Zero-Knowledge Protokolle für NP

Jens Harzer

12. August 2002

Zusammenfassung

Es soll ein Verfahren zur Konstruktion von Zero-Knowledge Protokollen für Sprachen aus NP gezeigt werden. Hierbei wird eine Zero-Knowledge Protokoll für das Problem der *3-Färbbarkeit* gezeigt. Da dieses Problem NP-Vollständig ist, läßt sich durch dieses Protokoll unter Verwendung der Reduktion in Polynomialzeit für jede andere Sprache aus NP ein Zero-Knowledge Protokoll ableiten. Die Motivation zur Kontruktion von Zero-Knowledge Protokollen für Sprachen aus NP liegt vor allem darin, diese Protokolle zu Authentifizierungszwecken nutzen zu können.

Das Zero-Knowledge Protokoll für 3-Färbbarkeit (G3C)

Das 3-Farben Problem *geg.:* Ein beliebiger Graph $G = (V, E)$
ges.: Existiert eine Abbildung $\psi : V \rightarrow \{1, 2, 3\}$ sodaß für jedes Paar $(u, v) \in E$ gilt:

$$\psi(u) \neq \psi(v)$$

Beschreibung des Zero-Knowledge Protokolles für G3C (Überblick) Sei ψ eine gültige 3-Färbung eines Eingabegraphen $G = (V, E)$. Dann läßt sich für G3C das folgende Zero-Knowledge-Protokoll angeben:

1. Im ersten Schritt wählt der Prover eine zuflige Permutation zwischen den Farben von ψ. Sei π diese Permutation. Es gilt

   $$\pi \in_R Sym[3]$$

   Der Prover berechnet durch die Permutation durch eine neue Abbildung $\phi = \pi(\psi(v))$ für jedes $v \in V$. Es existieren nun für jeden Knoten $v \in V$ ein "versiegelter Umschlag"die zunächst informell beschrieben werden, später dann durch *Commitment-Schemes* ersetzt werden. Diese versiegelten Umschläge können nur durch einen Schlüssel geöffnet werden, die im Besitz des Provers sind. In jeden dieser versiegelten Umschläge wird nun eine der neuberechneten Farben geschrieben.

2. Der Verifier wählt zufällig eine Kante $(u, v) \in E$ und sendet u und v an den Prover.

3. Der Prover sendet an den Verifier für die Knoten $u, v \in V$ die zugehörigen Schlüssel für die versiegelten Umschläge an den Verifier.
4. Der Verifier kann nun mit den Schlüsseln die Umschläge öffnen und die darin enthaltenen Farben vergleichen. Sind die Farben unterschiedlich, akzeptiert der Verifier, sind sie gleich, verwirft der Verifier.

Damit dieses Protokoll den Anforderungen für ein interaktives Beweissystem genügt, muß es natürlich mehrmals wiederholt werden. Der Prover muß dabei jedesmal eine andere Permutation wählen, damit der Verifier keine Informationen aus vorherigen Durchläufen nutzen kann.

Completeness Halten sich sowohl Prover und Verifier an das Protokoll, und ist der Eingabegraph $G = (V, E)$ tatsächlich 3-färbbar, akzeptiert der Verifier immer. Unter der Voraussetzung, daß G 3-färbbar ist, ist auch ϕ eine gültige Färbung und für jede Kante $e = (u, v)$ ist die Bedingung $\phi(u) \neq \phi(v)$ erfüllt. Somit gilt für $G \in G3C$:

$$Prob[(P, V)((G)) = accept] = 1$$

Soundness Sein nun $G \notin G3C$. Der Prover möchte nun den Verifier davon überzeugen, daß G doch 3-färbbar ist. Da nun $G \notin G3C$ gilt, muß es eine Kante $e = (u, v)$ geben mit $\phi(u) = \phi(v)$. Sei nun A das Ereignis, bei welchem für eine zufällig ausgewählte Kante e_0 die Eigenschaft $\phi(u) \neq \phi(v)$ gilt. Somit gilt für die Wahrscheinlichkeit von A:

$$Prob[A] \leq \frac{|Kanten - \{e_0\}|}{|E|} = 1 - \frac{1}{|E|}$$

Somit gilt bei $G \notin G3C$ für alle Prover P^*

$$Prob[\langle P^*, V\rangle(G)) = accept] = 1 - \frac{1}{|E|}$$

Bit-Commitment-Schemes

Informelle Beschreibung Im weiteren Verlauf des Beweises werden auf Seite des Provers *Bit-Commitment-Schemes* verwendet. Es handelt sich dabei um einen Mechanismus, mit welchem der Prover Information gegenüber dem Verifier geheimhält und kontrolliert wieder freigibt, indem er dem Verifier einen Schlüssel sendet, durch welchen ihm diese Information zugänglich gemacht wird. Im weiteren Verlauf wird die genaue Implementaion dieser *Bit-Commitment-Schemes* beschrieben.

