Elektronische Wahlen unter dem Einsatz kryptografischer Observer

Inaugural-Dissertation
zur Erlangung des Grades
„Doktor der Naturwissenschaften"

eingereicht beim
Fachbereich Mathematik und Informatik, Physik, Geographie
Justus-Liebig-Universität Gießen

von
Jörn Schweisgut
aus Alsfeld

Gutachter: Prof. Dr. A. Beutelspacher
Prof. Dr. J. Schwenk

Gießen, April 2007

Dekan:	Prof. Dr. Bernd Baumann
1. Berichterstatter:	Prof. Dr. Albrecht Beutelspacher (Gießen)
2. Berichterstatter:	Prof. Dr. Jörg Schwenk (Bochum)

Datum der Disputation: 27.07.2007

English Abstract

Long candidate lists and the possibility of vote-splitting and cross voting like in the local elections of Hesse in March 2006 strengthen the voter's position and his dimension of participation on the one hand; but they lead to a much more complex and expensive tallying. Electronic voting schemes may on the other hand reduce the expenditure of time for the tallying.

In the last years, several elections have been executed electronically, e.g. the election of the youth council in Esslingen, Fellbach, Bobenheim-Roxheim and Filderstadt, the works council of T-Systems, T-Com and Deutschen Telekom, the election of the executive board of the Gesellschaft für Informatik. These are only a few examples documenting a broadening trend towards electronic voting. But electronic elections do not only simplify the tallying or the voting, it also imports the risk to ease abuse and to permit large scale manipulation.

A great many of papers have been published in cryptographic literature describing how to obtain robust and verifiable election schemes. Hirt and Sako [HS00] presented the first electronic voting scheme in which voters were not able to prove their voting decision in 2000. This so-called receipt-freeness was achieved under the unrealistic assumption of an untappable channel from each authority to each voter. To solve this problem, Magkos et al. [MBC01] introduced an election scheme in 2001 based on a tamper-proof device, a so-called observer.

In this thesis it will be shown that the system of Magkos et al. is not receipt-free, the flaw will be fixed and the system will be improved. Then another receipt-free voting scheme with observer which combines the advantages of the Hirt and Sako scheme with the one of Magkos et al. will be presented.

Besides the long unsolved problem of receipt-freeness, there are further possibilities for an attack on electronic elections described by Juels et al. in [JCJ05] in 2005. They summed up these attacks by the notion of coercion-resistance and proposed a first coercion-resistant voting scheme. In this thesis, an election scheme based on the usage of credentials as a proof of authorisation to vote will be introduced. In this scheme the credentials are encrypted during registration. By using a MIX-cascade we can omit one time-consuming plaintext equivalence test from the tallying process. In addition, the observer facilitates registration and voting for the benefit of the voter. Pseudonymisation of the ciphertexts during the voting period implies a permanent secrecy of the submitted votes.

Inhaltsverzeichnis

A citizen of America will cross the ocean to fight for democracy, but won't cross the street to vote in a national election.

Bill Vaughan

Kapitel 1

Einleitung

1.1 Bedeutung demokratischer Wahlen

Das Wort *Wahlen* geht auf das indogermanische Wort *uel* zurück, das soviel wie *wollen* bedeutet. In der Tat entspricht eine Wahl einer Willenskundgebung derer, die wählen.

Wahlen und die Art ihrer Durchführung sind ein wesentliches Merkmal und ein notwendiger Bestandteil jeder Demokratie. Im Gegensatz zu Ländern mit totalitären Regimen, in denen keine Wahlmöglichkeit zwischen verschiedenen politischen Richtungen besteht, beruht die demokratische Ordnung in der Bundesrepublik auf dem Recht des Volkes, durch Wahlen regelmäßig über die Machtverteilung im Staat zu entscheiden.

In der Demokratie übt der mündige Bürger „Staatsgewalt" aus, indem er wählt, also von seinem aktiven Wahlrecht Gebrauch macht. Rein technisch betrachtet, sind Wahlen Mittel zur Bestellung von Personen in ein Amt oder zur Bildung von Körperschaften. Dazu bedarf es eines bestimmten Verfahrens, einer Bestellungstechnik und eines Wahlsystems. Diese müssen durch eine gesetzliche oder eine verfassungsrechtliche Regelung beschlossen werden. Werden bei dem Verfahren elektronische Medien wie z. B. das Internet genutzt, so spricht man von *elektronischen Wahlen*.

