Starke und schwache Einwegfunktionen

Daniela Weinberg
weinberg@informatik.hu-berlin.de

Seminar: Perlen der theoretischen Informatik
Dozenten: Prof. Johannes Köbler, Olaf Beyersdorff
Wintersemester 2002/2003

12. Dezember 2002

1 Schwache Einwegfunktionen implizieren starke Einwegfunktionen

Wir wollen zeigen, daß die Existenz von *schwachen* Einwegfunktionen die Existenz von *starke* Einwegfunktionen impliziert. Aber zunächst überlegen wir uns, daß nicht jede schwache Einwegfunktion unbedingt eine starke Einwegfunktion ist. Dazu betrachten wir folgendes Beispiel:

Sei $\mid z \mid = \mid x \mid$ und sei f eine Einwegfunktion, die o.B.d.A. längenerhaltend ist. D.h. $\mid f(x) \mid = \mid x \mid$. Nun konstruieren wir eine Funktion g wie folgt:

$$g(z,x) = \begin{cases} (z, f(x)) & z \textit{ beginnt mit } log_2 \mid x \mid \textit{ Nullen.} \\ (z,x) & \textit{sonst} \end{cases}$$

Die Funktion g kann keine starke Einwegfunktion sein, da für alle außer einem $\frac{1}{n}$ Teil der Zeichenketten der Länge $2n$ die Funktion mit der Identitätsfunktion gleich ist. Nun beweisen wir, daß g eine schwache Einwegfunktion ist.

Behauptung 1.1. *Sei f eine (schwache) Einwegfunktion. Dann ist die Funktion g, wie oben konstruiert, eine schwache Einwegfunktion.*

Beweis. Betrachten wir die Funktion g genauer. Invertieren wir g nun auf solchen Eingaben, so daß g nicht mit der Identitätsfunktion zusammen trifft, müssen wir uns nur Gedanken über die Invertierung von f machen. Mit einer Wahrscheinlichkeit von mehr als $1 - \frac{1}{n}$ wird g mit Eingaben der Länge größer als $2n$ invertiert. Somit muß g also mit einer hohen Wahrscheinlichkeit auf Eingaben invertiert werden, wo sie die Funktion f anwendet. Wenn g also keine schwache Einwegfunktion ist, so ist f es auch nicht.
Werden wir jetzt etwas konkreter: wir nehmen uns einen probabilistischen in Polynomialzeit arbeitenden Algorithmus B' her, der die Funktion g invertiert. Nun konstruieren wir noch einen weiteren probabilistischen in Polynomialzeit arbeitenden Algorithmus A', der f mit einer ähnlichen Erfolgswahrscheinlichkeit invertiert. Bei der Eingabe von y macht der Algorithmus A' nun folgendes:

- setze $n \stackrel{def}{=} \mid y \mid$
- setze $l \stackrel{def}{=} log_2 n$
- wähle gleichverteilt ein z' aus $\{0,1\}^{n-l}$
- berechne $z \stackrel{def}{=} B'(0^l z', y)$
- stoppe mit Ausgabe des $n - bit$ Suffixes von z

Definieren wir nun noch eine Menge S_{2n}, die alle $2n - bit$ langen Zeichenketten enthalten möge, die gerade mit $log_2 n$ Nullen anfangen:

$$S_{2n} \stackrel{def}{=} \left\{ 0^{log_2 n} \alpha : \alpha \; in \{0,1\}^{n-log_2 n} \right\}$$

Somit können wir uns folgendes überlegen:

$$\begin{aligned}
&\Pr[A'(f(U_n)) \in f^{-1}(f(U_n))] \\
&\qquad \geq \Pr[B'(0^l U_{n-l}, f(U_n)) \in (0^l U_{n-l}, f^{-1}(f(U_n))] \\
&\qquad = \Pr[B'(g(U_{2n}) \in g^{-1}(g(U_{2n})) \mid U_{2n} \in S_{2n}] \\
&\qquad \geq \frac{\Pr[B'(g(U_{2n}) \in g^{-1}(g(U_{2n}))] - \Pr[U_{2n} \notin S_{2n}]}{\Pr[U_{2n} \in S_{2n}]} \\
&\qquad = n \cdot \left(\Pr[B'(g(U_{2n}) \in g^{-1}(g(U_{2n}))] - \left(1 - \frac{1}{n}\right) \right) \\
&\qquad = 1 - n \cdot (1 - \Pr[B'(g(U_{2n}) \in g^{-1}(g(U_{2n}))])
\end{aligned}$$

