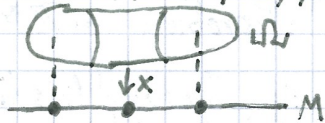


CRYPTO2-Übersicht: VL08+09: Zufall

VL08: Definition PRG, Distinguisher, Predictor; VL09: Blum-Blum-Shub & Beweis (b.w.)

- Recap Wk.rechnung: Wk.raum: $\Omega = \{\omega_1, \dots, \omega_n\}$ Wk.verteilg.: $P: \Omega \rightarrow [0,1] \in \mathbb{R}$
 VL: mit $\sum_{\omega \in \Omega} P[\omega] = 1$. Wenn $A \subseteq \Omega$ Ereignis, $P[A] = \sum_{\omega \in A} P[\omega]$
 Gleichförmige Wk.verteilg.: $U_n: \Omega \rightarrow [0,1], \omega \mapsto 1/n$
 de.wiki: Zufallsvariable = messbare Fkt. v. einem Wk.raum in einen Messraum
 VL: Zufallsvar. $X: \Omega \rightarrow M$ (M Menge) induziert auf M eine
 Wk.verteilg. $P_X: P_X[m] := P[X^{-1}(m)]$ „von X induzierte Wk.ver.“
 de.wiki: Bsp. Würfelsumme zweier Würfelwürfe:
 $S: \Omega \rightarrow \mathbb{N}; (x,y) \mapsto x+y$, wobei $\Omega = \{(1,1), \dots, (6,6)\}$
 VL: 

- Def. **E-Distinguisher**: Seien $X: \Omega_x \rightarrow M$ und $Y: \Omega_y \rightarrow M$ zwei Zufallsvar., $\epsilon > 0$.
 Ein ϵ -Distinguisher ist eine in **Polynomialzeit** berechenbare
 Abbildung $D: M \rightarrow \{0,1\}$ mit $|P[D(X)=1] - P[D(Y)=1]| \geq \epsilon$
 wobei $P[D(X)=1] := P_X[D^{-1}(1)] = P[X^{-1}(D^{-1}(1))]$
 D.h. die Wk., dass D auf von X generierten Elem. aus M 1 returned ist um ϵ größer/kleiner als für Y.
 wähle per X zufällig Elemente aus M und setze in D ein
 D darf dabei ausdrücklich auch eine Zufallskomponente haben!

- Def. **PRG/Pseudo Random Generator**: $G^L: \mathbb{B}^L \rightarrow \mathbb{B}^{L+s(L)}$ mit $s(L) \geq 1$.
 „Saatk“ „Stretch“
 Schön wäre es, wenn die Ergebnisse gleichförmig verteilt wären, d.h. $G(\mathbb{B}^L)$ nicht von $U_{\mathbb{B}^{L+s(L)}}$ zu unterscheiden wäre, doch das ist unmöglich! Ausweg:
 Fordere stattdessen, dass $G(\mathbb{B}^L)$ nicht mit **polynomiellen Aufwand** von $U_{\mathbb{B}^{L+s(L)}}$ zu unterscheiden ist! **Formal**: Zu jedem $\epsilon > 0$ gibt es ein L , sodass für alle $L > L$ G^L und $U_{\mathbb{B}^{L+s(L)}}$ nicht ϵ -unterscheidbar sind.
 Die Def. eines ϵ -Distinguishers eingesetzt bedeutet dies: Für alle polynomiellen $D: \mathbb{B}^{L+s(L)} \rightarrow \{0,1\}$ ist $|P[D(G^L)=1] - P[D(U_{\mathbb{B}^{L+s(L)}})=1]| < \epsilon$.

- Def. **(i, ϵ)-Predictor**: Sei X Zufallsvariable auf \mathbb{B}^L . Schreibe X komponentenweise: $X = X_1 \dots X_L$ (X_i = Komponente von X). Sei $\epsilon > 0$ und $2 \leq i \leq L$.
 Ein (i, ϵ) -Predictor ist eine Abbildung $P: \mathbb{B}^{i-1} \rightarrow \mathbb{B}$ mit $P[X_1 \dots X_{i-1}] = X_i \geq \frac{1}{2} + \epsilon$.
 D.h. ein Predictor kann das i -te Bit mit Wk. $> \frac{1}{2}$ vorhersagen.

- Gibt es einen (i, ϵ) -Predictor für eine Zufallsvariable X auf \mathbb{B}^L , dann gibt es einen ϵ -Distinguisher von X und $U_{\mathbb{B}^L}$:
 $D: \mathbb{B}^L \rightarrow \{0,1\}; D(b_1, \dots, b_{i-1}, b_i, \dots, b_L) := \begin{cases} 1 & \text{wenn } P(b_1, \dots, b_{i-1}) = b_i \\ 0 & \text{sonst} \end{cases}$

- Gibt es einen ϵ -Distinguisher für X und $U_{\mathbb{B}^L}$, dann gibt es einen $(i, \epsilon/L)$ -Predictor:
 $P(x_1, \dots, x_{i-1}) = \begin{cases} u_i & \text{wenn } P(x_1, \dots, x_{i-1}, u_i, \dots, u_L) = 1 \\ 1+u_i & \text{sonst} \end{cases} \rightarrow \text{wobei die } u_i \text{ gleichförmig zufällig aus } \mathbb{B} \text{ gewählt!}$
 Per Teleskopsummen-Argument muss es ein i mit Wk. $\geq \epsilon/L$ geben!
 BBS's Sicherheit lässt sich auf das Faktorisierungsproblem reduzieren (Äquivalenz) (b.w.)