In dieser Arbeit werden dabei nur sogenannte *remote e-voting* Verfahren erstellt, betrachtet und analysiert, bei denen der Wähler seine Stimme über ein offenes Netz wie dem Internet abgeben kann. Es werden keine Wahlmaschinen in Wahllokalen betrachtet.

Auf europäischer Ebene gibt es bisher lediglich eine Empfehlung für die gesetzlichen, organisatorischen und technischen Standards für elektronische Wahlen (vgl. [Eur05]). In Deutschland existieren bislang keine gesetzlichen Anforderungen. Es ist aber ein sich verbreitender Trend zur Entwicklung, Analyse aber auch zum Einsatz elektronischer Wahlen zu erkennen. Das Bundesamt für Sicherheit in der Informationstechnik erstellt zurzeit ein Schutzprofil [BSI06] für die grundlegenden Anforderungen an elektronische Wahlen.

Trotz der fehlenden speziellen gesetzlichen Anforderungen für elektronische Wahlen wurden in den vergangenen Jahren zahlreiche Wahlen in Deutschland elektronisch durchgeführt. Zu nennen sind unter anderem die Jugendgemeinderatswahlen in Esslingen [ME01], Fellbach

[Fel01], Bobenheim-Roxheim [BR01] und Filderstadt [Fil01] 2001, die Betriebsratswahlen bei der T-Systems 2002 und 2005 [Wed06], bei der T-Com und Deutschen Telekom 2006 [Die06], die Vorstandswahlen bei der Gesellschaft für Informatik 2004 [GI04] und 2005 [Win05] und bei der Initiative D21 in den Jahren 2003 und 2005 [Ahr05], die Hochschulwahlen in Hannover [Pet00] 2000 sowie in Bremerhaven 2001 [Bec01] und Bremen 2003 [Cyb03]. Aber auch bei politischen Wahlen beispielsweise bei der Landratswahl im Kreis Marburg-Biedenkopf 2001 [MSM+01] hatten Wähler versuchsweise die Möglichkeit ihre Stimme elektronisch abzugeben.

Stimmzettel, wie sie zum Beispiel bei der hessischen Kommunalwahl am 26.03.06 benutzt wurden, bieten zwar dem Wähler eine Vielzahl von weiteren Wahlmöglichkeiten und erhöhen den Einfluss der Wähler auf die Zusammensetzung der Parlamente und damit auf die Machtverteilung, aber sie steigern den Aufwand bei der Stimmabgabe und insbesondere bei der Stimmauszählung erheblich. Daher bietet sich eine Vereinfachung der Abstimmung und Auszählung durch die Nutzung von elektronischen Medien, wie dem Internet oder anderer Computernetze, an.

Hinsichtlich der Wahlbeteiligung belegt die Bundesrepublik von allen Demokratien, die keine Wahlpflicht kennen, einen der vorderen Plätze (vgl. [PG02]). Dabei war die Beteiligung der Jungwähler allerdings sehr gering. Eine Nutzung von Computernetzen und anderen elektronischen Medien für Wahlen würde gerade den jungen Menschen, die mit der Anwendung dieser Medien vertraut sind, die Stimmabgabe erleichtern und vermutlich die Wahlbeteiligung erhöhen. Ob sich die Wahlbeteiligung tatsächlich erhöht, hängt jedoch von der Art der Wahl selbst und von der Zusammensetzung und dem technischen Hintergrund der Wahlberechtigten ab. Die Akzeptanz und Nutzung elektronischer Wahlen wird zudem stark vom Vertrauen bestimmt, das die Wahlberechtigten den Sicherheitseigenschaften des Systems entgegenbringen. Das Wahlverfahren und das Ergebnis der Stimmenauszählung müssen transparent sein, damit der Wähler die Rechtmäßigkeit einer Wahl anerkennt. Nur dann sind Entscheidungen legitimiert. Es gilt also mit Hilfe der Kryptografie elektronische Wahlsysteme zu konstruieren, die die Anforderungen erfüllen, die an demokratische Wahlen zu stellen sind (vgl. Kapitel 4).

1.2 Stand der Forschung

In den letzten Jahren sind viele elektronische Wahlsysteme veröffentlicht worden. Die Anforderungen an solche Wahlprotokolle wurden dabei seit den Anfängen von David Chaum in [Cha81] immer weiter verfeinert und entsprechende Wahlsysteme konstruiert. Das Sicherheitsziel, das in den letzten Jahren die Forschung im Bereich elektronischer Wahlprotokolle am stärksten geprägt hat, ist die Unüberprüfbarkeit. Ein Wähler soll in einem Wahlsystem nicht die Möglichkeit haben, einem Angreifer oder Erpresser gegenüber *beweisen* zu können, wie er gewählt hat. Wäre er dazu in der Lage, könnte der Angreifer einen solchen Beleg fordern und der Wähler wird erpressbar. Der Begriff der Unüberprüfbarkeit wurde zuerst von Benaloh und Tuinstra in [BT94] eingeführt. Die seither bekanntesten Wahlsysteme sowie ihre Stärken und Schwächen, werden in diesem Abschnitt kurz vorgestellt und analysiert.

