Implementierbare Zustandsübergänge eines formalen IT-Sicherheitsmodells für Online-Wahlsysteme

Rüdiger Grimm und Melanie Volkamer

Institut für Wirtschafts- und Verwaltungsinformatik
Universität Koblenz-Landau
Universitätsstraße 1, 56070 Koblenz

Institut für IT-Sicherheit und Sicherheitsrecht
Universität Passau
Innstraße 43, 94032 Passau

grimm@uni-koblenz.de
volkamer@uni-passau.de

Abstract: Als erster Schritt für die formale Sicherheitsmodellierung von Online-Wahlsystemen werden zwei Sicherheitsziele beispielhaft ausgewählt, nämlich die Anforderung, dass kein unbefugter Wähler eine Stimme abgeben darf, und die Anforderung, dass jeder berechtigte Wähler genau einmal wählen darf und sein Wahlrecht erst dann verliert, wenn er eine Stimme abgegeben hat. Es wird gezeigt, wie diese Sicherheitsziele in sicheren Systemzuständen repräsentiert werden können und welche Zustandübergangsregeln das Erreichen dieser sicheren Zustände garantieren. Da es sich um einen ersten Schritt zur Modellierung von Online-Wahlsystemen handelt, darf die Definition der Systemzustände und -übergänge nicht andere Sicherheitsziele blockieren, die in weiteren Schritten zu formalisieren sind, zum Beispiel das Wahlgeheimnis. Es wird gezeigt, dass die Modellierung der obigen Anforderungen nicht im Widerspruch zur Implementierung des Wahlgeheimnisses steht.

1 Die Aufgabe: ein hochwertiges Schutzprofil nach Common Criteria

Die Common Criteria [CC06] standardisieren eine Vorgehensweise zur Spezifizierung von Sicherheitsanforderungen an IT-Produkte in Form von so genannten Schutzprofilen. Diese können auf Vollständigkeit und Konsistenz geprüft und zertifiziert werden. Im Mai 2008 hat das Bundesamt für Sicherheit in der Informationstechnik (BSI) einen Basissatz von Sicherheitsanforderungen an Online-Wahlprodukte als Schutzprofil [PP08] zertifiziert. Darin sind die Sicherheitsanforderungen nach den Regeln der CC in einer relativ niedrigen Vertrauenswürdigkeitsstufe (EAL2+) formuliert worden. Für eine höhere Vertrauenswürdigkeitsstufe (EAL 6 oder 7) ist eine formale Beschreibung der Sicherheitsanforderungen (und damit implizit auch der Sicherheitsziele) im Rahmen eines formalen Sicherheitsmodells erforderlich [M+02]. Das übliche Verfahren zur Formulierung von Sicherheitszielen ist die formale Spezifikation von sicheren Systemzuständen und erlaubten Zustandsübergängen, von denen nachzuweisen ist, dass sie von sicheren Zuständen wieder zu sicheren Zuständen führen [Gr08].

Dieser Beitrag wird nicht die Einzelheiten eines solchen Modells ausführen und insbesondere nicht den mathematischen Nachweis des Sicherheitstheorems, welches besagt, dass die erlaubten Zustandsübergänge das System innerhalb sicherer Zustände halten. Diese Einzelheiten sind bereits an anderer Stelle ausgeführt: im Tagungsband der IRIS 2008 [GV08a], sowie mit einem etwas theoretischeren Gewicht und in englischer Sprache in EVote 2008 [GV08b].

Hier nun soll der grundsätzliche Gedankengang zur Entwicklung eines formalen Sicherheitsmodells für Online-Wahlen nachgezeichnet werden, wobei ein besonderes Gewicht auf die Implementierbarkeit der Zustandsübergangsregeln und auf die Verträglichkeit der Modellwahl mit zukünftigen Formalisierungen weiterer Sicherheitsziele gelegt wird. Es wird insbesondere gezeigt, dass die hier gewählte Modellierung ohne Verlust des Wahlgeheimnisses durchgesetzt werden kann. Folglich ist der weitere Weg zur Formalisierung des Sicherheitsziels einer geheimen Wahl offen gehalten.

