]) \\
 &= (\Pr[D(\underbrace{\mathcal{H}_{\ell(k)+1}}_{f(\mathcal{U}_k)}) = 1] - \Pr[D(\underbrace{\mathcal{H}_1}_{\mathcal{U}_{\ell(k)}}) = 1]) / \ell(k) \\
 &\geq \varepsilon(\ell(k)) / \ell(k)
 \end{aligned}$$

- Betrachte nun folgenden NBP N :

```

1 input  $(y_1 \cdots y_{i-1}, 1^n)$  mit  $1 \leq i \leq n$ 
2   rate zufällig  $b_i, \dots, b_n \in_R \{0, 1\}$ 
3 output  $(D(y_1 \cdots y_{i-1} b_i \cdots b_n) \oplus b_i \oplus 1)$ 

```

Beweis (Fortsetzung)

- Der NBP $N(y_1 \cdots y_{i-1}, 1^n) = D(y_1 \cdots y_{i-1} \mathcal{B}_i \cdots \mathcal{B}_n) \oplus \mathcal{B}_i \oplus 1$ sagt also das nächste Bit y_i mit $\mathcal{B}_i$ vorher, falls $D(y_1 \cdots y_{i-1} \mathcal{B}_i \cdots \mathcal{B}_n)$ akzeptiert, und sonst mit $\mathcal{B}_i \oplus 1$
- Da $D(\mathcal{H}_{\mathcal{I}+1})$ mit größerer Wahrscheinlichkeit akzeptiert als $D(\mathcal{H}_{\mathcal{I}})$ und $\mathcal{H}_{i+1}$ anstelle des Zufallsbits $\mathcal{B}_i$ das Pseudozufallsbit $f_i(\mathcal{U}_k)$ enthält, deutet $D(\mathcal{H}_i) = 1$ darauf hin, dass das i -te Bit von $\mathcal{H}_i$ pseudozufällig ist
- Tatsächlich folgt nun die Aussage des Satzes aus folgender Behauptung

Behauptung.

$$\Pr[N(f_{[i-1]}(\mathcal{U}_k), 1^{\ell(k)}) = f_i(\mathcal{U}_k)] = 1/2 + \Pr[D(\mathcal{H}_{i+1}) = 1] - \Pr[D(\mathcal{H}_i) = 1]$$

$$\begin{aligned} \Pr[N(f_{[\mathcal{I}-1]}(\mathcal{U}_k), 1^{\ell(k)}) = f_{\mathcal{I}}(\mathcal{U}_k)] \\ = 1/2 + \underbrace{\Pr[D(\mathcal{H}_{\mathcal{I}+1}) = 1] - \Pr[D(\mathcal{H}_{\mathcal{I}}) = 1]}_{\geq \varepsilon(\ell(k))/\ell(k)} \quad (\text{nach obiger Beh.}) \end{aligned}$$



Behauptung.

$$\Pr[N(f_{[i-1]}(\mathcal{U}_k), 1^{\ell(k)}) = f_i(\mathcal{U}_k)] = 1/2 + \Pr[D(\mathcal{H}_{i+1}) = 1] - \Pr[D(\mathcal{H}_i) = 1]$$

Beweis.

Wegen $N(f_{[i-1]}(\mathcal{U}_k), 1^{\ell(k)}) = \underbrace{D(f_{[i-1]}(\mathcal{U}_k) \mathcal{B}_i \cdots \mathcal{B}_{\ell(k)})}_{\mathcal{H}_i} \oplus \mathcal{B}_i \oplus 1$ folgt