1.2.1 Observer

Das Konzept eines Observers wurde erstmals von Chaum und Pedersen in [CP92] eingeführt und von Cramer und Pedersen in [CP93] weiter verfeinert. Ein Observer ist eine manipulationssichere Hardware, die sich im Besitz des Wählers befindet. Der Observer darf nicht direkt mit der Außenwelt kommunizieren. Die komplette Kommunikation muss über den Wähler ablaufen.

1.2.2 Wahlverfahren ohne Observereinsatz

Bislang ist über die Einsatzmöglichkeiten eines Observers in elektronischen Wahlsystemen wenig bekannt. Fast alle bisher bekannten Wahlverfahren setzen keinen Observer ein.

Man kann Wahlverfahren weiterhin klassifizieren in Verfahren mit homomorpher Verschlüsselung und solche, die auf MIX-Netzen basieren.

Wahlverfahren mit homomorpher Verschlüsselung

Bei elektronischen Wahlsystemen, die homomorphe Verschlüsselungsverfahren einsetzen, wird die Geheimhaltung der Stimme unter Ausnutzung dieser Homomorphie erreicht. Die abgegebenen Geheimtextstimmen werden zu einem verschlüsselten Auszählungsergebnis verrechnet, das zum Schluss entschlüsselt wird. In der Auszählung treten also keine einzelnen Klartextstimmen auf.

Benaloh und Tuinstra führten in [BT94] die Sicherheitsanforderung der Unüberprüfbarkeit ein und stellten ein Wahlsystem vor, das diese Eigenschaft aufweisen sollte. Hirt und Sako zeigten aber in [HS00], dass der Wähler im Wahlsystem von Benaloh und Tuinstra doch einen Beleg erstellen kann, wie er gewählt hat. Der Wähler muss in [BT94] mit einem cut-and-choose-Beweis zeigen, dass seine Stimme gültig ist. Wenn sich der Wähler vor der Wahl auf die innere Anordnung jedes Stimmenpaares in diesem cut-and-choose-Beweis festlegt und die Anordnung dem Angreifer offenbart, so kann der Angreifer in jeder Runde des Beweises überprüfen, ob die offengelegten Stimmen mit der Festlegung übereinstimmen. Der Wähler hat demzufolge nur noch eine Wahrscheinlichkeit von 2^{-N} im Sicherheitsparameter N, den Beleg zu fälschen und den Angreifer über die verlangte Stimmabgabe zu täuschen.

Okamoto stellte in [Oka96] ein Wahlverfahren vor, dessen postulierte Unüberprüfbarkeit von ihm selbst in [Oka97] widerlegt wurde. Der Fehler in der Unüberprüfbarkeit von [Oka96] ist ähnlich dem in [BT94]. Im System wird ein Trapdoor-Bit-Commitment vom Wähler verlangt. Wird das Commitment allerdings durch einen Angreifer festgelegt und der Wähler gezwungen dieses zu verwenden, so kann der Wähler nicht mehr über den Inhalt der gewählten verschlüsselten Stimme hinwegtäuschen.

Sako und Kilian stellten in [SK95] ein MIX-Netz-basiertes Wahlsystem mit homomorpher Auszählung vor, das dem System von Hirt und Sako [HS00] als Vorlage diente. Letzteres hat, wenn man von dem MIX-Netz-basierten Verfahren von Juels, Catalano und Jakobsson [JCJ05] absieht, die Unüberprüfbarkeit am besten realisiert. In diesem Schema (siehe Abbildung 1.1) verschlüsselt und permutiert ein MIX-Netz von Autoritäten jede der Wahloptionen und beweist dem Wähler in Designated-Verifier-Beweisen die verwendete Permutation.[1] Dieser Beweis muss

[1] In Abbildung 1.1 ist die Wahlphase von [HS00] schematisch für drei Autoritäten A_1, A_2, A_3 und drei Wahl-

