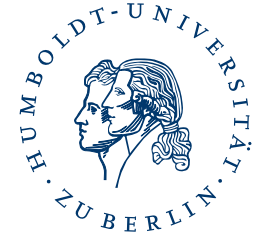
Kryptologie

Johannes Köbler



Institut für Informatik
Humboldt-Universität zu Berlin

WS 2020/21

Pseudozufallszahlen-Generatoren

- 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 gewählt und die Ausgabe $f(x)$ sollte länger sein als x und möglichst zufällig aussehen
- Zudem sollte f von einem deterministischen Algorithmus effizient berechenbar sein

Linear-Kongruenz-Generator

Beispiel

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

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

```
1 for i := 1 to l do
2     x_i := a x_{i-1} + b mod n
3     b_i := x_i mod 2
4 output(b_1 ... b_l)
```

Power-Generator

Beispiel

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

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

```
1 for i := 1 to l do
2     x_i := x_{i-1}^e mod n
3     b_i := x_i mod 2
4 output(b_1 ... 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 $\ell : \mathbb{N} \to \mathbb{N}$ 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

Definition

- Seien (X_k) und (Y_k), $k \geq 0$, Familien von Zufallsvariablen mit Wertebereich $W(X_k), W(Y_k) \subseteq \{0,1\}^{\ell(k)}$ und sei $\varepsilon : \mathbb{N} \to \mathbb{R}^+$ eine Funktion
- Ein ε-Unterscheider zwischen (X_k) und (Y_k) ist ein effizienter probabilistischer Algorithmus D mit:

$$\Pr[D(X_k) = 1] - \Pr[D(Y_k) = 1] \geq \varepsilon(\ell(k))$$

- Hierbei ist $\Pr[D(X_k) = 1]$ die Wahrscheinlichkeit, dass D bei einer zufällig gemäß X_k gewählten Eingabe akzeptiert (bzw. 1 ausgibt)
- In diesem Fall heißen die beiden Familien (X_k) und (Y_k) ε-unterscheidbar
- Ein $\ell(k)$-Generator f heißt ε-unterscheidbar, falls die beiden Familien $(f(U_k))$ und $(U_{\ell(k)})$ von Zufallsvariablen ε-unterscheidbar sind, wobei U_n eine auf $\{0,1\}^n$ gleichverteilte Zufallsvariable ist

Definition (Fortsetzung)

- Eine Funktion $\varepsilon : \mathbb{N} \to \mathbb{R}$ 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 $\varepsilon : \mathbb{N} \to \mathbb{R}^+$ ε-unterscheidbar ist

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

$$\Pr[D(f(U_k)) = 1] - \Pr[D(U_{\ell(k)}) = 1] \geq 1/p(\ell(k))$$

  ist
- 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 ··· y_{k+1} ∈ {0,1}^{k+1}
2 output(y_1)
```

- Dann gilt $\Pr[D(f(U_k)) = 1] = 1$ und $\Pr[D(U_{k+1}) = 1] = 1/2$ und somit

$$\Pr[D(f(U_k)) = 1] - \Pr[D(U_{k+1}) = 1] = 1/2$$

  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 ◁

Kryptografische Sicherheit von Pseudozufallsgeneratoren

- Es ist nicht bekannt, ob kryptografisch sichere PZGen existieren
- Eine notwendige Bedingung hierfür ist $\mathrm{P} \neq \mathrm{NP}$, da $\mathrm{P} = \mathrm{NP}$ 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
- Für $i \in \{1, \ldots, \ell(k)\}$ bezeichne $f_i(x)$ das i-te Bit und für $i \in \{0, \ldots, \ell(k)\}$ bezeichne $f_{[i]}(x)$ die Folge der ersten i Bits von $f(x)$
- Ein next bit predictor (NBP) N für f ist ein effizienter probabilistischer Algorithmus, der bei jeder Eingabe $(v, 1^{\ell(k)})$ mit $v \in \{0,1\}^{i-1}$ für ein $i \in \{1, \ldots, \ell(k)\}$ ein Bit $N(v, 1^{\ell(k)})$ ausgibt
- N heißt ε-next bit predictor (ε-NBP) für f, falls für alle k gilt:

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

wobei die Zufallsvariable I auf der Menge $\{1, \ldots, \ell(k)\}$ 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** $(v, 1^n)$ mit $v = v_1 \cdots v_{i-1} \in \{0,1\}^{i-1}$ für ein $i \in \{1, \ldots, n\}$
2 **output**(1)

- Dann gilt

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

- Somit gilt

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

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

Satz. Sei f ein $\ell(k)$-Generator und sei $\varepsilon : \mathbb{N} \to \mathbb{R}^+$ 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 v = v_1 ··· v_n
2    wähle i ∈_R {1, ..., n}
3 output(N(v_1 ··· v_{i−1}, 1^n) ⊕ v_i ⊕ 1)
```

- D gibt also bei Eingabe $v = v_1 \cdots v_n$ genau dann 1 aus, wenn der Prediktor N bei Eingabe $(v_1 \cdots v_{i-1}, 1^n)$ das i-te Bit von v richtig rät, wobei i zufällig aus $\{1, \ldots, n\}$ gewählt wird
- Daher gilt für alle $k \geq 0$,

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

wobei I eine auf $\{1, \ldots, \ell(k)\}$ gleichverteilte Zufallsvariable ist

Beweis (Fortsetzung)

- Andererseits ist klar, dass für jeden NBP N

$$\Pr[N(B_1, \ldots, B_{l-1}, 1^{\ell(k)}) = B_l] = 1/2,$$

ist, wobei $B_1, \ldots, B_{\ell(k)}$ unabhängig und gleichverteilt auf $\{0,1\}$ sind

- Folglich gilt wegen $U_{\ell(k)} = B_1 \ldots B_{\ell(k)}$

$$\Pr[D(U_{\ell(k)}) = 1] = \Pr[N(B_1, \ldots, B_{l-1}, 1^{\ell(k)}) = B_l] = 1/2$$

und es folgt

$$\underbrace{\Pr[D(f(U_k)) = 1]}_{\geq 1/2 + \varepsilon(\ell(k))} - \underbrace{\Pr[D(U_{\ell(k)}) = 1]}_{=1/2} \geq \varepsilon(\ell(k)),$$

- D ist also ein $\varepsilon(\ell(k))$-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,

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

wobei I eine auf $\{1, \ldots, \ell(k)\}$ gleichverteilte Zufallsvariable 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. es gilt

$$\Pr[D(f(U_k)) = 1] - \Pr[D(U_{\ell(k)}) = 1] \geq \varepsilon(\ell(k))$$

  für alle $k \geq 0$
- Die Ausgabe $D(y) = 1$ deutet also darauf hin, dass y tendenziell ein Pseudozufallsstring ist, während die Ausgabe $D(y) = 0$ darauf hindeutet, dass y ein echter Zufallsstring ist

Beweis (Fortsetzung)

- Betrachte folgenden probabilistischen Algorithmus N

---

1 **input** $(v_1 \cdots v_{i-1}, 1^n)$ mit $1 \leq i \leq n$
2 rate zufällig $b_i, \ldots, b_n \in_R \{0,1\}$
3 **output**$(D(v_1 \cdots v_{i-1} b_i \cdots b_n) \oplus b_i \oplus 1)$

- N sagt also das i-te Bit v_i mit b_i vorher, falls $D(v_1 \cdots v_{i-1} b_i \cdots b_n) = 1$ ist (also D seine Eingabe $v_1 \cdots v_{i-1} b_i \cdots b_n$ für pseudozufällig hält), und sonst mit $b_i \oplus 1$
- Betrachte für $i = 1, \ldots, \ell(k) + 1$ die Zufallsvariablen

$$H_i = f_{[i-1]}(U_k) B_i \cdots B_{\ell(k)},$$