Im folgenden Kapitel 2 wird der Begriff des (formalen) IT-Sicherheitsmodells erläutert. Nach der Auswahl von zwei Sicherheitszielen und ihrer Modellierung in Kapitel 3 werden dann in Kapitel 4 die Implementierbarkeit der zugehörigen Sicherheits-mechanismen und ihre Verträglichkeit mit dem Wahlgeheimnis diskutiert.

2 Formale IT-Sicherheitsmodelle

Eine gängige Methode zur Modellierung von IT-Sicherheitsanforderungen ist die Spezifikation von sicheren Systemzuständen und einer Übergangsregel, die dafür sorgt, dass sichere Zustände wieder zu sicheren Zuständen führen [Gr08], siehe zum Beispiel das Bell-LaPadula-Modell zur sicheren Kommunikation klassifizierter Information in hierarchischen Systemumgebungen [BL73]. Zunächst sind die sicheren Systemzustände zu spezifizieren und es ist zu zeigen, dass sie die gewünschten Sicherheitsziele repräsentieren. Dann sind die Zustandsübergangsregeln zu formulieren, welche sichere Zustände nur in sichere Zustände übergehen lassen. Dabei ist zu beachten, dass diese Zustandsübergänge eine realistische Implementierung von Sicherheitsmechanismen erlauben. Im Fall einer formalen Modellierung ist in einem formalen Sicherheitstheorem zu beweisen, dass ein sicherer Zustand unter Beachtung der Übergangsregeln wieder in einen sicheren Zustand führt.

Ein IT-Sicherheitsmodell hat also zwei Lücken zu schließen, nämlich die Lücken zwischen den sicheren Systemzuständen und dem übergeordneten Sicherheitsziel und zwischen den erlaubten Zustandsübergängen und den sicheren Systemzuständen. Die erste Lücke schließt das vorliegende Schutzprofil [PP08] durch die „Security Problem Definition“ in Form von Annahmen, Bedrohungen und organisatorischen Sicherheitspolitiken, sowie durch die „Erklärung der Sicherheitsziele“. Die zweite Lücke wird durch das formale Sicherheitstheorem mit seinem Nachweis geschlossen, dass erlaubte Zustandsübergänge einen sicheren Systemzustand wieder in einen sicheren Systemzustand überführen [GV08a, GV08b]. Diese beiden Lücken sind durch die jeweils zitierten Arbeiten geschlossen.

Ein wichtiges Kriterium für die Qualität eines IT-Sicherheitsmodells ist die Implementierbarkeit der Zustandsübergangsregeln. Denn diese sollen ja eine Perspektive für die Sicherheitsmechanismen liefern, die die Sicherheitsanforderungen des Modells erfüllen, indem sie durchsetzen, dass das System in sicheren Zuständen verbleibt. Die Mechanismen dürfen natürlich nicht im Widerspruch zu anderen Sicherheitszielen stehen, die gleichzeitig zu erfüllen wären, vor allem zum Wahlgeheimnis. Das ist Gegenstand dieses Beitrags in den folgenden Kapiteln 3 und 4. Dabei wird zunächst das Modell für zwei ausgewählte Sicherheitsziele vorgestellt und dann gezeigt, dass hier kein Konflikt mit anderen Sicherheitszielen zu erwarten ist.