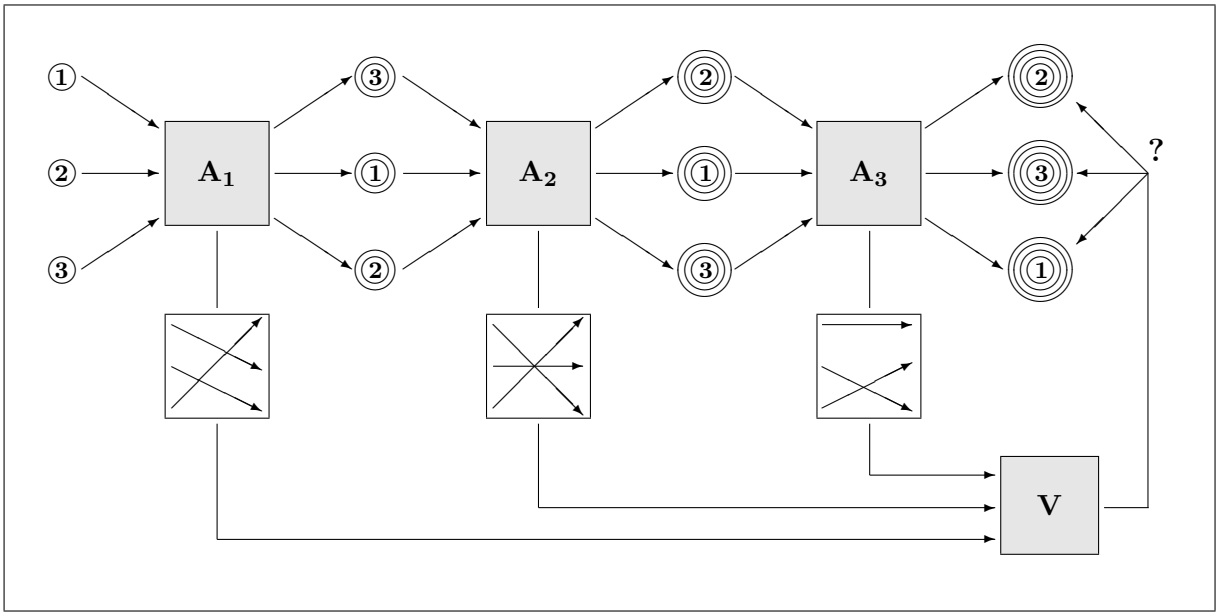

Abbildung 1.1: Schematische Darstellung der Wahlphase im Wahlsystem [HS00].

jedoch über einen physikalisch sicheren Kanal an den Wähler übertragen werden. Ein solcher Kanal ist durch Verschlüsselung alleine nicht zu erreichen, da der Wähler das zur Entschlüsselung verwendete Geheimnis kennt und so den Designated-Verifier-Beweis einem Angreifer gegenüber offenlegen kann. Die Verarbeitung von jeder Wahloption für jeden Wähler in der Wahlphase ist nicht effizient und die Annahme eines physikalisch sicheren Kanals von jeder Wahlautorität zu jedem Wähler ist unrealistisch. Daher wird in Kapitel 5 ein neues unüberprüfbares effizientes Wahlsystem vorgestellt, das ich im Februar 2006 auf der 3. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik [Sch06a] vorgestellt und veröffentlicht habe.

Wahlverfahren mit MIX-Netzen

Bereits das erste kryptografische Wahlverfahren [Cha81] basiert auf der Verwendung eines MIX-Netzes zur Anonymisierung der Stimmen. Wahlverfahren mit MIX-Netzen erreichen die Anonymität durch Permutation und gleichzeitige Ver- oder Entschlüsselung der abgegebenen einzelnen Geheimtextstimmen. Durch die Permutation wird der Zusammenhang zwischen ein- und ausgehenden Nachrichten verschleiert (siehe [BSW06]). Die Klartextstimmen können nicht mehr einzelnen Wählern zugeordnet werden. Daher werden sie veröffentlicht, und jeder ist nun in der Lage, die Stimmen auszuzählen. In [Oka97] stellte Okamoto ein weiteres Wahlverfahren vor, das tatsächlich Unüberprüfbarkeit bietet, allerdings einen abhörsicheren Kanal von jedem Wähler zum Wahlausschuss voraussetzt.

Im November 2005 haben Juels, Catalano und Jakobsson in [JCJ05] ein Wahlverfahren vorgestellt, das neben der Unüberprüfbarkeit eine weitere Sicherheitsanforderung, die sogenannte Erpressungsresistenz, erfüllt. In einem erpressungsresistenten Wahlsystem kann ein Angreifer

optionen $1, 2, 3$ und einen Wähler V dargestellt. Die Verschlüsselungen werden durch Kreise um die Wahloptionen symbolisiert.

nicht anstelle des Wählers abstimmen und den Wähler auch nicht zwingen, zufällig zu wählen (wie es beispielsweise in [HS00] möglich ist) oder der Wahl fernzubleiben.