wobei U_k, $B_i, \ldots, B_{\ell(k)}$ unabhängig und gleichverteilt auf $\{0,1\}^k$ bzw. auf $\{0,1\}$ sind

- Insbesondere ist also $H_1 = B_1 \cdots B_{\ell(k)} = U_{\ell(k)}$ gleichverteilt auf $\{0,1\}^{\ell(k)}$ und $H_{\ell(k)+1} = f(U_k)$ pseudozufällig verteilt auf $\{0,1\}^{\ell(k)}$

Behauptung

Es gilt

$$\Pr[N(f_{[i-1]}(U_k), 1^{\ell(k)}) = f_i(U_k)] = 1/2 + \Pr[D(H_{i+1}) = 1] - \Pr[D(H_i) = 1]$$

Beweis.

Wegen $N(f_{[i-1]}(U_k), 1^{\ell(k)}) = D(\underbrace{f_{[i-1]}(U_k) B_i \cdots B_{\ell(k)}}_{H_i}) \oplus B_i \oplus 1$ folgt

$$\begin{aligned}
\Pr[N(f_{[i-1]}(U_k), 1^{\ell(k)}) = f_i(U_k)] &= \Pr[D(H_i) \oplus B_i \oplus 1 = f_i(U_k)] \\
&= \underbrace{\Pr[D(H_i) = 1 \wedge B_i = f_i(U_k)]}_{\Pr[B_i = f_i(U_k)] - \Pr[B_i = f_i(U_k) \wedge D(H_i) = 0]} + \underbrace{\Pr[D(H_i) = 0 \wedge B_i \neq f_i(U_k)]}_{\Pr[D(H_i) = 0] - \Pr[D(H_i) = 0 \wedge B_i = f_i(U_k)]} \\
&= \underbrace{\Pr[B_i = f_i(U_k)]}_{1/2} + \underbrace{\Pr[D(H_i) = 0]}_{1 - \Pr[D(H_i) = 1]} - 2\underbrace{\Pr[D(H_i) = 0 \wedge B_i = f_i(U_k)]}_{\Pr[D(H_{i+1}) = 0 \wedge B_i = f_i(U_k)]} \\
&\qquad \underbrace{= \Pr[D(H_{i+1}) = 0]}_{1 - \Pr[D(H_{i+1}) = 1]} \underbrace{\Pr[B_i = f_i(U_k)]}_{1/2} \\
&= 1/2 + \Pr[D(H_{i+1}) = 1] - \Pr[D(H_i) = 1]
\end{aligned}$$

□

Beweis (Schluss)

- Sei I eine auf $\{1, \ldots, \ell(k)\}$ gleichverteilte Zufallsvariable
- Dann folgt

$$
\begin{aligned}
&\Pr[N(f_{[I-1]}(U_k), 1^{\ell(k)}) = f_I(U_k)] \\
&= 1/2 + \Pr[D(H_{I+1}) = 1] - \Pr[D(H_I) = 1] \text{ (nach obiger Beh.)} \\
&= 1/2 + \sum_{i=1}^{\ell(k)} \underbrace{\Pr[I = i]}_{1/\ell(k)} (\Pr[D(H_{i+1}) = 1] - \Pr[D(H_i) = 1]) \\
&= 1/2 + (\Pr[D(\underbrace{H_{\ell(k)+1}}_{f(U_k)}) = 1] - \Pr[D(\underbrace{H_1}_{U_{\ell(k)}}) = 1])/\ell(k) \\
&= 1/2 + (\underbrace{\Pr[D(f(U_k)) = 1] - \Pr[D(U_{\ell(k)}) = 1]}_{\geq \varepsilon(\ell(k))})/\ell(k) \\
&\geq 1/2 + \varepsilon(\ell(k))/\ell(k)
\end{aligned}
$$

□

Ganz ähnlich wie der obige Satz lässt sich auch folgendes Resultat beweisen

Satz

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

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

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

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

Quadratische Reste

- Die quadratische Kongruenz $x^2 \equiv_m a$ besitzt also für ein $a \in \mathbb{Z}_m^*$ genau dann eine Lösung, wenn $a \in \mathrm{QR}_m$ ist
- Da mit $a, b \in \mathrm{QR}_m$ auch $ab \in \mathrm{QR}_m$ 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

Quadratische Reste

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) $a \in \mathrm{QR}_p$
  2) $a^{(p-1)/2} \equiv_p 1$
  3) $\log_{p,g}(a)$ ist gerade

Beweis.

1) $\Rightarrow$ 2): Ist $a \in \mathrm{QR}_p$, 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 $a \in \mathrm{QR}_p$ $\square$