3 Formale Modellierung ausgewählter Sicherheitsziel

3.1 Auswahl der Sicherheitsziele

Eine vollständige Auflistung von Sicherheitszielen für einen „Basissatz von Sicherheitsanforderungen an Online-Wahlprodukte“ ist in [PP08] ausgeführt. Das formale Sicherheitsmodell in [GV08a, GV08b], das wir hier nachzeichnen, ist nur ein erster Ansatz, der mit zwei ausgewählten Sicherheitszielen des Schutzprofils beginnt. Auf diese Weise wird aufzeigt, wie auch die weiteren Sicherheitsziele formal zu spezifizieren sind. Die beiden ausgewählten Sicherheitsziele sind:

O.UnbefugterWähler: Am EVG können nur Wähler mit Stimmberechtigung, die vom EVG eindeutig identifiziert und authentisiert werden, eine Stimme abgeben und damit einen Stimmdatensatz in der Urne speichern.

O.OneVoterOneVote: Der EVG stellt sicher, (A) dass jeder Wähler mit Stimmberechtigung nur eine Stimme abgeben kann und (B) dass er seine Stimmberechtigung nicht verliert, ohne eine Stimme abgegeben zu haben.

Bei der Modellierung ist es wichtig sicherzustellen, dass eine Erweiterung auf andere Sicherheitsziele und hierbei insbesondere die Modellierung der geheimen Wahl nicht blockiert wird..

3.2 Allgemeiner Systemzustand

Ein Systemzustand wird allgemein durch ein Tripel (W, S, *voter*) dargestellt, wobei gilt:

1. W – Menge der Wahlberechtigten, die noch eine Stimme abgeben dürfen
2. S – Menge der (verschlüsselten) Stimmen in der elektronischen Urne.
3. *voter:* $S \rightarrow M$ – Abbildung von Stimmen zum zugehörigen Wähler. Dabei ist M eine Obermenge $M \supseteq W_{total}$ von Systemnutzern, die auf das System zugreifen, egal, ob mit oder ohne Stimmberechtigung.

Die Wahldurchführung wird nun als eine Folge von Zuständen (W_i, S_i, $voter_i$), (W_{i+1}, S_{i+1}, $voter_{i+1}$), (W_{i+2}, S_{i+2}, $voter_{i+2}$), … modelliert, wobei der Übergang vom (i)-ten zum (i+1)-ten Zustand durch eine erfolgreiche oder erfolglos versuchte Abgabe einer Stimme zustande kommt.

Die Einführung der Abbildung *voter* ist kritisch! Wenn sie für die gesamte Wahldurchführung auch nach der endgültigen Stimmabgabe praktisch auswertbar sein soll, müsste die Wahlleitung (oder jedenfalls eine unter ihrer Kontrolle befindliche Funktion) in der Lage sein, Stimmen, die in der Urne liegen, ihren Wählern zuzuordnen. Das wäre ein Bruch des Wahlgeheimnisses, sofern die Stimmen in der Urne lesbar sind. Für die Wahrung des Wahlgeheimnisses ist es im Gegenteil erforderlich, die Abbildung *voter* nicht mehr praktisch auf die in der Urne befindlichen lesbaren Stimmen anwenden zu können.

Allerdings gibt es eine bestimmte – kurze – Phase während der Wahldurchführung, in der die Abbildung *voter* sehr wohl und ohne Bruch des Wahlgeheimnisses praktisch ausgewertet werden kann und soll: Bei der Präsenzwahl im Wahllokal wird geprüft, ob der Wähler, der gerade eine verdeckte Stimme in die Urne einwirft, dazu berechtigt ist oder nicht. Das heißt, dass die Abbildung *voter* auf den verdeckten Stimmzettel angewendet wird, bevor sie unter Kontrolle der Wahlbeobachter in die Urne geworfen wird. Dabei wird festgestellt, ob $voter(s) \in W_{total}$ oder $voter(s) \in M \setminus W_{total}$. Falls $voter(s) \in M \setminus W_{total}$, wird die Stimme s nicht in die Urne weitergeleitet, sondern vernichtet. Falls $voter(s) \in W_{total}$, wird die Stimme s in die Urne geworfen. Nachdem die Stimme in der Urne versenkt ist, ist der konkrete Wählerbezug zwar noch vorhanden, aber nicht mehr feststellbar. Und bei der Briefwahl wird die Funktion *voter* durch den äußeren Briefumschlag realisiert, die den Absender aufzeigt. Erst bei der Auszählung wird die Absenderadresse auf dem äußeren Umschlag geprüft, d.h. es wird festgestellt, ob $voter(s) \in W_{total}$ oder $voter(s) \in M \setminus W_{total}$, Danach wird entweder der gesamte Brief vernichtet, falls $voter(s) \in M \setminus W_{total}$, oder der äußere Briefumschlag wird entfernt und die durch einen unbeschrifteten inneren Umschlag anonymisierte Stimme in die Briefwahlurne versenkt, falls $voter(s) \in W_{total}$.