Die Stimmabgabe und Auszählung des Wahlsystems von Juels et al. ist in Abbildung 1.2 schematisch dargestellt. Im Unterschied zu anderen Wahlsystemen werden die vom Wähler abge-

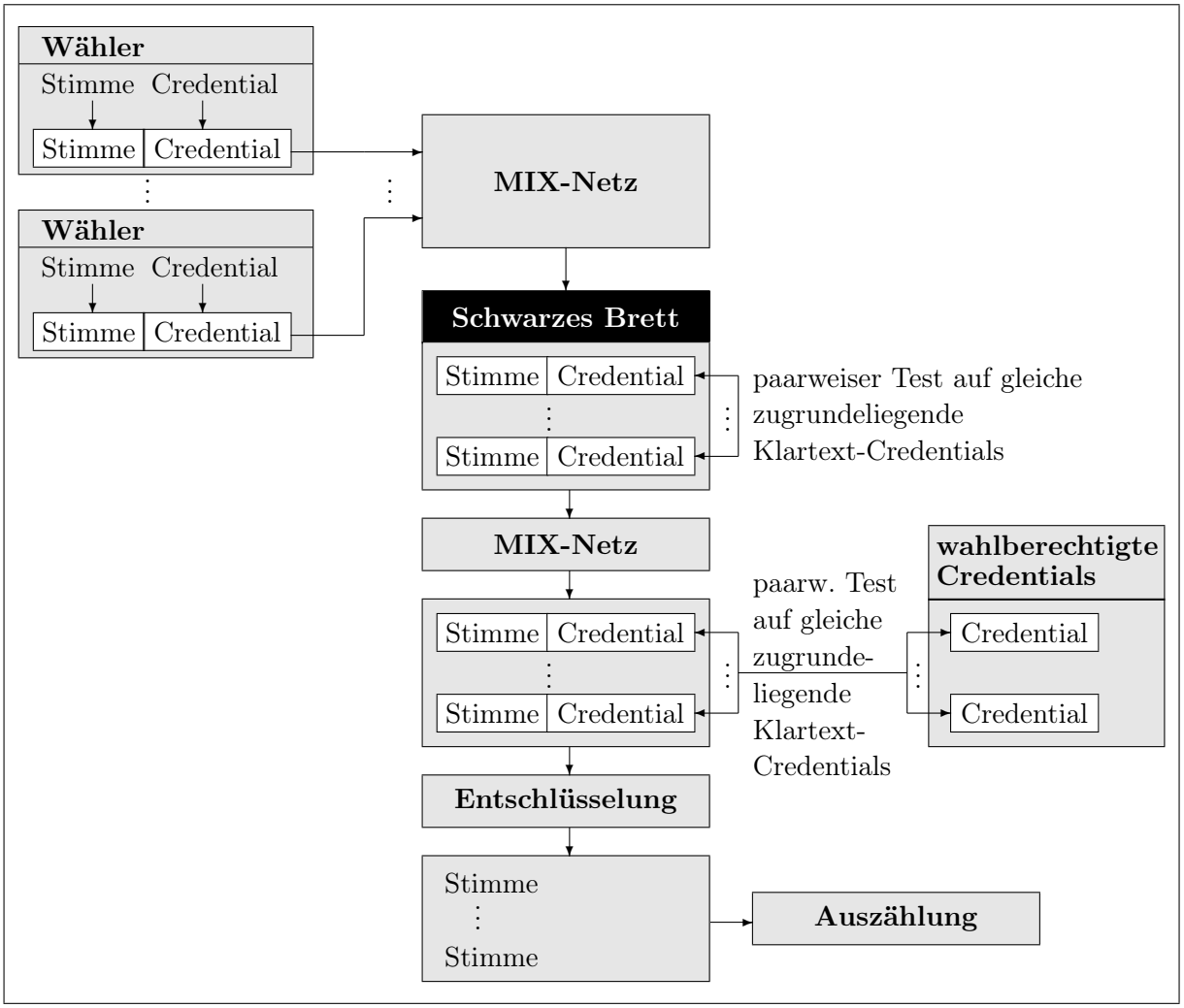

Abbildung 1.2: Darstellung der Wahlphase und Auszählung im Wahlsystem [JCJ05].

gebenen Stimmen in [JCJ05] nicht digital signiert, sondern zusammen mit einer verschlüsselten Wahlberechtigung (Credential) abgeschickt. In Abbildung 1.2 sind Geheimtexte von Credentials oder Stimmen durch Kasten um Credential bzw. Stimme dargestellt.