$$\begin{aligned} \Pr[N(f_{[i-1]}(\mathcal{U}_k), 1^{\ell(k)}) = f_i(\mathcal{U}_k)] &= \Pr[D(\mathcal{H}_i) \oplus \mathcal{B}_i \oplus 1 = f_i(\mathcal{U}_k)] \\ &= \underbrace{\Pr[D(\mathcal{H}_i) = 1 \wedge \mathcal{B}_i = f_i(\mathcal{U}_k)]}_{\Pr[\mathcal{B}_i = f_i(\mathcal{U}_k)] - \Pr[\mathcal{B}_i = f_i(\mathcal{U}_k) \wedge D(\mathcal{H}_i) = 0]} + \underbrace{\Pr[D(\mathcal{H}_i) = 0 \wedge \mathcal{B}_i \neq f_i(\mathcal{U}_k)]}_{\Pr[D(\mathcal{H}_i) = 0] - \Pr[D(\mathcal{H}_i) = 0 \wedge \mathcal{B}_i = f_i(\mathcal{U}_k)]} \\ &= \underbrace{\Pr[\mathcal{B}_i = f_i(\mathcal{U}_k)]}_{1/2} + \underbrace{\Pr[D(\mathcal{H}_i) = 0]}_{1 - \Pr[D(\mathcal{H}_i) = 1]} - \underbrace{2\Pr[D(\mathcal{H}_i) = 0 \wedge \mathcal{B}_i = f_i(\mathcal{U}_k)]}_{\Pr[D(\mathcal{H}_{i+1}) = 0 \wedge \mathcal{B}_i = f_i(\mathcal{U}_k)]} \\ &\quad = \underbrace{\Pr[D(\mathcal{H}_{i+1}) = 0]}_{1 - \Pr[D(\mathcal{H}_{i+1}) = 1]} \underbrace{\Pr[\mathcal{B}_i = f_i(\mathcal{U}_k)]}_{1/2} \\ &= 1/2 + \Pr[D(\mathcal{H}_{i+1}) = 1] - \Pr[D(\mathcal{H}_i) = 1] \end{aligned}$$



Quadratische Reste

- Als nächstes betrachten wir den BBS-Generator
- Dieser beruht auf dem Problem, die Lösbarkeit von quadratischen Kongruenzgleichungen zu entscheiden

Definition

- Ein Element $a \in \mathbb{Z}_m^*$ heißt **quadratischer Rest modulo m** (kurz: $a \in \text{QR}_m$), falls ein $x \in \mathbb{Z}_m^*$ mit $x^2 \equiv_m a$ existiert
- Die Menge $\text{QNR}_m := \mathbb{Z}_m^* \setminus \text{QR}_m$ enthält alle **quadratischen Nichtreste modulo m**
- Für eine Primzahl $p > 2$ und eine Zahl $a \in \mathbb{Z}$ heißt

$$\mathcal{L}(a, p) = \left(\frac{a}{p} \right) = \begin{cases} 1, & a \bmod p \in \text{QR}_p \\ -1, & a \bmod p \in \text{QNR}_p \\ 0, & \text{sonst} \end{cases}$$

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

- Die quadratische Kongruenz $x^2 \equiv_m a$ besitzt also für ein $a \in \mathbb{Z}_m^*$ genau dann eine Lösung, wenn $a \in \text{QR}_m$ ist
- Da mit $a, b \in \text{QR}_m$ auch $ab \in \text{QR}_m$ ist, bildet QR_m eine Untergruppe von $\mathbb{Z}_m^*$
- Wie das folgende Lemma zeigt, kann die Lösbarkeit von $x^2 \equiv_m a$ für primes m effizient entschieden werden

Lemma

- Sei $a \in \mathbb{Z}_p^*$, $p > 2$ prim, und sei g ein beliebiger Erzeuger von $\mathbb{Z}_p^*$
- Dann sind die folgenden drei Bedingungen äquivalent:
 - 1) $a \in \text{QR}_p$
 - 2) $a^{(p-1)/2} \equiv_p 1$
 - 3) $\log_{p,g}(a)$ ist gerade

Beweis.

1) $\Rightarrow$ 2): Ist $a \in \text{QR}_p$, d. h. $b^2 \equiv_p a$ für ein $b \in \mathbb{Z}_p^*$, so folgt mit dem Satz von Fermat

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

2) $\Rightarrow$ 3): Gilt $a \equiv_p g^k$ für ein ungerades $k = 2 \cdot j + 1$, so folgt

$$a^{(p-1)/2} \equiv_p g^{k(p-1)/2} \equiv_p g^{(p-1)j} g^{(p-1)/2} \equiv_p g^{(p-1)/2} \equiv_p -1 \not\equiv_p 1$$