Hieraus sehen wir, daß für uns nur der Fall interessant ist wo
$\Pr[B'(g(U_{2n}) \in g^{-1}(g(U_{2n}))] > 1 - \frac{1}{n}$ gilt.
Eine kurze Anmerkung noch zu den Umformungen: $\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]}$ und $\Pr[A \cap B] \geq \Pr[A] - \Pr[\neg B]$

Es folgt nun folgendes für jedes Polynom $p(\cdot)$ und jede ganze Zahl n:
wenn B' die Funktion g auf $g(U_{2n})$ mit einer Wahrscheinlichkeit größer als $1 - \frac{1}{p(2n)}$ invertiert, so invertiert auch A' die Funktion f auf $f(U_n)$ mit einer Wahrscheinlichkeit größer als $1 - \frac{n}{p(2n)}$. Nehmen wir jetzt an, daß g keine schwache Einwegfunktion ist. Das hieße also, daß für jedes Polynom $p(\cdot)$ unendlich viele $m's$ existieren müssen, so daß g auf $g(U_m)$ mit einer Wahrscheinlichkeit von mehr als $1 - \frac{1}{p(m)}$ erfolgreich invertiert werden kann. Dann wäre auch f keine schwache Einwegfunktion, da es dann für jedes Polynom $q(\cdot)$ unendlich viele $n's$ geben müßte, so daß f auf $f(U_n)$ mit einer Wahrscheinlichkeit von mehr als $1 - \frac{1}{q(n)}$ erfolgreich invertiert werden kann, wobei $q(n) = p(2n)/n$. Das steht aber im Widerspruch zu unserer Annahme, daß f eine schwache Einwegfunktion ist! □

Was haben wir nun gezeigt? So es denn Einwegfunktionen gibt, gibt es schwache, die keine starken Einwegfunktionen sind. Somit können wir also ausschließen, daß alle Einwegfunktionen starke Einwegfunktionen sind. Im weiteren Verlauf wollen wir nun zeigen, daß folgendes gilt.

2 Schwache Einwegfunktionen existieren genau dann, wenn starke Einwegfunktionen existieren

Theorem 2.1. *Schwache Einwegfunktionen existieren genau dann, wenn starke Einwegfunktionen existieren.*

Beweis. Sei f nun eine schwache Einwegfunktion und sei p gerade das Polynom, welches wir bei der Definition von schwachen Einwegfunktionen eingeführt haben. Jeder probabilistische in Polynomialzeit arbeitende Algorithmus invertiert f auf $f(U_n)$ mit einer Wahrscheinlichkeit von mindestens $\frac{1}{p(n)}$ falsch. Weiterhin nehmen wir an, daß f längenerhaltend ist. Wir definieren nun die Funktion g wie folgt:

$$g(x_1, \ldots, x_{t(n)}) \stackrel{def}{=} f(x_1), \ldots, f(x_{t(n)})$$

wobei gilt: $\mid x_1 \mid = \cdots = \mid x_{t(n)} \mid = n$ und $t(n) \stackrel{def}{=} n \cdot p(n)$.

Wir zerlegen also die $n^2 p(n)$ bit lange Eingabe von g in $t(n)$ Blöcke und wenden f auf jeden Block an. Demnach müssen wir nun um die Funktion g auf $g(x_1, \ldots, x_{t(n)})$ zu invertieren praktisch das *pre-image* zu jedem $f(x_i)$ finden.

Wir können nun folgern, daß g eine starke Einwegfunktion ist und wir einen Invertierungsalgorithmus haben, der separat für jedes $f(x_i)$ angesetzt arbeitet. Wäre dies der Fall, ist die Wahrscheinlichkeit dafür, daß der Algorithmus erfolgreich alle $f(x_i)$ invertiert, höchstens

$$\Big(1 - \frac{1}{p(n)}\Big)^{n \cdot p(n)} < 2^{-n}.$$

Die Annahme, daß der Algorithmus, welcher unsere Funktion g invertiert, unabhängig für jedes $f(x_i)$ arbeitet, kann nicht gerechtfertig werden. Daher müssen wir uns dem Problem anders nähern.