Quadratische Reste

- 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 $a, b \in \mathrm{QR}_p$ oder $a, b \in \mathrm{QNR}_p$ 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 $a \bmod p \in \mathrm{QR}_p$, da dann $a^{(p-1)/2} \equiv_p 1 = \left(\frac{a}{p}\right)$ ist
- Es bleibt also der Fall, dass $a \bmod p \in \mathrm{QNR}_p$ 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 □

Quadratische Reste

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)

Quadratische Reste

Korollar

- Sei $p > 2$ prim, dann besitzt die quadratische Kongruenzgleichung $x^2 \equiv_p a$ für jedes $a \in \mathrm{QR}_p$ in $\mathbb{Z}_p$ genau zwei Lösungen
- Im Fall $p \equiv_4 3$ sind dies $\pm a^k \bmod p$ (für $k = (p+1)/4$), wovon nur $a^k \bmod p$ ein quadratischer Rest ist

Beweis.

- Da $a \in \mathrm{QR}_p$ ist, existiert ein $b \in \mathbb{Z}_p^*$ mit $b^2 \equiv_p a$
- Mit b ist auch $-b$ Lösung von $x^2 \equiv_p a$ mit $-b \not\equiv_p b$ (p ist ungerade)
- Da $\mathbb{Z}_p$ ein Körper ist, existieren keine weiteren Lösungen
- Im Fall $p \equiv_4 3$ liefert Eulers Kriterium für $k = (p+1)/4$ die Kongruenz

$$(a^k)^2 = a^{(p+1)/2} = a^{(p-1)/2} \cdot a \equiv_p a$$

- Da mit a auch $a^k \bmod p \in \mathrm{QR}_p$ ist, folgt

$$\left(\frac{-a^k}{p}\right) = \left(\frac{-1}{p}\right) \cdot \left(\frac{a^k}{p}\right) = -\left(\frac{a^k}{p}\right) = -1$$

- Also ist $-a^k \bmod p$ ein quadratischer Nichtrest □

Quadratische Reste

Satz

- Sei $n = pq$ für Primzahlen p, q mit $p \equiv_4 q \equiv_4 3$
- Dann besitzt die quadratische Kongruenz $x^2 \equiv_n a$ für jedes $a \in \mathrm{QR}_n$ 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

$$u_1 = a^{(p+1)/4} \bmod p \in \mathrm{QR}_p \qquad u_2 = -a^{(p+1)/4} \bmod p \in \mathrm{QNR}_p$$
$$v_1 = a^{(q+1)/4} \bmod q \in \mathrm{QR}_q \qquad v_2 = -a^{(q+1)/4} \bmod q \in \mathrm{QNR}_q$$

- 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

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

  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. $x_{1,1} \in \mathrm{QR}_n$ □

Quadratische Reste

- Als weitere für die Kryptografie interessante zahlentheoretische Funktionen erhalten wir somit für jedes $n = pq$, wobei p, 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-Generator

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

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

  mit $n = p \cdot q$ für p, q prim und $p \equiv_4 q \equiv_4 3$
- 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 $x \in \mathrm{QR}_n$ 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 $x_i \in \mathrm{QR}_n$ berechnet, deren Paritäten die Bits der Ausgabefolge liefern

Der BBS-Generator

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