Während der Auszählung werden die Credential zunächst paarweise auf Gleichheit überprüft, ohne sie zu entschlüsseln. Dann werden Stimmen und Credentials gemeinsam der gleichen Permutation unterworfen. Anschließend werden die verschlüsselten Credentials paarweise mit den verschlüsselten Credentials auf der Liste der Wahlberechtigungen verglichen. Schließlich werden die Stimmen entschlüsselt und ausgezählt.

Der Nachteil des Systems [JCJ05] liegt in der Effizienz. Alle abgegeben Stimmen müssen

zweimal während der Auszählung *paarweise* auf Gleichheit der zugrundeliegenden Klartexte untersucht werden, ohne diese Klartexte explizit zu bestimmen.

Daher wird in Kapitel 6 ein neues, deutlich effizienteres erpressungsresistentes System vorgestellt. Dieses Wahlsystem mit Observer habe ich im August 2006 auf dem 2nd International Workshop on Electronic Voting, Electronic Voting 2006, in [Sch06b] veröffentlicht.

1.2.3 Wahlverfahren mit Observereinsatz

Magkos, Burmester und Chrissikopoulos stellten 2001 ein Wahlverfahren vor, das eine manipulationssichere Hardware, einen Observer, einsetzt. Sie können so auf die Anforderung eines abhörsicheren Kanals (vgl. [Oka97]) verzichten. Das Wahlverfahren ist allerdings entgegen der Meinung der Autoren nicht unüberprüfbar, wie ich im September 2005 auf dem Kryptotag der Gesellschaft für Informatik in [Sch05a] gezeigt habe. In Kapitel 5 wird das System von Magkos et al. beschrieben und gezeigt, warum das Wahlsystem nicht unüberprüfbar ist und eine Lösung des Problems aufgezeigt.

1.3 Ziele, Aufbau und Ergebnisse der Arbeit

Bis November 2005 gab es kein elektronisches Wahlsystem, das die Eigenschaft der Unüberprüfbarkeit (siehe Abschnitt 4.1.7) erfüllt, ohne dabei unrealistische Anforderungen, wie einen physikalisch sicheren Kanal von jeder Autorität im Wahlausschuss zu jedem Wähler, aufzustellen. In dieser Arbeit soll daher ein elektronisches Wahlsystem entwickelt und vorgestellt werden, das diese Eigenschaft aufweist. Es soll untersucht werden, ob sich ein Observer zur Erreichung der Unüberprüfbarkeit eignet.

In der Arbeit von Juels, Catalano und Jakobsson wird mit der sogenannten Erpressungsresistenz eine neue Anforderung definiert, die über die Unüberprüfbarkeit hinausgeht. Das in [JCJ05] vorgestellte Wahlsystem erreicht diese Erpressungsresistenz ohne den Einsatz eines Observers. Ein weiteres Ziel dieser Arbeit ist es daher, zu untersuchen, ob sich ein Observer vorteilhaft bei der Konstruktion erpressungsresistenter elektronischer Wahlsysteme einsetzen lässt.

Die Grundlagen für die in den Wahlprotokollen eingesetzten kryptografischen Bausteine werden in Kapitel 2 beschrieben. Die für die Konstruktion der elektronischen Wahlprotokolle aufgestellten kryptografischen Protokolle wie z. B. Witness-Indistinguishable-Beweise für zwei Prover oder spezielle Zero-Knowledge-Beweise werden in Kapitel 3 vorgestellt und analysiert. Die Anforderungen, die an elektronische Wahlen zu stellen sind, insbesondere die Erpressungsresistenz werden in Kapitel 4 definiert. In Kapitel 5 wird das bestehende Wahlsystem mit Observer [MBC01] erläutert, die von mir entdeckten Sicherheitslücken in diesem System aufgezeigt und geschlossen. Anschließend wird in Abschnitt 5.3 ein neues Wahlsystem[2] mit Observer vorgestellt, das die Vorzüge von [HS00] und [MBC01] kombiniert und den Aufwand insbesondere auf Seiten des Wählers reduziert. In Kapitel 6 wird ein neues observerbasiertes, erpressungsresistentes Wahlsystem[3] beschrieben und analysiert. Schließlich werden die präsentierten Wahlsyteme in

[2] Dieses Wahlsystem habe ich in [Sch06a] veröffentlicht.
[3] Das Wahlystem wurde von mir in [Sch06b] veröffentlicht.