Anforderungen an ein Bit-Commitment-Scheme Ein Bit-Commitment-Scheme ist ein Zwei-Phasen-Protokoll, welches wiederum aus zwei interaktiven Turing-Maschinen besteht. Hierbei kommunizieren ein Absender S und ein Empfänger R miteinander. Der Empfänger kann dabei einen Wert schützen. Ein Bit-Commitment-Scheme muß zwei Anforderungen erfüllen:

1. *Secrecy* Am Ende der ersten Phase hat der Empfänger keine Informationen über den geschützten Wert des Absenders. Insbesondere sind für den Verifier verschiedene verschlüsselte Werte (Commitments) berechnungsmäßig nicht zu unterscheiden.
2. *Unambiguity* In der zweiten Phase gibt es höchstens einen Wert, den der Empfänger als gültigen Schlüssel für den geschützten Wert akzeptiert. Es wird hierbei keine echte Eindeutigkeit verlangt. Sei $(r, \overline{m})$ die Sicht des Verifiers auf die Antwort des Provers. Diese Antwort ist dann mehrdeutig, wenn sie fr verschiedene verschlsselte Werte steht. Die Bedingung der Unambiguity ist genau dann er"ullt, wenn fü alle Antworten des Provers bis einen vernachl"assigbaren Bruchteil, wobei die Länge von r polynomiell beschränkt ist, kein $\overline{m}$ existiert, sodass $(r, \overline{m})$ mehrdeutig ist.

Sei $S(s, \sigma)$ die Nachrichten, die der Absender dem Empfänger während der ersten Phase schickt, wobei δ der geschützte Wert und s die Zufallsbits des Absenders sind. Durch die Bedingung *Secrecy* ist gefordert, daß die Zufallsvariablen $S(s, 0)$ und $S(s, 1)$ durch einen polynomiellen Schaltkreis nicht zu unterscheiden sind.

Sei die Sicht von S (R), bezeichnet durch $view(S)$, bzw. $view(R)$ die gesammelten Informationen, die dem Absender, bzw. dem Empfänger während der gesamten Kommunikation bekannt sind. Sei nun r die Zufallsbits von r und m die Sequenz von Nachrichten, die R von S erhalten hat. Dann ist

$$view(R) = (r, m)$$

Sei δ das geschütze Bit von S und bezeichne die Zufallsbits von S mit s. Dann ist

$$view(S) = (s, \sigma)$$

Die Bedingung der *Unambiguity* fordert, daß für jedes r bis auf einen vernachlässigbaren Bruchteil, folgende Bedingung erfüllt ist:

$$view(S) = (s, 0) \; und \; view(R) = (r, m)$$

und

$$view(S) = (s', 0) \; und \; view(R) = (r, m)$$

Inituitiv ist damit gemeint, daß der Schlüssel jeweils eindeutig sein muß.

Verwendung von Bit-Commitment-Schemes Im folgenden wrid ein interaktives Beweissystem für G3C unter Verwendung von Bit-Commitment-Schemes beschrieben. Gegeben sei ein Graph $G = (V, E)$.

1. Der Prover berechnet eine 3-Färbung von G. Diese Färbung sei bezeichnet durch ψ. Der Prover wählt eine zufällige Permutation π über $\{1, 2, 3\}$ und berechnet $\phi := \pi(\psi(v))$ für jedes $v \in V$. Der Prover benutzt nun ein Bit-Commitment-Scheme um seine Permutation zu verschlüsseln und zu signieren. Der Prover wählt dazu unabhängig zu jedem Knoten einen String $s_1, ..., s_n \in \{0, 1\}^n$, wobei n für die Verschlußselungsstärke steht, und berechnet für jeden Knoten mit einer one-Way Funktion C ein $c_i = C_{s_i}(\phi(i))$ für jedes $i \in V$ und schickt $c_1, ..., c_n$ an den Verifier.