```
1 for i := 1 to ℓ do
2     x_i := x_{i-1}^2 mod n
3     b_i := x_i mod 2
4 output(b_1, ... b_ℓ)
```

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 $\mathrm{BBS}_{209}(20) = 11001100\ldots$

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	…

◁

Quadratische Pseudoreste

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
- Ist zwar $\left(\frac{a}{m}\right) = 1$, aber $a \in \mathrm{QNR}_m$ kein quadratischer Rest modulo m, so heißt a ***quadratischer Pseudorest modulo*** m (kurz: $a \in \widetilde{\boldsymbol{QR}}_m$)

Quadratische Pseudoreste

- Man beachte, dass im Gegensatz zum Legendre-Symbol die Eigenschaft $(\frac{a}{m}) = 1$ für ein $a \in \mathbb{Z}_m^*$ nicht immer mit $a \in \mathrm{QR}_m$ gleichbedeutend ist
- Zum Beispiel gibt es in $\mathbb{Z}_n^*$ ($n = p \cdot q$ für Primzahlen p und q mit $p \equiv_4 q \equiv_4 3$) wie wir gesehen haben, genau $\varphi(n)/4$ quadratische Reste und $3\varphi(n)/4$ quadratische Nichtreste
- Dagegen gilt nur für die Hälfte aller $a \in \mathbb{Z}_n^*$ die Gleichung $(\frac{a}{m}) = -1$
- 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

Quadratische Pseudoreste

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 $\mathrm{ggT}(a, m) = 1$; 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 den 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

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

◁

Quadratische Pseudoreste

- Sei $n = pq$ das Produkt zweier Primzahlen p, q mit $p \equiv_4 q \equiv_4 3$
- Wie bereits erwähnt, ist das Finden einer Wurzel für ein gegebenes $x \in \mathbb{Z}_n^*$ genau so schwer wie die Faktorisierung von n
- Tatsächlich wird bereits das zugehörige Entscheidungsproblem, ob eine gegebene Zahl $x \in \mathbb{Z}_n^*$ eine Wurzel hat (also ein quadratischer Rest ist), als schwierig betrachtet
- Da dieses Problem für Eingaben x mit Jacobisymbol $\left(\frac{x}{n}\right) = -1$ trivial ist, werden sie nicht als Eingaben zugelassen

Quadratische-Reste-Problem (QR-Problem):

Gegeben: Zahlen n und $x \in \mathbb{Z}_n^*$ mit Jacobisymbol $\left(\frac{x}{n}\right) = 1$, wobei n das Produkt zweier unbekannter Primzahlen ist

Gefragt: Ist $x \in \mathrm{QR}_n$?

Beim QR-Problem geht es also darum, quadratische Reste von quadratischen Pseudoresten zu unterscheiden

Sicherheit des BBS-Generators

- Wir zeigen nun, dass sich aus jedem effizienten Unterscheider für den BBS-Generator ein effizienter probabilistischer Algorithmus für das QR-Problem gewinnen lässt
- Im Umkehrschluss bedeutet dies, dass der BBS-Generator sicher ist, falls das QR-Problem hart ist
- Sei also D ein effizienter ε-Unterscheider für den Generator $\mathtt{BBS}_{n,\ell}$
- Dann ex. ein effizienter (ε/ℓ)-PBP P für $\mathtt{BBS}_{n,\ell}$
