Bingo Voting – technische Kurzbeschreibung

Stefan Röhrich

1 Das Wahlverfahren Bingo Voting

Das am E.I.S.S. entwickelte Verfahren Bingo Voting [BMQR07] setzt eine Wahlmaschine und einen vertrauenswürdigen Zufallszahlengenerator ein, um nachweislich korrekte Wahlen zu erreichen, bei denen ein Wähler nicht erpreßbar ist. Eine Wahl besteht dabei aus drei Phasen.

Vorbereitung In der Wahlvorbereitung werden für eine Wahl mit l Wahlberechtigten für jeden Kandidaten P_i l Zufallszahlen $r_1^i, \ldots, r_l^i$ erzeugt, daraus werden jeweils Commitments[1] auf (P_i, r_j^i) erzeugt („Dummy-Stimmen“), die veröffentlicht werden. Es wird bewiesen, daß für jeden Kandidaten genau gleich viele Commitments erzeugt wurde (z. B. durch Erzeugen von mehr Commitments, als eigentlich benötigt werden, und zufälliges Aufdecken. Wir empfehlen Randomized Partial Checking, wie es weiter hinten erklärt wird).

Wahl Die Wahl läuft wie in Abb. 1 beschrieben ab. Der Wähler wählt seinen Kandidaten an einer Wahlmaschine aus, danach erzeugt der vertrauenswürdige Zufallszahlengenerator eine Zufallszahl, zeigt diese an und übermittelt sie an die Wahlmaschine. Diese druckt einen Beleg aus, der für jeden Kandidaten (geordnet) den Kandidatennamen und eine Zufallszahl enthält: (P_i, s_i). Dabei muß beim gewählten Kandidaten die Zufallszahl des vertrauenswürdigen Zufallszahlengenerators in der Wahlmaschine stehen, der Wähler sollte dies überprüfen. Die anderen Zufallszahlen müssen Zahlen sein, auf die in der Vorbereitungsphase für den entsprechenden Kandidaten ein Commitment erzeugt wurde (d. h. $s_i = r_j^i$ für ein j). Der Beleg kann, wenn man die Korrektheit seiner Stimme nicht selber prüfen will, einem Wahlprüfungsverein o. ä. übergeben werden.

Nachbearbeitung Die Auszählung der Wahl besteht aus einem Veröffentlichen des Wahlergebnisses zusammen mit einem Beweis der Korrektheit. Dazu werden folgende Daten veröffentlicht:

- die Anzahl der Stimmen für die einzelnen Kandidaten
- alle erzeugten Belege

[1] Ein Commitment ist eine krypographische Operation, durch die man sich auf einen Wert festlegt, ohne diesen zu veröffentlichen, ein Beispielverfahen findet sich in Abschnitt 2.

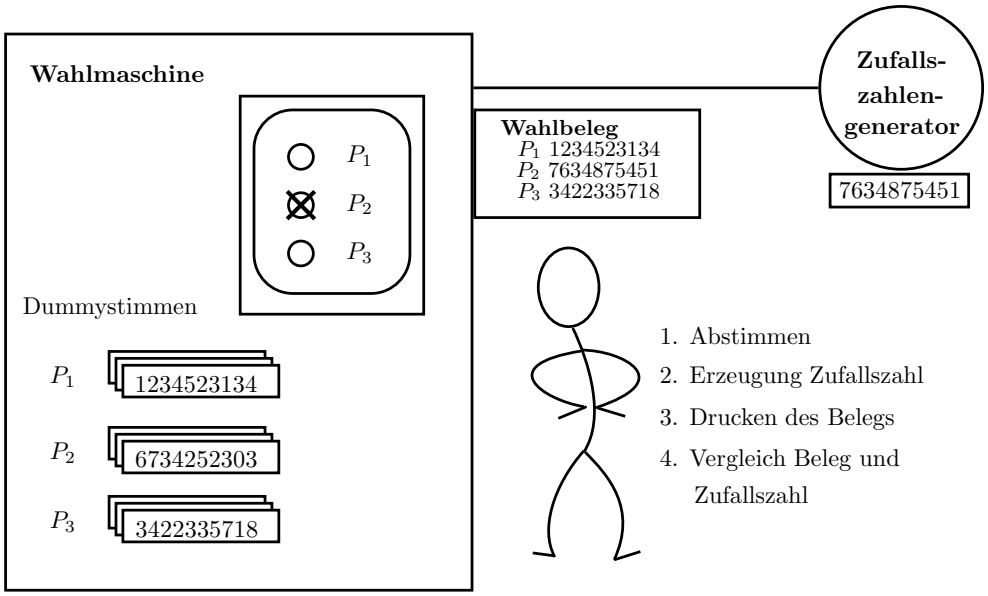

Abbildung 1: Wahlablauf bei Bingo Voting

- eine Liste der ungebrauchten (P_i, r_j^i) (zusammen mit der Information, die benötigt wird, um zu überprüfen, daß sich auf diese Werte zuvor festgelegt wurde)
- Beweis für die Korrektheit, das bedeutet v. a., daß jedes ungeöffnete Commitment auf genau einem Wahlbeleg verwendet wurde.

