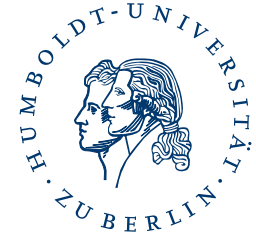
Kryptologie

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 in Polynomialzeit 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 in Polynomialzeit berechenbarer 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

Kryptografische Sicherheit von Pseudozufallsgeneratoren

Definition

- 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
- 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 in Polynomialzeit berechenbaren probabilistischen Algorithmus 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

Beispiel

- Betrachte folgenden Unterscheider D für den $(k+1)$-Generator f mit $f(x) = 1x$ 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 $\Pr[D(f(U_k)) = 1] = 1$ 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 probabilistischer Algorithmus N heißt next bit predictor (NBP) für f, falls N bei jeder Eingabe $(v, 1^{\ell(k)})$ mit $k \geq 0$ und $v \in \{0,1\}^{i-1}$ 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 $(k+1)$-Generator f mit $f(x) = 1x$ für alle $x \in \{0,1\}^*$:

```
1 input (v, 1^n) mit v = v_1 ··· v_{i-1} ∈ {0,1}^{i-1} für ein i ∈ {1,...,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

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

- N ist also ein $(1/2n)$-NBP für f ◁

Satz. Sei f ein $\ell(k)$-Generator.

Falls es einen ε-NBP N 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 jeder NBP für einen rein zufällig gewählten String $v \in_R \{0,1\}^{\ell(k)}$ das i-te Bit v_i von v bei Eingabe $(v_{[i-1]}, 1^{\ell(k)})$ genau mit Wahrscheinlichkeit $1/2$ richtig rät und somit $\Pr[D(U_{\ell(k)}) = 1] = 1/2$ ist
- Daher folgt

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

□

Definition

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

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

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 N für f gibt, so ex. auch ein ε-Unterscheider für f

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

Satz

Falls es einen ε-Unterscheider D 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 $D(y) = 0$ deutet also darauf hin, dass y tendenziell ein echter Zufallsstring ist, während die Ausgabe $D(y) = 1$ darauf hindeutet, dass y ein Pseudozufallsstring ist

Beweis (Fortsetzung)

- Betrachte folgenden probabilistischen Algorithmus N

```
1 input (v_1 ··· v_{i-1}, 1^n) mit 1 ≤ i ≤ n
2    rate zufällig b_i, ..., b_n ∈_R {0,1}
3 output(D(v_1 ··· v_{i-1} b_i ··· b_n) ⊕ b_i ⊕ 1)
```

- N sagt also das i-te Bit v_i mit b_i vorher, falls D den String $v_1 \cdots v_{i-1} b_i \cdots b_l$ für pseudozufällig hält (also $D(v_1 \cdots v_{i-1} b_i \cdots b_l) = 1$ ist), 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. $\{0,1\}$ 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

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