- Der folgende Satz zeigt, wie sich aus einem δ-PBP P für $\mathtt{BBS}_{n,\ell}$ ein probabilistischer Algorithmus gewinnen lässt, der das QR-Problem bei einer zufällig gewählten Eingabe $x \in_R \mathrm{QR}_n \cup \widetilde{\mathrm{QR}}_n$ mit einem Vorteil von δ korrekt entscheidet

Satz

Falls es einen δ-PBP für den Generator $\mathtt{BBS}_{n,\ell}$ gibt, so lässt sich für ein zufälliges $x \in_R \mathrm{QR}_n \cup \widetilde{\mathrm{QR}}_n$ mit Wahrscheinlichkeit $\geq 1/2 + \delta$ korrekt entscheiden, ob $x \in \mathrm{QR}_n$ ist

Sicherheit des BBS-Generators

Beweis.

- Sei P ein δ-PBP für den Generator $\mathtt{BBS}_{n,\ell}$
- Betrachte folgenden Entscheidungsalgorithmus für das QR-Problem:

Algorithmus $\mathtt{QR\text{-}Test}(x, n)$

1 wähle $i \in_R \{1, \ldots, \ell\}$
2 $x_i := x$
3 **for** $j := i + 1$ **to** ℓ **do**
4 $\quad x_j := x_{j-1}^2 \bmod n$
5 $\quad b_j := x_j \bmod 2$
6 $b_i := P(b_{i+1} \cdots b_\ell, 1^\ell)$
7 **if** $x \equiv_2 b_i$ **then output**(1) **else output**(0)

- Dann folgt die Aussage des Satzes unmittelbar aus folgender Behauptung

Sicherheit des BBS-Generators

Behauptung

$\Pr_{x \in_R \mathrm{QR}_n \cup \widetilde{\mathrm{QR}}_n}[\texttt{QR-Test}(x, n) = 1 \Leftrightarrow x \in \mathrm{QR}_n] \geq 1/2 + \delta$

Beweis.

- Wird x zufällig aus $\mathrm{QR}_n \cup \widetilde{\mathrm{QR}}_n$ gewählt, so ist $x_{i+1} = x^2 \bmod n$ ein zufälliger quadratischer Rest in QR_n
- Die Eingabe für den PBP P besteht also aus $\ell - i$ konsekutiven Bits $b_{i+1} \cdots b_\ell$ einer mit $\mathrm{BBS}_{n,\ell}$ generierten Pseudozufallsfolge
- Daher liefert die Ausgabe b_i von $P(b_{i+1} \cdots b_\ell, 1^\ell)$ in Zeile 6 mit Wahrscheinlichkeit $1/2 + \delta$ die Parität der diskreten Wurzel $\sqrt{x_{i+1}}$ von x_{i+1}
- Da $x \in \mathrm{QR}_n \cup \widetilde{\mathrm{QR}}_n$ und $x_{i+1} = x^2 \bmod n$ ist, gilt $\sqrt{x_{i+1}} \in \{x, n - x\}$
- Zudem hat $\sqrt{x_{i+1}}$ wegen $x \not\equiv_2 n - x$ genau dann die gleiche Parität wie x, wenn $x = \sqrt{x_{i+1}}$ ist
- Da dies wiederum mit $x \in \mathrm{QR}_n$ äquivalent ist, folgt die Behauptung $\square$

Sicherheit des BBS-Generators

Als nächstes zeigen wir, wie sich `QR-Test` in einen Algorithmus verwandeln lässt, der jede Eingabe $x \in \mathrm{QR}_n \cup \widetilde{\mathrm{QR}}_n$ mit Wahrscheinlichkeit $\geq 1/2 + \delta$ korrekt entscheidet

Satz

Falls es einen effizienten Algorithmus A gibt, der für eine zufällig aus $\mathrm{QR}_n \cup \widetilde{\mathrm{QR}}_n$ gewählte Eingabe x das QR-Problem mit Vorteil δ entscheidet, so ex. auch ein effizienter Algorithmus A', der dies für jede Eingabe $x \in \mathrm{QR}_n \cup \widetilde{\mathrm{QR}}_n$ tut