2. Der Verifier bestimmt eine Kante $(u, v) \in E$ und schickt diese an den Prover
3. Der Prover schickt für die beiden Knoten (u, v) die Strings, welche den Schlüssel für das Bit-Commitment-Scheme darstellen sowie die Färbung der Knoten an den Verifier. Der Prover sendet somit die Paare $(s_u, \phi(u))$ und $(s_v, \phi(v))$.
4. Seien (s, σ) und (s, τ) die Paare, die der Verifier vom Prover erhalten hat. Der Verifier kann nun durch die Funktion C verifizieren, ob der Prover ehrlich war und die richtigen Farben gschickt hat, indem er überprüft, ob $c_u = C_s(\sigma)$ und ob $c_v = C_s(\tau)$ ist. Dann prüft der Verifier ob $\sigma \neq \tau$ ist. Sind alle Bedingungen erfüllt, akzeptiert der Verifier, ansonsten verwirft er.

Dieses Protokoll kann durch einen probabilistischen Simulator M^* simuliert werden, der wie folgt arbeitet:

1. Sei q ein Polynom, welches die Laufzeit des Verifiers V^* begrenzt. Der Simulator wählt einen String $r \in \{0,1\}^{q(n)}$, wobei $n = |V|$, welcher die Inschrift des Zufallsbandes von V^* darstellt.
2. M^* wählt nun zufällig n Werte $e_1, ..., e_n \in \{1,2,3\}$ und n zufällige Strings $s_1, ..., s_n \in \{0,1\}^{q(n)}$ die benutzt werden, um diese Werte zu verschlüsseln und zu signieren. Der Simulator berechnet für jedes $i \in V$ ein $d_i = C_{s_i}(e_i)$.
3. Der erste Schritt des Verifiers wird simuliert: G wird auf das Eingabeband von V^* geschrieben, r aus Schritt 1 wird auf das Zufallsband beschrieben und die Sequenz $(d_1, ..., d_n)$ wird auf das Nachrichtenband geschrieben. Nach einer polynomiell begrenzten Anzhahl von Schritten kann der Simulator die Ausgabe von V^* auf dem Nachrichtenband lesen. Diese Ausgabe sei durch m bezeichnet.
4. Der zweite Schritt des Provers wird simuliert: Im Fall $e_u \neq e_v$ war die Simulation erfolgreich und der Simulator hält mit der Bandinschrift $(G, r, (d_1, ..., d_n), (s_u, e_u, s_v, e_v))$. Andernfalls ist die Simulation gescheitert und der Simulator stoppt mit Ausgabe $\perp$.

Um den Beweis für $NP \subseteq CZK$ zu vervollständigen, müssen Completeness, Soundness un Zero-Knowledge Eigenschaften bewiesen werden:

Completeness Unter der Voraussetzung, daß ein ehrlicher Verifier vorliegt und sich sowohl Prover und Verifier an das Protokoll halten, wird V im Fall $G \in G3C$ immer akzeptieren, da der Prover jedesmal eine gültige Färbung ϕ berechnen kann. Es gilt somit

$$Pr[\langle P, V^* \rangle(G) = accept] = 1$$

Soundness Es gelte nun $G \notin G3C$, somit existert mindestens eine Kante $(u, v) \in E$ mit $\phi(u) = \phi(v)$. Somit sind folgende zwei Fälle denkbar, in denen V^* trotz $G \notin G3C$ den Beweis von P als gültig akzeptiert.

1. V könnte zufällig eine Kante wählen, für welche die Eigenschaft $\phi(u) \neq \phi(v)$ erfüllt ist. Diese tritt wie oben schon gezeigt mit der Wahrscheinlichkeit von höchstens $p_0 = \frac{|E|-1}{|E|}$ auf.

2. Da für die Eigenschaft der *Unambiguity* von Commitment-Schemes keine echte Eindeutigkeit, sondern nur eine berechnungsmäßige Ununterscheidbarkeit gefordert ist, kann mit einer begrenzten Wahrscheinlichkeit der Fall auftreten, daß der Prover eine Mehrdeutigkeit von Commitments vorgeben kann. Eine solche Mehrdeutigkeit darf in diesem Fall mit einer Wahrscheinlichkeit von höchstens $p_1 = \frac{1}{2 \cdot |E|}$ auftreten.