3) $\Rightarrow$ 1): Ist $a \equiv_p g^k$ für $k = 2j$, so folgt $a \equiv_p (g^j)^2$, also $a \in \text{QR}_p$ □

- Somit zerfällt $\mathbb{Z}_p$ in die drei Teilmengen QR_p , QNR_p und $\mathbb{Z}_p \setminus \mathbb{Z}_p^* = \{0\}$
- Die beiden Teilmengen QR_p und QNR_p enthalten jeweils $(p-1)/2$ Elemente
- Zudem ist das Produkt ab von $a, b \in \mathbb{Z}_p^*$ genau dann in QR_p , wenn $a, b \in QR_p$ oder $a, b \in QNR_p$ sind
- Als weitere Folgerung erhalten wir folgende Formel zur effizienten Berechnung des Legendre-Symbols

Satz (Eulers Kriterium)

Für alle $a \in \mathbb{Z}$ und $p > 2$ prim gilt

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

Beweis.

- Es ist klar, dass diese Kongruenz im Fall $a \equiv_p 0$ gilt
- Nach obigem Lemma gilt sie auch im Fall $a \bmod p \in \text{QR}_p$, da dann $a^{(p-1)/2} \equiv_p 1 = \left(\frac{a}{p} \right)$ ist
- Es bleibt also der Fall, dass $a \bmod p \in \text{QNR}_p$ ist
- Da das Polynom $x^2 - 1$ in $\mathbb{Z}_p$ höchstens zwei Nullstellen hat und neben $x = 1$ nach dem Satz von Fermat auch $a^{(p-1)/2} \bmod p$ eine Nullstelle ist, muss $a^{(p-1)/2} \equiv_p \pm 1$ sein
- Daraus folgt nun $a^{(p-1)/2} \equiv_p -1$, da im Fall $a^{(p-1)/2} \equiv_p 1$ die Zahl $a \bmod p$ in QR_p und somit nicht in QNR_p wäre



Korollar

Für alle $a, b \in \mathbb{Z}$ und $p > 2$ prim gilt

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

- Als weiteres Korollar aus Eulers Kriterium erhalten wir eine Methode, quadratische Kongruenzgleichungen im Fall $p \equiv_4 3$ effizient zu lösen
- Im Fall $p \equiv_4 1$ ist dagegen kein effizienter deterministischer Lösungsalgorithmus bekannt
- Allerdings gibt es hierfür effiziente probabilistische Algorithmen (z.B. von Tonelli und Shanks)

Korollar

- Sei $p > 2$ prim, dann besitzt die quadratische Kongruenz $x^2 \equiv_p a \quad (*)$ für jedes $a \in \text{QR}_p$ in $\mathbb{Z}_p$ genau zwei Lösungen
- Im Fall $p \equiv_4 3$ sind dies $x_1 = a^k \bmod p$ und $x_2 = -a^k \bmod p$, wobei $k = (p+1)/4$ sowie $x_1 \in \text{QR}_p$ und $x_2 \in \text{QNR}_p$ ist

Beweis.

- Wegen $a \in \text{QR}_p$ hat $(*)$ eine Lösung $x_1 \in \mathbb{Z}_p^*$ und diese liefert eine zweite Lösung $x_2 = p - x_1 \in \mathbb{Z}_p^*$ mit $x_2 \neq x_1$ (p ist ungerade)
- Da $\mathbb{Z}_p$ ein Körper ist, existieren keine weiteren Lösungen in $\mathbb{Z}_p$
- Im Fall $p \equiv_4 3$ liefert Eulers Kriterium für $k = (p+1)/4$ die Kongruenz

$$(a^k)^2 = a^{(p+1)/2} = a^{(p-1)/2} a \equiv_p a, \text{ d.h. } x_1 = a^k \bmod p \text{ löst } (*)$$
- Mit a ist auch x_1 in QR_p und wegen $\left(\frac{-x_1}{p}\right) = \left(\frac{-1}{p}\right) \cdot \left(\frac{x_1}{p}\right) = -\left(\frac{x_1}{p}\right) = -1$ ist $x_2 = p - x_1$ in QNR_p



Satz

- Sei $n = pq$ für Primzahlen p, q mit $p \equiv_4 1$ $q \equiv_4 3$
- Dann besitzt die quadratische Kongruenz $x^2 \equiv_n a$ für jedes $a \in \text{QR}_n$ genau vier Lösungen, wovon genau eine ein quadratischer Rest ist

Beweis.