Der Beweis wird wie folgt aussehen:

Als erstes nehmen wir an, daß unsere Funktion g keine starke Einwegfunktion ist. Es existiert also ein in Polynomialzeit arbeitender Algorithmus, der g mit einer nicht unbedeutenden Wahrscheinlichkeit invertiert. Wir werden einen Algorithmus angeben, der für unendlich viele n die Funktion f auf $f(U_n)$ mit einer Wahrscheinlichkeit größer als $1 - \frac{1}{p(n)}$ invertiert (ganz im Widerspruch zu unserer Hypothese). Der Invertierungsalgorithmus von f nutzt gerade den Invertierungsalgorithmus von g als eine Unterfunktion.

Wir nehmen ja an, daß g keine starke Einwegfunktion ist. Daher gilt folgendes: Es gibt einen Algorithmus B' und ein Polynom $q(\cdot)$, so daß gilt:

$$\Pr[B'(g(U_m)) \in g^{-1}(g(U_m))] > \frac{1}{q(m)}, \tag{1}$$

wobei B' der probabilistische in Polynomialzeit arbeitende Algorithmus zur Invertierung von g ist. Noch ein paar Festlegungen: sei M' eine unendliche Menge von natürlichen Zahlen, sei N' eine unendliche Menge von $n's$ für die gilt $n^2 \cdot p(n) \in M'$. Somit sind also alle $m's$ der Form $n^2 \cdot p(n)$.
Unter der Benutzung von B' bilden wir nun also einen in polynomieller Zeit arbeitenden Algorithmus A', der $a(n)$ mal wiederholt wird, zum Invertieren von f. Dabei setzen wir $a(n) \stackrel{def}{=} 2n^2 \cdot p(n) \cdot q(n^2 p(n))$. A' arbeitet mit folgender Prozedur:

---

wiederhole $a(n)$ mal:

Prozedur I

1: Input: y (setze $n \stackrel{def}{=} \mid y \mid$).
2: **for** i = 1 to t(n) **do**
3: wähle unabhängig und gleichverteilt eine Sequenz von Strings der Form $x_1, \ldots, x_{t(n)} \in \{0,1\}^n$
4: berechne $(z_1, \ldots, z_{t(n)}) \leftarrow B'(f(x_1), \ldots, f(x_{i-1}), y, f(x_{i+1}), \ldots, f(x_{t(n)}))$
5: wenn $f(z_i) = y$, dann stoppe und gebe z_i aus
6: **end for**

Zusammen mit der Gleichung (1) können wir uns nun eine untere Grenze für die Erfolgswahrscheinlichkeit unseres Algorithmuses A' überlegen. Vorerst müssen wir allerdings noch eine Menge definieren. Nennen wir sie S_n. Sie möge alle $n - bit$ Zeichenketten enthalten, welche die Prozedur I mit einer Wahrscheinlichkeit von mehr als $\frac{n}{a(n)}$ zum Erfolg führt.

$$S_n \stackrel{def}{=} \left\{ x \in \{0,1\}^n : Pr[I(f(x)) \in f^{-1}(f(x))] > \frac{n}{a(n)} \right\} \tag{2}$$

Nun wollen wir zeigen, daß unsere Menge S_n alle, aber höchstens $\frac{1}{2p(n)}$ viele, Teilmengen von Zeichenketten der Länge $n \in N'$ enthält. Für jede Zeichenkette $x \in S_n$ invertiert unser Algorithmus A' die Funktion f auf $f(x)$ mit einer Wahrscheinlichkeit, die exponentiell nahe 1 ist.

Behauptung 2.1. *Es gilt für jedes $x \in S_n$*

$$Pr[A'(f(x)) \in f^{-1}(f(x))] > 1 - \frac{1}{2^n}$$

Beweis. Aus der Definition der Menge S_n folgt, daß die Prozedur I die Funktion $f(x)$ mit einer Wahrscheinlichkeit von mindestens $\frac{n}{a(n)}$ invertiert. Da der Algorithmus A' die Prozedur I $a(n)$ mal wiederholt, folgt nun

$$Pr[A'(f(x)) \notin f^{-1}(f(x))] < \left(1 - \frac{n}{a(n)}\right)^{a(n)} < \frac{1}{2^n}$$

□

Behauptung 2.2. *Für jedes $n \in N'$ gilt:*

$$\mid S_n \mid > \left(1 - \frac{1}{2p(n)}\right) \cdot 2^n$$

Beweis. Nehmen wir an, es gelte im Gegensatz zur der oben genannten Gleichung folgendes:

$$\mid S_n \mid \leq \left(1 - \frac{1}{2p(n)}\right) \cdot 2^n$$

Wir versuchen nun einen Widerspruch zu unserer Gleichung (2) zu finden. Es sei:

$$s(n) \stackrel{def}{=} \Pr[B'(g(U_{n^2p(n)})) \in g^{-1}(g(U_{n^2p(n)}))] > \frac{1}{q(n^2p(n))} \tag{3}$$