Sicherheit des BBS-Generators

Beweis (Fortsetzung).

- Betrachte folgenden Entscheidungsalgorithmus:

Algorithmus $A'(x, n),\ x \in \mathrm{QR}_n \cup \widetilde{\mathrm{QR}}_n$

1 wähle zufällig eine Zahl $z \in_R \mathbb{Z}_n^*$
2 wähle zufällig ein Bit $b \in_R \{0, 1\}$
3 $x' := (-1)^b z^2 x \bmod n$
4 **output** $A(x', n) \oplus b$

- Da $-1 \in \mathrm{QNR}_p \cap \mathrm{QNR}_q$ ist, folgt $-1 \in \widetilde{\mathrm{QR}}_n$
- Daher ist $x \mapsto -x \bmod n$ eine Bijektion zwischen QR_n und $\widetilde{\mathrm{QR}}_n$
- Zudem ist z^2 für $z \in_R \mathbb{Z}_n^*$ gleichverteilt auf QR_n
- Daher ist x' für jedes $x \in \mathrm{QR}_n \cup \widetilde{\mathrm{QR}}_n$ gleichverteilt auf $\mathrm{QR}_n \cup \widetilde{\mathrm{QR}}_n$ und $z^2 x \bmod n$ ist genau dann ein quadratischer Rest, wenn dies für x gilt
- Folglich ist die Ausgabe $A'(x, n) = A(x', n) \oplus b$ genau dann korrekt, wenn die Ausgabe $A(x', n)$ korrekt ist

□

Sicherheit des BBS-Generators

- Schließlich zeigen wir noch, wie sich die Fehlerwahrscheinlichkeit von A' exponentiell klein machen lässt
- Hierzu benötigen wir das folgende Lemma

Lemma

- Sei E ein Ereignis, das mit Wahrscheinlichkeit $1/2 - \delta$, $\delta > 0$, auftritt
- Dann ist die Wahrscheinlichkeit, dass sich E bei $m = 2t + 1$ unabhängigen Wiederholungen mehr als t-mal ereignet, kleiner als $1/2(1 - 4\delta^2)^t$

Beweis.

- Für $i = 1, \ldots, m$ sei X_i die Indikatorvariable

$$X_i = \begin{cases} 1, & \text{Ereignis } E \text{ tritt beim } i\text{-ten Versuch ein,} \\ 0, & \text{sonst} \end{cases}$$

und X sei die Zufallsvariable $X = \sum_{i=1}^{m} X_i$

Sicherheit des BBS-Generators

Beweis (Fortsetzung).

- Dann ist X binomial verteilt mit Parametern m und $p = 1/2 - \delta$
- Folglich gilt für $i > m/2$,

$$\begin{aligned}
\Pr[X = i] &= \binom{m}{i} (1/2 - \delta)^i (1/2 + \delta)^{m-i} \\
&= \binom{m}{i} (1/2 - \delta)^{m/2} (1/2 + \delta)^{m/2} \underbrace{\left(\frac{1/2 - \delta}{1/2 + \delta}\right)}_{<1}^{i-m/2} \\
&< \binom{m}{i} \underbrace{(1/2 - \delta)^{m/2} (1/2 + \delta)^{m/2}}_{(1/4 - \delta^2)^{m/2}}
\end{aligned}$$

Sicherheit des BBS-Generators

Beweis (Schluss).

- Somit erhalten wir

$$\sum_{i=t+1}^{m} \Pr[X=i] < \left(1/4-\delta^2\right)^{m/2} \underbrace{\sum_{i=t+1}^{m} \binom{m}{i}}_{=2^m/2=4^{m/2}/2} = \frac{(1-4\delta^2)^{m/2}}{2}$$

$$< \frac{(1-4\delta^2)^t}{2}$$

□

- Falls wir also A' $m=(2t+1)$-mal ausführen und einen Mehrheitsentscheid treffen, so reduziert sich die Fehlerwahrscheinlichkeit wegen $1-x<e^{-x}$ für $x>0$ von $1/2-\delta$ auf einen Wert kleiner $(1-4\delta^2)^t/2 < e^{-4\delta^2 t}/2 < e^{-4\delta^2 t}$
- Wählen wir beispielsweise $t=s/4\delta^2$, so wird diese kleiner als 2^{-s}