Anhand dieser veröffentlichten Daten kann nun die Korrektheit der Wahl überprüft werden, der Wähler kann dabei überprüfen, ob sein Beleg richtig veröffentlicht wurde. Für jede Stimme für einen Kandidaten wurde jeweils eine Dummy-Stimme nicht benötigt, d. h. die Anzahl der ungebrauchten Commitments minus der (für alle Kandidaten gleichen) Anzahl Wahlberechtigten, die nicht gewählt haben, ist die Stimmenzahl für den jeweiligen Kandidaten. Nun genügt es für die Korrektheit der Wahl zu beweisen, daß auf jedem Wahlbeleg Anzahl Kandidaten minus der einen echten Stimme Dummy-Stimmen erscheinen.

Dieser Beweis kann entweder durch einen großen Zero-Knowledge-Beweis geführt werden, oder er kann, wenn die Commitments eine zusätzliche Homomorphitätseigenschaft besitzen, über sogenanntes „Randomized Partial Checking“ [JJR02] erfolgen. Die zusätzliche Eigenschaft (die z. B. Pedersen-Commitments haben, s. Abschnitt 2) bedeutet, daß man Commitments in neue Commitments maskieren kann, die ein Commitment auf den alten Wert (unter Verwendung neues Zufalls) sind, dies läßt sich auch beweisen. Damit kann ein Beweis für die Korrektheit eines Stimmzettels nun wie folgt aussehen:

- Es wird ein neues Commitment (P_i, s_i) für die echte Stimme des Wählers mit der vom vertrauenswürdigen Zufallszahlengenerator erzeugten Zufallszahl s_i und dem gewählten Kandidaten P_i erzeugt.

- Die Commitments auf die für diesen Beleg verwendeten Dummy-Stimmen werden ausgewählt und zusammen mit dem neuen Commit veröffentlicht. Jeder kann überprüfen, daß dies ein neues und ansonsten alte Commitments aus der Vorbereitsphase sind.

- Die Commitments werden zu neuen Commitments mit dem gleichen Inhalt maskiert (s. Abschnitt 2) und zufällig permutiert, die neue Liste der Commitments wird veröffentlicht.

- Der letzte Schritt wird mit der neuen Commitmentliste wiederholt, so daß wir eine dritte Commitmentliste erhalten.

- Die Commitments der dritten Liste werden aufgedeckt, sie müssen den Inhalt des Belegs (bis auf die Reihenfolge) wiedergeben, d. h. jeder Kandidat kommt einmal vor und ihm zugeordnet ist die auf dem Beleg angegebene Zufallszahl.

- Es wird zufällig (externer Zufall) ausgewählt, ob für alle Commitments der zweiten Liste die Zuordnung zu den Commitments der ersten oder der dritten Liste aufgedeckt wird, und bewiesen, daß dies korrekt geschehen ist.

Dadurch kann die Korrektheit der Wahl garantiert werden durch die Voraussetzung eines vertrauenswürdigen Zufallszahlengenerators. Ein solcher Zufallszahlengenerator in der Wahlkabine könnte etwa vergleichbar einer Lottomaschine oder eines Bingo-Käfigs (daher der Name Bingo Voting) sein, diese Maschinen arbeiten auf recht einsichtige Weise, so daß unentdeckte Manipulationen sehr schwierig wären. Der Erpessungsschutz ist durch eine Ununterscheidbarkeit der Dummy-Stimmen für die anderen Kandidaten und der echten Zufallszahl für den gewählten erreicht.

Das Verfahren stellt keine hohen Anforderungen an den Wähler, er muß nur wählen und sollte zwei Zahlen vergleichen, die restliche Verifikation läuft öffentlich. Allerdings wird einiges an organisatorischem Aufwand für die Wahlvorbereitung und die Sicherung der Dummy-Stimmen vor unbefugtem Zugriff benötigt.