Nach dieser Überprüfung der berechtigten Herkunft der Wahlstimme s ist die Wahlstimme zwar immer noch ihrem (als berechtigt erkannten) Wähler zugeordnet, aber durch die Versenkung der Stimme in der Urne, bzw. durch die Entfernung des äußeren Briefumschlags bei der Briefwahl kann diese Zuordnung nun niemand mehr ausführen. Die Wahlbeobachter vertrauen lediglich darauf, dass die Zuordnung nach wie vor jedenfalls auf einen berechtigten Wähler verweist, wenn man auch nicht mehr weiß, auf welchen.

In Bezug auf die Abbildung *voter* bedeutet das, dass das Abbild von *voter* nur für die jetzt gerade abzugebende Stimme, d.h. für $s \in S_{i+1} \setminus S_i$ feststellbar sein darf, und auch nur, wenn sie gleichzeitig nicht gelesen werden kann. Nachdem s anonymisiert wird, ist der Link zwischen Wähler und seiner Stimme nicht mehr praktisch rekonstruierbar. Dann darf sie auch lesbar sein. Daher sollte die Zuordnung $voter_{i+1}$ praktisch nur während des Zustandsübergangs auf der „sichtbaren“ Untermenge $S_{i+1} \setminus S_i$ von S_{i+1} genutzt werden, solange die Stimme noch verdeckt ist. Das wird unten in Abschnitt 3.5 über die „erlaubten Zustandsübergänge“ genau so berücksichtigt werden. Für den „unsichtbaren“ Teil S_i der Stimmen in der Urne definieren wir $voter_{i+1}/S_i := voter_i$.

3.3 Sicherer Systemzustand

Die beiden in Abschnitt 3.1 ausgewählten Sicherheitsziele „O.UnbefugterWähler“ und „O.OneVoterOneVote“ lassen sich folgendermaßen als formale Systemzustände formuliert:

O.UnbefugterWähler: $\forall\ s \in S:\ voter(s) \in W_{total;}$.

O.OneVoterOneVote (A): $\forall\ s, s' \in S:\ voter(s) = voter(s') \Rightarrow\ s = s'$

O.OneVoterOneVote (B): $\forall\ x \in W_{total} \backslash W: \exists\ s \in S:\ voter(s) = x$

3.4 Anfangszustand

Als Anfangszustand wird definiert: ($W_0 = W_{total}$, $S_0 = \{\}$, $voter_0 = \{\}$). Dabei steht W_{total} für die Menge aller Wahlberechtigten im Wählerverzeichnis.. Die leeren Mengen S_0 und $voter_0$ repräsentieren die leere Urne zu Beginn der Wahl und die leere Abbildung von der leeren Urne auf die Benutzer des Systems.