Seien $U_n^{(1)}, \ldots, U_n^{(n \cdot p(n))}$ $n-bit$ lange Blöcke der Zufallsvariablen $U_{n^2p(n)}$, d.h. diese $U_n^{(i)}$ sind unabhängige Zufallsvariablen gleichverteilt auf $\{0,1\}^n$. Nun zerlegen wir das Ereignis beschrieben durch die Gleichnung (2.2) in zwei disjunkte Ereignisse. Diese entstehen dadurch, indem wir zum einen die $U_n^{(i)}$'s betrachten die in S_n liegen und zum anderen eben jene die nicht in der Menge liegen. Intuitiv können wir sagen, daß B' in einem solchen Fall nicht gut funktionieren kann, da eben dieser Fall mit der Erfolgswahrscheinlichkeit von I korrespondiert. Auf der anderen Seite ist die Wahrscheinlichkeit, daß alle $U_n^{(i)}$'s in S_n liegen recht gering. Konkret definieren wir folgendes:

$$s_1(n) \stackrel{def}{=} \Pr[B'(g(U_{n^2p(n)})) \in g^{-1}(g(U_{n^2p(n)})) \wedge (\exists i : U_n^{(i)} \notin S_n)]$$

und

$$s_2(n) \stackrel{def}{=} \Pr[B'(g(U_{n^2p(n)})) \in g^{-1}(g(U_{n^2p(n)})) \wedge (\forall i : U_n^{(i)} \in S_n)]$$

Offensichtlich gilt: $s(n) = s_1(n) + s_2(n)$ (die Ereignisse in den s_i's sind disjunkt). Wir versuchen nun einen Widerspruch zur unteren Schranke von $s(n)$ aufzustellen, indem wir obere Schranken für $s_1(n)$ und $s_2(n)$ angeben, die sich dann zu weniger als $s(n)$ addieren.
Als erstes stellen wir eine obere Schranke für $s_1(n)$ auf. Richten wir unser Augenmerk auf die Prozedur I. I invertiert ja f auf Eingaben $f(x)$ mit einer Wahrscheinlichkeit, die von dem Erfolg von B' zur Invertierung von g auf einer Sequenz der zufälligen f-Bilder, die $f(x)$ beinhalten, abhängt. Für jedes $x \in \{0,1\}^n$ und jedes $1 \leq i \leq n \cdot p(n)$, ist die Wahrscheinlichkeit, daß I die Funktion f auf $f(x)$ invertiert größer oder gleich der Wahrscheinlichkeit, daß B' g auf $g(U_{n^2p(n)})$ invertiert. Dabei gilt $U_n^{(i)} = x$. (Der Erfolg von B' g zu invertieren, heißt, daß f auf dem $i-ten$ Block invertiert wurde und trägt zur Erfolgswahrscheinlichkeit von I bei.) Es folgt nun, daß für jedes $x \in \{0,1\}^n$ und jedes $1 \leq i \leq n \cdot p(n)$:

$$\Pr[I(f(x)) \in f^{-1}(f(x))] \geq \Pr[B'(g(U_{n^2p(n)})) \in g^{-1}g(U_{n^2p(n)})) \mid U_n^{(i)} = x], \tag{4}$$

gilt. Da für $x \notin S_n$ die linke Seite nicht groß werden kann, wollen wir zeigen, daß $s_1(n)$ nicht groß sein kann. Somit können wir uns nun folgendes überlegen:

$$
\begin{aligned}
s_1(n) &= \Pr[\exists i : B'(g(U_{n^2p(n)})) \in g^{-1}(g(U_{n^2p(n)})) \wedge U_n^{(i)} \notin S_n] \\
&\leq \sum_{i=1}^{n\cdot p(n)} \Pr[B'(g(U_{n^2p(n)})) \in g^{-1}(g(U_{n^2p(n)})) \wedge U_n^{(i)} \notin S_n] \\
&\leq \sum_{i=1}^{n\cdot p(n)} \sum_{x \notin S_n} \Pr[B'(g(U_{n^2p(n)})) \in g^{-1}(g(U_{n^2p(n)})) \wedge U_n^{(i)} = x] \\
&= \sum_{i=1}^{n\cdot p(n)} \sum_{x \notin S_n} \Pr[U_n^{(i)} = x] \cdot \Pr[B'(g(U_{n^2p(n)})) \in g^{-1}(g(U_{n^2p(n)})) \mid U_n^{(i)} = x] \\
&\leq \sum_{i=1}^{n\cdot p(n)} max_{x \notin S_n}\{\Pr[B'(g(U_{n^2p(n)})) \in g^{-1}(g(U_{n^2p(n)})) \mid U_n^{(i)} = x]\} \\
&\leq \sum_{i=1}^{n\cdot p(n)} max_{x \notin S_n}\{\Pr[I(f(x)) \in f^{-1}(f(x))]\} \\
&\leq n \cdot p(n) \cdot \frac{n}{a(n)} \\
&= \frac{n^2 \cdot p(n)}{a(n)}
\end{aligned}
$$