Somit liegt die Geamtwahrscheinlichkeit eines solchen Fehlers bei

$$p_0 + p_1 = \frac{|E| - 1}{|E|} + \frac{1}{2 \cdot |E|} = \frac{|E| - \frac{1}{2}}{|E|} \tag{1}$$

Da der Verifier im Fall $G \in G3C$ immer akzeptiert, gilt bei ausreichend vielen Widerholungen des Protokolls:

$$Pr[\langle P, V^* \rangle(G) = accept] < \frac{1}{3}$$

Zero-Knowledge In diesem Abschnitt soll gezeigt werden, daß $G3C \in CZK$ (Computational Zero-Knowledge). Hierbei wird als erstes gezeigt, daß die Wahrscheinlichkeit, daß die Simulator mit einer Wahrscheinlichkeit kleiner als $\frac{1}{2}$ das Fehlersymbol $\perp$ ausgibt, und als zweites wird gezeigt, daß unter der Voraussetzung, daß kein Fehler bei der Simulation auftritt, die Wahrscheinlichkeitsverteilungen von M^* und $\langle P, V^* \rangle$ berechnungsmäßig nicht zu unterscheiden sind.

Satz 1
Für jeden ausreichend großen Graphen $G = (V, E)$ ist die Wahrscheinlichkeit, für $M^*(G) = \perp$ kleiner als $\frac{1}{2}$.

Wir bezeichnen nun mit $p_{u,v}(G, r, (e_1, ..., e_2))$ die Wahrscheinlichkeit über allen Strings $s_1, ..., s_n \in \{0,1\}^n$, die V^*, bei Zufallsbits r und Ausgabe $(C_{s_1}(e_1), ..., C_{s_n}(e_n))$ des Provers, daß V^* mit (u, v) antwortet.

2 Lemma
Für jeden ausreichend großen Graphen $G = (V, E)$ gilt für jedes $r \in \{0,1\}^{q(n)}$, jede Kante $(u, v) \in E$ und alle möglichen Sequenzen von Farben $\alpha, \beta \in \{1,2,3\}^n$, daß die Bedingung

$$|p_{u,v}(G, r, \alpha) - p_{u,v}(G, r, \beta)| \leq \frac{1}{2|E|} \tag{2}$$

erfüllt ist.

Wir nehmen nun an, es existiert ein Polynom p und eine unendliche Folge von Integers, so daß für jede Zahl n aus der Sequenz ein n-Knoten Graph $G_n = (V_n, E_n)$, ein String $r_n \in \{0,1\}^{q(n)}$, eine Kante $(u, v) \in E_n$ und zwei Sequenzen $\alpha_n, \beta_n \in \{1,2,3\}$ existieren mit

$$|p_{u_n,v_n}(G_n, r_n, \alpha_n) - p_{u_n,v_n}(G_n, r_n, \beta_n)| > \frac{1}{p(n)}$$

Nun läßt sich eine Menge von Schaltkreisen $\{A_n\}$ konstruieren, die V^* und $G_n, r_n, u_n, v_n, \alpha_n, \beta_n$ beinhaltet. Nun läßt sich zeigen daß $\{A_n\}$ ein polynomiell großer Schaltkreis ist und A_n zwischen Schlüsseln für α_n und β_n unterscheiden kann, was nun zu einem Widerspruch zu der Secrecy des Commitment-Schemes geführt werden kann. □

Im folgenden wird dieses Lemma zum Beweis von Satz 1 genutzt, um die Wahscheinlichkeit für die Ausgabe von $\perp$ nach oben hin zu begrenzen. Sei $M_r^*(G)$ die Ausgabe von Maschine M^* bei Eingabe G unter der Bedingung, daß zu Beginn die Zufallsbits r gewählt wurden. Betrachte nun $Pr[M_r^*(G) = \perp]$. Sei $\overline{e} = (e_1, ..., e_n) \in \{1,2,3\}^n$ alle Möglichkeiten von Farbsequenzen. Es gilt dann:

$$\Pr[M_r^*(G) = \perp] = \sum_{\overline{e} \in \{1,2,3\}^n} \frac{1}{3^n} \cdot \sum_{(u,v) \in E_{\overline{e}}} p_{u,v}(G, r, \overline{e})$$

Sei nun $B_{u,v}$ die Menge von n-Tupeln $(e_1, ..., e_n) \in \{1,2,3\}^n$ welche $e_u = e_v$ erfüllen. Dann gilt:

$$\begin{aligned}
Pr[M_r^*(G) = \perp] &= \frac{1}{3^n} \cdot \sum_{\overline{e} \in \{1,2,3\}^n} \sum_{(u,v) \in E_{\overline{e}}} p_{u,v}(G, r, \overline{e}) \\
&= \frac{1}{3^n} \cdot \sum_{(u,v) \in E} \sum_{\overline{e} \in B_{u,v}} p_{u,v}(G, r, \overline{e}) \\
&\leq \frac{1}{3^n} \cdot \sum_{(u,v) \in E} |B_{u,v}| \cdot (p_{u,v}(G, r, (1, ..., 1)) + \frac{1}{2|E|}) \\
&= \frac{1}{6} + \frac{1}{3} \cdot \sum_{(u,v) \in E} p_{u,v}(G, r, (1, ..., 1)) \\
&= \frac{1}{6} + \frac{1}{3}
\end{aligned}$$