3.5 Erlaubter Zustandsübergang

Ein Zustandsübergang von Zustand (W_i, S_i, $voter_i$) nach (W_{i+1}, S_{i+1}, $voter_{i+1}$) ist zulässig, wenn eine der beiden folgenden Regeln eingehalten wird:

[Regel 1] gilt für einen Zustandsübergang, bei dem keine Stimme abgegeben wird, wobei sich die Mengen S und W nicht ändern: $W_i = W_{i+1} \wedge S_i = S_{i+1} \wedge voter_i = voter_{i+1}$

[Regel 2] gilt für einen Zustandsübergang, bei dem eine Stimme erfolgreich abgegeben wird, wobei sich die Mengen S und W ändern:
$\exists\ s \in S_{i+1} : (voter_{i+1}(s) \in W_i \wedge W_{i+1} = W_i \backslash \{voter_{i+1}(s)\} \wedge S_i = S_{i+1} \backslash \{s\}\ )$

Es ist zu beachten, dass die Zustandsübergangsregel die Abbildung *voter* nur auf den sichtbaren Bereich der Urne anwendet, in dem die Stimme noch verdeckt sein kann (und muss), d.h. auf $S_{i+1} \backslash S_i$. Dadurch wird die Zustandsübergangsregel nutzbar für den Einsatz in der Praxis, das wird im Folgenden Kapitel 4 diskutiert.

Mithilfe der Definition der sicheren Zustände und der Zustandsübergangsregeln lässt sich nun das entsprechende Sicherheitstheorem formal beweisen:Der Beweis des Theorems ist in [GV08b] vollständig ausgeführt.

4 Diskussion der Zustandsübergangsregeln

Für die Implementierung der Zustandsübergangsregeln ist nur eine einzige Sache zu überprüfen, nämlich ob die in Frage kommende Stimme *s*, die noch nicht in S_i liegt, dazu berechtigt ist, in S_{i+1} zu liegen. Das Berechtigungskriterium wird in Regel 2 formuliert: $voter_{i+1}(s) \in W_i$. Die Prüfung kann analog zur Praxis bei physischen Präsenzwahlen und bei Briefwahlen durchgeführt werden. Die Stimme *s* wird verdeckt vorgezeigt (gefaltet bei der Präsenzwahl, im unbeschrifteten Umschlag bei der Briefwahl) und von der Wahlbeobachtung eindeutig ihrem Wähler $voter_{i+1}(s)$ zugeordnet. Hier steht der Index auf (i+1), da man sich im (i)-ten Zustand befindet und der Wahlversuch einen Übergang zum (i+1)-ten Zustand vorbereitet.

Falls $voter_{i+1}(s) \in W_i$ festgestellt ist, müssen die beiden anderen Eigenschaften von Regel 2 durchgesetzt werden: Der Wähler $voter_{i+1}(s)$ wird aus der Liste der Personen gestrichen, die noch nicht abgestimmt haben und deshalb noch abstimmen dürfen, wodurch das aktuelle Verzeichnis der (noch) stimmberechtigten Wähler in den (i+1)-ten Zustand überführt wird: $W_{i+1} = W_i \setminus \{voter_{i+1}(s)\}$. Und die Stimme *s* wird der Urne hinzugefügt, wodurch sie in den (i+1)-ten Zustand überführt wird: $S_{i+1} = S_i + \{s\}$. Falls dagegen $voter_{i+1}(s) \notin W_i$ festgestellt ist, muss die Regel 1 durchgesetzt werden: Das Verzeichnis der (noch) stimmberechtigten Wähler und die Urne bleiben unberührt: $W_i = W_{i+1} \wedge S_i = S_{i+1} \wedge voter_i = voter_{i+1}$.

Die Zustandsübergangsregel ist in all ihren Teilen implementierbar, denn es gilt folgendes: Das Hinzufügen eines Datensatzes in die Urnendatei, bzw. das Entfernen eines Datensatzes aus der Wählerdatei sind elementare Operationen. Die Überprüfung der Wahlberechtigung ist ebenfalls eine implementierbare Operation, die aus mehreren Schritten zusammengesetzt wird, darunter der Vergleich einer vorgelegten Wähleridentität mit einer Liste berechtigter Wähler $W_i \subseteq W_{total}$. Der Vergleich kann sehr aufwändig sein, wenn weitere Sicherheitsziele, etwa die Authentizität des Wählers, berücksichtigt werden.