Kapitel 7 vergleichend analysiert.

Mit den im Kapitel 5 entworfenen und analysierten Wahlsystemen wurde gezeigt, dass es möglich ist, mit Hilfe eines Observers effiziente elektronische Wahlsysteme aufzustellen, die Unüberprüfbarkeit bieten.

Das in Kapitel 6 vorgestellte erpressungsresistente Wahlsystem mit Observer erfüllt alle bisher geforderten Sicherheitsziele einschließlich der Erpressungsresistenz. Es zeigt sich, dass ein Observer in erpressungsresistenten Wahlsystemen nicht mehr im ursprünglichen Sinn von Chaum und Pedersen bzw. Cramer und Pedersen (vgl. [CP92], [CP93]) obligatorisch an den Berechnungen zur Stimmabgabe beteiligt werden kann. Der Observer dient in diesem System im Wesentlichen als sicheres Transportmedium der Wahlberechtigung.

Ich danke Prof. Dr. Albrecht Beutelspacher für die Gelegenheit, diese Arbeit zu verfassen und für die hervorragende Betreuung. Mein Dank gilt ebenso Herrn Prof. Dr. Jörg Schwenk für die Bereitschaft, die Arbeit zu begutachten. Weiterhin bedanke ich mich bei meinen Kolleginnen und Kollegen Dr. Matthias Baumgart, Björn Fay, Dr. Christine Fremdt, Dr. Heike Neumann, Thomas Schwarzpaul und Dr. Christian Tobias für die wertvollen Gespräche und Diskussionen. Ferner danke ich meinen Kolleginnen und Kollegen im Beirat zur Erstellung des Schutzprofils für Grundanforderungen an elektronische Wahlen für die interessanten Diskussionen. Ich danke Mareike Eisel, meiner Familie und meinen Freunden für ihre Unterstützung und Aufmunterung.

Wenn der Stimmzettel
gesprochen hat, so hat die
höchste Instanz gesprochen.

Victor Hugo

Kapitel 2

Grundlagen

Protokolle für elektronische Wahlen bestehen aus einzelnen kryptografischen Verfahren und Protokollen. In diesem Kapitel werden die kryptografischen Basismechanismen und Grundlagen vorgestellt, die in den Kapiteln 3, 5 und 6 verwendet werden.

2.1 Definitionen

2.1.1 Verschlüsselung, Verschlüsselungsschema

Es sei M die Menge der Klartexte, C die Menge der Geheimtexte und K die Menge der Schlüssel. Als *Verschlüsselung* (*Verschlüsselungsfunktion*) bezeichnet man eine injektive Abbildung $E : M \to C$. Ein *Verschlüsselungsschema* (*Verschlüsselungsalgorithmus*) ist eine Menge injektiver Funktionen $E_k : M \to C$ mit $k \in K$.

2.1.2 Hashfunktion

Hashfunktionen sind Funktionen, die beliebig große Eingabewerte auf einen festen Wertebereich abbilden. In der Kryptografie werden meist Hashfunktionen gefordert, die sowohl die Einweg-Eigenschaft, als auch die starke Kollisionsresistenz aufweisen. Bei einer Hashfunktion f mit Einwegeigenschaft darf es praktisch nicht möglich sein, zu einem zufällig gewählten Bild y ein passendes Urbild $f^{-1}(y)$ zu finden. Die starke Kollisionsresistenz verlangt, dass es nur schwer möglich sein darf, zwei verschiedene Eingabewerte x und x' mit dem gleichen Bild (also $f(x) = f(x')$) zu finden. Kryptografische Hashfunktionen bilden idealerweise auf jeden Wert im Bildbereich der Funktion etwa gleich oft ab. Weitere Informationen zu Hashfunktionen sind im Buch von Beutelspacher, Schwenk und Wolfenstetter [BSW06] oder in [MOV97] nachzulesen.

2.2 Asymmetrische Kryptografie

Im Jahr 1976 gaben Diffie und Hellman in [DH76] den Anstoß zur Entwicklung eines neuen Konzeptes, der sogenannten *asymmetrischen Kryptografie* oder *Public-Key-Kryptografie*. Jeder Teilnehmer besitzt ein Schlüsselpaar, bestehend aus einem geheimen Schlüssel und einem öffentlichen Schlüssel. Der öffentliche Schlüssel wird für jeden Teilnehmer zugänglich in einem

Verzeichnis eingetragen.