Die letzte Ungleichung nutzt die Definition von S_n und die davor nutzt die Gleichung (2.2).

Nun können wir eine obere Grenze für $s_2(n)$ angeben. Erinnern wir uns, daß unsere Hypothese ja $\mid S_n \mid \leq (1 - \frac{1}{2p(n)}) \cdot 2^n$ war. Es folgt also:

$$
\begin{aligned}
s_2(n) &\leq \Pr[\forall i : U_n^{(i)} \in S_n] \\
&\leq \left(1 - \frac{1}{2p(n)}\right)^{n\cdot p(n)} \\
&< \frac{1}{2^{n/2}} \\
&< \frac{n^2 \cdot p(n)}{a(n)}
\end{aligned}
$$

(Die letzte Ungleichung gilt für hinreichend große n.)

Führen wir nun unsere oberen Grenzen der s_i's zusammen, erhalten wir $s_1(n) + s_2(n) < \frac{2n^2 \cdot p(n)}{a(n)} = \frac{1}{q(n^2p(n))}$, wo Gleichheit eben bei der Definition von $a(n)$ gilt. Auf der anderen Seite gilt aber $s_1(n)+s_2(n) = s(n) > \frac{1}{q(n^2p(n))}$, wo Ungleichheit wegen der Gleichung 2.2 gilt. Wir haben also einen Widerspruch und unsere Behauptung folgt!

□

Kombinieren wir nun unsere letzten beiden Behauptungen, so erhalten wir folgendes:

$$
\begin{aligned}
Pr[A'(f(U_n)) \in f^{-1}(f(U_n))] & \\
& \geq Pr[A'(f(U_n)) \in f^{-1}(f(U_n)) \wedge U_n \in S_n] \\
& = Pr[U_n \in S_n] \cdot Pr[A'(f(U_n)) \in f^{-1}(f(U_n)) | U_n \in S_n] \\
& \geq \left(1 - \frac{1}{2p(n)}\right) \cdot (1 - 2^{-n}) \\
& > 1 - \frac{1}{p(n)}
\end{aligned}
$$

Es muß also ein probabilistischer in polynomieller Zeit arbeitender Algorithmus A' existieren, der f auf $f(x)$, für $n \in N'$, mit einer Wahrscheinlichkeit von mehr als $1 - \frac{1}{p(n)}$ invertiert. Diese Schlußfolgerung steht aber nun ganz im Widerspruch zu der Hypothese, daß jeder probabilistische in polynomieller Zeit arbeitende Algorithmus mit einer Wahrscheinlichkeit von mindestens $\frac{1}{p(n)}$ bei der Invertierung der Funktion f scheitert. Somit folgt also unsere Behauptung. □

3 Zusammenfassung

Was haben wir denn eigentlich im letzten Absatz gemacht? Wir haben bewiesen, daß es schwache Einwegfunktionen genau dann gibt, wenn es auch starke Einwegfunktionen gibt. Wir konstruierten uns zu einer schwachen Einwegfunktion f eine in Polynomialzeit berechenbare Funktion g und wollten später beweisen, daß diese Funktion eine starke Einwegfunktion ist. Für diesen Beweis nutzten wir ein Reduzierbarkeitsargument. Wir reduzierten also das Lösen des einen Problems auf das Lösen des anderen. Da wir aber eine Prozedur nutzen, die das Problem nur für eine bestimmte Wahrscheinlichkeit auf einer bestimmten Verteilung löst, sprechen wir von einem Reduzierbarkeitsargument und nicht von einer Reduzierung im herkömmlichen Sinne.

Literatur

[1] Goldreich, Oded. Foundations of Cryptography, Cambridge University Press, 2001.