- Blum-Blum-Shub-Generator (BBS-Generator)**: **Faktorisierungsproblem reduzieren (Äquivalenz) (b.w.)**
 Seien p, q zwei große Primzahlen mit $p, q \equiv 3 \pmod{4}$ und geheim.
 de.wiki: Es sollte $2 < p, q < 1000$ gelten und $p \pm 1$ und $q \pm 1$ sollten jew. einen Primfaktor $> \sqrt{n}$ haben.
 [1] Sei $n = p \cdot q$ (öffentlich) und $s_0 \in \mathbb{Z}_n^*$ der Seed/Startwert (d.h. $\text{ggT}(s_0, n) = 1$).
 Es sollen L Zufallsbits generiert werden.
 [2] Generiere $s_{i+1} = s_i^2 \pmod{n}$ für $i = 1, \dots, L$.
 [3] Behalte nur die jeweiligen least significant bits: $b_i = s_i \pmod{2}$.
 [4] Output: $(b_1, b_2, \dots, b_L) = \text{BBS}_L(s_0)$ (Achtung: keine Rückgabe von b_0)
 de.wiki: Bsp.: $p=7, q=11, n=77, s_0=64: 64 \rightarrow 15, 71, 36, 64, \dots$ (ab hier wiederholt es sich)
 $\text{BBS}_4(64) = (1, 1, 0, 0)$

- Beweis: BBS ist sicher: $\xrightarrow{\text{reduziere auf}}$ Faktorisierungsproblem \leq $\xrightarrow{\text{reduziere auf}}$ Quadratisches Restproblem \leq BBS-"Problem"

Leich lösbar, wenn man $\sqrt{\cdot}$ ziehen kann

Dabei sei das Quadratische Restproblem wie folgt definiert:
Geg. n und $x \in \mathbb{Z}_n^*$, gibt es ein $y \in \mathbb{Z}_n^*$ mit $y^2 = x$?
D.h.: Ist x ein quadratischer Rest modulo n , oder nicht?

Red. QRP auf BBS: [1] Wenn man einen ϵ -Distinguisher D konstruieren könnte, der $BBS(\mathbb{Z}_n^*)$ und U_{BL} unterscheidet, so könnte man das Quadratische Restproblem in \mathbb{Z}_n^* lösen:

Frage: Ist $x \in \mathbb{Z}_n^*$ ein Quadrat(-ischer Rest), d.h. ist $x \in (\mathbb{Z}_n^*)^2$?

Angenommen, D sei ein ϵ -Distinguisher von $BBS(\mathbb{Z}_n^*)$ und U_{BL} .
D.h. $D(b_1, \dots, b_L) = 1$ mit Wk. $\frac{1}{2} + \epsilon$, wenn $\exists x: BBS(x) = (b_1, b_2, \dots, b_L)$. \approx iterte. 2
Bau einen Distinguisher D mit $D(w_1, \dots, w_L) = D(w_1, \dots, w_1)$. \approx iterten 2

Wir wissen: Dann gibt es auch einen (ϵ/L) -Predictor P für D , für den gilt: $P(w_1, \dots, w_{i-1}) = w_i$ mit Wk. $\frac{1}{2} + \epsilon/L$.

P kann jetzt quasi Wurzeln ziehen, d.h.:
Wenn $P(\text{reverse}(BBS(x))) = \text{least-sign-bit}(\sqrt{x})$ "berechnet" uns P
(in VL: "...") $= \text{least-sign-bit}(x \text{ oder } -x)$ (3-3=0=1)
 $= x \bmod 2 \neq -x \bmod 2$ (da n ungerade),
und nicht...

dann ist x ein Quadrat (mit Wk. $\frac{1}{2} + \epsilon/L$ versteht sich).

Warum ist das so?

Das Quadrieren auf $(\mathbb{Z}_n^*)^2$ ist ein Gruppenisomorphismus, d.h. wenn $x \in (\mathbb{Z}_n^*)^2$, existiert ein $y \in (\mathbb{Z}_n^*)^2$ mit $x = y^2$.

Wir schreiben: $y = \sqrt{x}$, Hauptwurzel.

P berechnet uns genau diese Hauptwurzel, da es ja auf Mustern der Form $(\dots, x^{32}, x^{16}, x^8, \dots)$, sprich auf Hauptwurzeln "trainiert" wurde (meine Worte).

[2] Wenn man mod n Wurzeln ziehen könnte, so könnte man n faktorisieren:

Bsp.: aus der VL: $2 \xrightarrow{\text{quadr.}} 4 \xrightarrow{\text{Wurzel}} 5$ \leftarrow Sage gibt wieder 2 zurück
 $2^2 \equiv 4 \equiv 5^2 \pmod{21} \Rightarrow 5^2 - 2^2 \equiv 0 \pmod{21}$
 $\Rightarrow 21 \mid (5^2 - 2^2)$
 $\Rightarrow 21 \mid (5-2) \cdot (5+2) \leftarrow$ "habe 21 faktorisiert"
 $\Rightarrow \text{ggT}(21, 5-2) = 3 = p$ (in Anführungszeichen)
 $\wedge \text{ggT}(21, 5+2) = 7 = q$
(vorausgesetzt: $21 \nmid (5-2) \wedge 21 \nmid (5+2)$,
vgl. CRYPTO1: Faktorisierung)