- Mit $x^2 \equiv_n a$ besitzen wegen $n = pq$ auch die beiden Kongruenzen $x^2 \equiv_p a$ und $x^2 \equiv_q a$ Lösungen, und zwar jeweils genau zwei

$$u_1 = a^{(p+1)/4} \bmod p \in \text{QR}_p \quad u_2 = -a^{(p+1)/4} \bmod p \in \text{QNR}_p$$

$$v_1 = a^{(q+1)/4} \bmod q \in \text{QR}_q \quad v_2 = -a^{(q+1)/4} \bmod q \in \text{QNR}_q$$
- Mit dem chinesischen Restsatz lässt sich für jedes Paar $(i, j) \in [2] \times [2]$ eine Lösung x_{ij} des folgenden Systems bestimmen

$$x \equiv_p u_i$$

$$x \equiv_q v_j$$

Quadratische Reste

Beweis (Fortsetzung).

- Die Kongruenz $x^2 \equiv_n a$ kann nicht mehr als diese vier Lösungen haben, da sonst für mindestens eine der beiden Kongruenzen $x^2 \equiv_p a$ und $x^2 \equiv_q a$ mehr als zwei Lösungen existieren würden
- Wegen

$$x_{ij} \in \text{QR}_n \Rightarrow \exists s: s^2 \equiv_n x_{ij} \Rightarrow s^2 \equiv_p u_i \wedge s^2 \equiv_q v_j \Rightarrow u_i \in \text{QR}_p \wedge v_j \in \text{QR}_q$$

können $x_{1,2}, x_{2,1}, x_{2,2}$ keine quadratischen Reste modulo n sein

- Da aber u_1 und v_1 quadratische Reste modulo p bzw. q sind, gibt es Zahlen $s \in \mathbb{Z}_p^*$ und $t \in \mathbb{Z}_q^*$ mit $s^2 \equiv_p u_1$ und $t^2 \equiv_q v_1$
- Folglich erfüllt die Lösung $w \in \mathbb{Z}_n^*$ des Systems

$$x \equiv_p s$$

$$x \equiv_q t$$

die Kongruenzen

$$w^2 \equiv_p s^2 \equiv_p u_1 \equiv_p x_{1,1} \quad \text{und} \quad w^2 \equiv_q t^2 \equiv_q v_1 \equiv_q x_{1,1}$$

und somit $w^2 \equiv_n x_{1,1}$, d.h. $x_{1,1} \in \text{QR}_n$



- Als eine für die Kryptografie interessante zahlentheoretische Funktion erhalten wir somit für jedes $n = pq$, wobei p und q Primzahlen mit $p \equiv_4 q \equiv_4 3$ sind, die **diskrete Quadratfunktion** $x \mapsto x^2 \bmod n$, die nach vorigem Satz eine Permutation auf QR_n ist
- Ihre Umkehrfunktion $x \mapsto \sqrt{x} \bmod n$ heißt **diskrete Quadratwurzelfunktion** auf QR_n
- Es ist bekannt, dass die effiziente Berechnung dieser Wurzelfunktion äquivalent zur effizienten Faktorisierung von n ist

- Der BBS-Pseudozufallsgenerator wurde 1986 von Blum, Blum und Shub vorgestellt und verwendet die Quadratfunktion

$$x^2 : \text{QR}_n \mapsto \text{QR}_n$$

mit $n = p \cdot q$ für p, q prim und $p \equiv_4 3$ $q \equiv_4 3$

- Seine Sicherheit beruht auf der Annahme, dass das Problem schwer ist, ohne Kenntnis der Primfaktoren von n für ein $x \in \mathbb{Z}_n^*$ zu entscheiden, ob $x \in \text{QR}_n$ ist
- Als Keim wird eine zufällig aus $\mathbb{Z}_n^*$ gewählte Zahl x_0 verwendet
- Dann ist $x_1 = x_0^2 \bmod n$ ein zufällig aus QR_n gewählter quadratischer Rest
- Beginnend mit x_1 wird durch wiederholtes Quadrieren eine Folge von Zahlen $x_i \in \text{QR}_n$ berechnet, deren Paritäten die Bits der Ausgabefolge liefern