Das Sicherheitstheorem beweist zwar, dass die Zustandsübergangsregeln die hier ausgewählten Sicherheitsziele absichern. Aber darüber hinaus muss die Frage beantwortet werden, ob die Implementierung der Regeln anderen Sicherheitsanforderungen, deren Modellierung noch aussteht, widerspricht. Das muss für jedes einzelne Sicherheitsziel aus [PP08] einzeln behandelt werden. An dieser Stelle sei nur der Vergleich mit dem wichtigen Sicherheitsziel der geheimen Wahl angestellt: Da die Übergangsregel die Abbildung *voter* nur auf den Bereich $S_{i+1} \setminus S_i$ anwendet, kann sie in der Phase ausgeführt werden, in der die Stimme noch nicht in der Urne liegt, das heißt, sie kann mit der verdeckten Stimme überprüft werden. Die Übergangsregel bleibt gewahrt, wenn nach der Überprüfung der Wahlberechtigung $voter_{i+1}(s) \in W_i$ die Auswertung der Funktion $voter_{i+1}(s)$ unmöglich gemacht wird: Das Sicherheitstheorem besagt, dass die Zuordnung auch dann noch integer ist, wenn man sie nicht mehr praktisch auswerten kann.

Damit garantiert eine Implementierung der Zustandsübergangsregel die Wahrung der Integrität der Stimmberechtigung auch ohne Zuordnung der Stimmen in der Urne zu ihren Wählern, indem sie den Prüfungsprozess auf die Phase vor der Stimmenabgabe und auf die verdeckt Stimme beschränkt. Damit ist das Wahlgeheimnis durch die Sicherheitsmechanismen, die hier zu implementieren sind, nicht gefährdet.

Aufbauend auf diesen Ergebnissen sind nun weitergehende Sicherheitsziele zu modellieren und mit den hier spezifizierten Zuständen und Übergangsregeln zu einem größeren Modell zu integrieren.

Literaturverzeichnis

[CC06] Common Criteria for Information Technology Security Evaluation, Version 3.1, 2006.

[BL73] Bell, D.E.; LaPadula, L.J.: Secure Computer Systems: Mathematical Foundations, and A mathematical model. ESD-TR-73-278, MTR-2547, Vols 1&2. The MITRE Corporation, Bedford, MA, Nov 1973.

[Gr08] Grimm, R.: IT-Sicherheitsmodelle. WISU – das Wirtschaftsstudium. Herausgeber: Hartmann-Wendels, Thome, Woll, http://www.wisu.de, wisu 5/08, Lange Verlag, Düsseldorf, Mai 2008, S. 720-727.

[GV08a] Grimm, R.; Volkamer, M.: Entwicklung eines formalen IT-Sicherheitsmodells für Online-Wahlsysteme. Tagungsband der Tagung „Internationales Rechtsinformatik Symposion“, IRIS 2008. Universität Salzburg, 21.-23. Februar 2008.

[GV08b] Grimm, R.; Volkamer, M.: Development of a Formal IT-Security Model for Remote Electronic Voting Systems. EVote08, Bregenz, 6.-9. August 2008.

[M+02] Mantel, H.; Stephan, W.; Ullmann, M.; Vogt, R.: Leitfaden für die Erstellung und Prüfung formaler Sicherheitsmodell im Rahmen von ITSEC und Common Criteria. Version 1.0c http://david.von-oheimb.de/cs/teach/BSI-Leitfaden_1.0c.pdf, 2002.

[PP08] Volkamer, M.; Vogt, R.: Basissatz von Sicherheitsanforderungen an Online-Wahlprodukte. Common Criteria Schutzprofil BSI-CC-PP-0037, Version 1.0, 18. April 2008. BSI-Zertifikat erteilt im Mai 2008.