2 Pedersen-Commitments

Commitment-Verfahren bilden einen wichtigen Grundbaustein für viele kryptographische Protokolle, auch für die Realisierung von Bingo Voting wird ein Commitment-Vefahren benötigt, dafür können z. B. Pedersen-Commitments [Ped91] verwendet werden. Commitment-Verfahren bestehen normalerweise aus zwei Operationen:

1. *Commit/Festlegen*: Es wird aus einer Nachricht m ein Wert berechnet, der veröffentlicht werden kann. Aus diesem Wert kann man nicht auf die Nachricht schließen, man ist aber auf die Nachricht festgelegt.

2. *Unveil/Aufdecken*: Durch Veröffentlichen einer Zusatzinformation (z. B. m selbst und Zufallswahlen) wird m öffentlich und es kann überprüft werden, daß sich im Commit-Schritt wirklich auf m festgelegt wurde.

Intuitiv kann man sich den Commit-Schritt so vorstellen, daß m in einen öffentlichen Tresor gelegt wird, der Schlüssel aber bei der commitenden Partei bleibt. Im Unveil-Schritt wird dann der Schlüssel herausgegeben.

Normalerweise betrachtet man bei Commitments zwei Sicherheitsforderungen: sie müssen *concealing* oder *hiding* (verbergend) sein, d. h. aus der nach dem Commit-Schritt veröffentlichten Information kann nicht auf die Nachricht geschlossen werden, und sie müssen *binding* (bindend) sein, d. h. im Unveil-Schritt ist es nicht möglich, ein Commitment auf eine Nachricht m korrekt für eine Nachricht m' mit $m \neq m'$ zu öffnen. Diese Forderungen sind unter verschiedenen Annahmen erfüllbar.

Pedersen-Commitments beruhen nun auf der Annahme, daß das diskrete Logarithmus-Problem in einer zyklischen Gruppe G von Ordnung q schwer ist. $h, g \in G$ seien Erzeuger von G, so daß der Commitende $\log_g h$ nicht kennt. Um sich nun auf eine Nachricht $m \in \mathbb{Z}_q$ zu committen, wählt man ein zufälliges $r \in \mathbb{Z}_q$, das Commitment ist dann $h^m g^r$. Das Aufdecken geschieht durch Veröffentlichen von m und r, der Empfänger der Nachricht kann dann überprüfen, ob er damit denselben Wert wie das ursprüngliche Commitment errechnet. Das Commitment ist unter der DLOG-Annahme bindend; denn könnte es auf einen anderen Wert m' unter Verwendung von r' aufgedeckt werden, könnte man $\log_g h = \frac{r-r'}{m'-m}$ berechnen, da $h^{m'} g^{r'} = h^m g^r$ und somit $h^{m'-m} = g^{r-r'}$. Die Geheimhaltung ist sogar informationstheoretisch garantiert, da g^r ein zufälliges Element von G ist, wenn r zufällig gezogen wurde, damit ist auch $h^m g^r$ ein zufälliges Element von G.

Für Bingo Voting wird außerdem eine Maskierungseigenschaft benötigt, dies kann wie im folgenden beschrieben erreicht werden. Sei $c = h^m g^r$ ein Commitment auf m. Wähle ein zufälliges $s \in \mathbb{Z}_q$, $c' = cg^s = h^m g^{r+s}$ ist dann ein maskiertes Commitment c immer noch auf die Nachricht m. Durch Veröffentlichen von s kann gezeigt werden, daß c und c' Commitments auf m sind, ohne m selbst aufzudecken.

Literatur

[BMQR07] Bohli, Jens-Matthias, Jörn Müller-Quade und Stefan Röhrich: *Bingo Voting: Secure and Coercion-Free Voting Using a Trusted Random Number Generator.* In: Alkassar, A. und M. Volkamer (Herausgeber): *VOTE-ID 2007*, Band 4896 der Reihe *Lecture Notes in Computer Science*, Seiten 111–124. Springer-Verlag, 2007.

[JJR02] Jakobsson, Markus, Ari Juels und Ronald L. Rivest: *Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking.* In: *USENIX Security Symposium*, Seiten 339–353, 2002.

[Ped91] Pedersen, Torben Pryds: *Non-interactive and Information-Theoretic Secure Verifiable Secret Sharing.* In: Feigenbaum, Joan (Herausgeber): *Advances in Cryptology – CRYPTO '91: Proceedings*, Band 576 der Reihe *Lecture Notes in Computer Science*, Seiten 129–140. Springer, 1991.