Algorithmus $\text{BBS}_{n,\ell}(x_0)$

```
1 for  $i := 1$  to  $\ell$  do
2    $x_i := x_{i-1}^2 \bmod n$ 
3    $b_i := x_i \bmod 2$ 
4 output( $b_1, \dots, b_\ell$ )
```

Beispiel

Wählen wir z. B. die Primzahlen $p = 11$, $q = 19$, also $n = 209$, und als Keim $x_0 = 20$, so erhalten wir die Pseudo-Zufallsbitfolge $\text{BBS}_{209}(20) = 11001100 \dots$

i	0	1	2	3	4	5	6	7	8	...
x_i	20	191	115	58	20	191	115	58	20	...
b_i	0	1	1	0	0	1	1	0	0	...



Zum Nachweis der Sicherheit des BBS-Generators erweitern wir das Legendre-Symbol zum Jacobi-Symbol

Definition

- Das **Jacobi-Symbol** ist für alle a und alle ungeraden $m = p_1^{e_1} \cdots p_r^{e_r} \geq 3$ 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 $p_1 < \cdots < p_r$ die Primfaktoren von m sind

- Ein quadratischer Nichtrest $a \in \text{QNR}_m$ mit dem Jacobi-Symbol $\left(\frac{a}{m} \right) = 1$ wird als **quadratischer Pseudorest modulo m** bezeichnet (hierfür schreiben wir kurz $a \in \widetilde{\text{QR}}_m$)

- Im Gegensatz zum Legendre-Symbol muss also nicht jeder quadratische Nichtrest $a \in \text{QNR}_m$ das Jacobi-Symbol $\left(\frac{a}{m}\right) = -1$ haben
- Zum Beispiel gibt es in $\mathbb{Z}_n^*$ im Fall $n = p \cdot q$ für Primzahlen p und q mit $p \equiv_4 q \equiv_4 3$ genau $\varphi(n)/4$ quadratische Reste und genau $3\varphi(n)/4$ quadratische Nichtreste
- Dagegen gilt nur für die Hälfte aller $a \in \mathbb{Z}_n^*$ die Gleichung $\left(\frac{a}{n}\right) = -1$
- Folglich gibt es in diesem Fall genau so viele quadratische Reste wie quadratische Pseudoreste
- 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 (Quadratisches Reziprozitätsgesetz, Gauß)

Seien $m, n > 2$, ungerade und teilerfremd. Dann gilt

$$\left(\frac{n}{m}\right) \cdot \left(\frac{m}{n}\right) = (-1)^{(m-1)(n-1)/4}$$

Satz

Für ungerades m gilt

$$\left(\frac{2}{m}\right) = (-1)^{\frac{m^2-1}{8}}$$

- Man beachte, dass $\frac{m^2-1}{8}$ genau dann gerade ist, wenn $m \equiv_8 1$ oder $m \equiv_8 7$ gilt
- Zudem ist $(m-1)(n-1)/4$ genau dann gerade, wenn $m \equiv_4 1$ oder $n \equiv_4 1$ gilt

Quadratische Pseudoreste

Korollar

Seien a und m gegeben mit $m \geq 3$ ungerade und $\text{ggT}(a, m) = 1$; dann lässt sich $\left(\frac{a}{m}\right)$ durch einen Algorithmus der Zeitkomplexität $O(n^3)$ berechnen

Beweis.

Dies folgt ähnlich wie beim euklidischen Algorithmus aus folgenden Gleichungen

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

□

Beispiel. Das Jacobi-Symbol von 73 modulo 83 ist

$$\left(\frac{73}{83}\right) = \left(\frac{10}{73}\right) \underbrace{(-1)^{82 \cdot 72 / 4}}_{=1} = \left(\frac{2}{73}\right) \left(\frac{5}{73}\right) = \left(\frac{3}{5}\right) \underbrace{(-1)^{72 \cdot 4 / 4}}_{=1} = \left(\frac{2}{3}\right) = -1$$

◀