Daraus folgt nun Satz 1. □

Satz 3

Die Wahrscheinlichkeitsverteilung der Ausgabe von M^* unter der Bedingung, daß nicht $\perp$ ausgegeben wird, ist berechnungsmäßig nicht von der Wahrscheinlichkeitsverteilung von $\{view_{V^*}^{P_{G3C}}(G)\}_{G \in G3C}$ zu unterscheiden.

Sei $m^*(G)$ die Verteilung von $M^*(G)$ unter der Bedingung, daß $M^*(G) \neq \perp$. Für jeden Algorithmus A bezeichnen wir die unterscheidende Differenz von A, die die Verteilungen wie in Satz 3 betrachten mit $\varepsilon_A(G)$.

$$\varepsilon_A(G) := |Pr[A(m^*(G)) = 1] - Pr[A(\langle P, V^* \rangle(G)) = 1]|$$

Weiter seien nun folgende zwei Zufallsvariablen definiert:

- $\mu_{u,v}(G)$ beschreibt $M^*(G)$, unter der Bedingung, daß der Verfier die Kante $(u,v) \in E$ gewählt hat
- $\nu_{u,v}(G)$ beschreibt $\langle P, V^* \rangle(G)$ unter der Bedingung, daß der Verifier die Kante $(u,v) \in E$ gewählt hat.

Weiter sei $p_{u,v}$ die Wahrscheinlichkeit, daß in $m^*(G)$ der Verifier die Kante (u, v) gewählt hat, analog dazu sei $q_{u,v}$ die Wahrscheinlichkeit für $\langle P, V^*\rangle(G)$, daß der Verifier (u, v) gewählt hat. Nun lassen sich unter der Annahme, daß die Verteilungen doch berechnungsmäßig unterscheidbar sind, folgende zwei Fälle konstruieren:

1. Es exisitert eine nicht-vernachlässigbare Differenz zwischen der Wahl von V^* in $\langle, P, V^*\rangle(G)$ und V^* in $m^*(G)$. Dies bedeutet, es existiert ein Polynom p und eine unendliche Zahlenfolge, so daß für jede Zahl n aus dieser Folge ein n-Kanten Graph $G_n(V_n, E_n)$ mit einer Kante $(u_n, v_n) \in E_n$, daß gilt:

$$|p_{u_n,v_n}(G_n) - q_{u_n,v_n}(G_n)| > \frac{1}{p(n)}$$

2. Es existiert Ein Polynomialzeit-Algorithmus, der die oben genannten Wahrscheinlichkeitsverteilungen unterscheidet. Es existiert somit ein Algorithmus A und folgende Eigenschaften sind erfüllt:
   - $q_{u_n,v_n}(G_n) > \frac{1}{p(n)}$
   - $|p_{u_n,v_n}(G_n) - q_{u_n,v_n}(G_n)| < \frac{1}{3 \cdot p(n)^2}$
   - $|Pr[A(\mu_{u_n,v_n}(G_n)) = 1] - Pr[A(\nu_{u_n,v_n}(G_n))]| > \frac{1}{p(n)}$

   Fall 1 muß nicht weiter betrachtet werden, da er sofort zu einem Widerspruch führt, da sonst verschiedene Commitments für den Prover unterscheidbar wären und somit ein Widerspruch zur Definition der Commitment-Schemes darstellt. Im weiteren Verlauf des Beweises wird für den Fall 2 ein polynomieller Schaltkreis konstruiert, der die Verteilungen unterscheidet und somit Fall 2 zum Widerpruch geführt. Hierbei sei auf Literatur [1] verwiesen.

   Zusammen mit den vorhergehenden Sätzen folgt nun daraus $NP \subseteq CZK$ und der Beweis ist somit komplett. □

Literatur

[1] Oded Goldreich, Foundations of Cryptography, Fragments of a book, Februar 1995, zu finden unter http://theory.lcs.mit.edu/ oded/frag.html

[2] Oded Goldreich, Introduction to Complexity Theory - Lecture Notes, Juli 1999