Möchte ein Teilnehmer eine Nachricht versenden, verschlüsselt er diese unter dem öffentlichen Schlüssel des Kommunikationspartners. Dieser kann den Chiffretext unter seinem zugehörigen privaten Schlüssel entschlüsseln. Zwar sind die beiden Schlüssel voneinander abhängig, aber es ist praktisch unmöglich den geheimen Schlüssel aus dem öffentlichen Schlüssel zu bestimmen. Dieses Merkmal nennt man *Public-Key-Eigenschaft*. Ein Beispiel einer solchen Verschlüsselung ist die ElGamal-Verschlüsselung, die im Abschnitt 2.2.4 erläutert wird.

Der Vorteil der Public-Key-Kryptografie ist offensichtlich. Jeder Teilnehmer besitzt lediglich sein eigenes Schlüsselpaar und muss nicht mehr mit jedem Kommunikationspartner einen gemeinsamen Schlüssel vereinbaren. Auch hinzukommende Benutzer lassen sich ohne großen Aufwand integrieren.

Mit Public-Key-Verfahren kann man nicht nur verschlüsseln, sondern auch Nachrichten auf Integrität und Authentizität prüfen sowie Autorisation beim Zugang zu Systemen regeln.

2.2.1 Public-Key-Verschlüsselung

Eine *Public-Key-Verschlüsselung* ist ein Tripel (G, E, D) probabilistischer, polynomieller Algorithmen, das die folgenden Eigenschaften aufweist. Bei der Schlüsselgenerierung G wird unter Eingabe eines Sicherheitsparameters k ein Schlüsselpaar (PK, SK) generiert. Man nennt PK den öffentlichen und SK den geheimen bzw. privaten Schlüssel. Die Verschlüsselungsfunktion E berechnet bei Eingabe des öffentlichen Schlüssels und einer Nachricht $m \in \{0,1\}^k$ einen Geheimtext $c = E_{PK}(m) \in \{0,1\}^{k'}$. Die Entschlüsselung D erhält als Eingabe den Geheimtext c und den privaten Schlüssel und berechnet $m = D_{SK}(E_{PK}(m))$.

Semantische Sicherheit

Eine Public-Key-Verschlüsselung (G, E, D) nennt man *semantisch sicher*, wenn ein effizienter Algorithmus X' und eine vernachlässigbare[1] Funktion ν für alle zufällig gewählten Schlüsselpaare (PK, SK), alle Angreifer X und alle Funktionen h existiert, so dass gilt:

$$P[X(k, c, PK) = h(m)] \leq P[X'(k, PK) = h(m)] + \nu(k).$$

Bei einem semantisch sicheren Verschlüsselungsverfahren darf ein Angreifer X, der den Geheimtext kennt, nur eine vernachlässigbar bessere Wahrscheinlichkeit haben, über den dazugehörenden Klartext Angaben machen zu können, als es ein Angreifer X' kann, der den Geheimtext nicht kennt.

Polynomielle Ununterscheidbarkeit

Man nennt eine Public-Key-Verschlüsselung (G, E, D) *polynomiell ununterscheidbar*, wenn es für jeden zufällig gewählten Schlüssel PK zum Sicherheitsparameter k und für jeden effizienten

[1] Eine Größe $\nu(k)$ heißt *vernachlässigbar* in k, wenn für jede natürliche Zahl c ein ℓ_c mit $\nu(k) < k^{-c}$ für $k > \ell_c$ existiert.

Algorithmus X eine vernachlässigbare Funktion ν gibt, so dass für zwei Nachrichten m_0, m_1 gilt:

$$P[X(k, PK, m_0, m_1, c) = m | c = E(PK, m), m \in_R \{m_0, m_1\}] \leq \frac{1}{2} + \nu(k).$$

Da $\nu(k)$ mit wachsendem Sicherheitsparameter k exponentiell gegen 0 konvergiert, liegt die Erfolgswahrscheinlichkeit des Angreifers X für ausreichend große Schlüssellängen vernachlässigbar über der Ratewahrscheinlichkeit.

Satz 2.1 (Semantischer Sicherheit und Polynomielle Ununterscheidbarkeit)
Eine Public-Key-Verschlüsselung ist genau dann semantisch sicher, wenn sie polynomiell ununterscheidbar ist.

Beweis:
Der Beweis dieses Satzes findet sich beispielsweise in [MRS86] oder [GM84]. □

2.2.2 Angriffstypen

Wenn man die semantische Sicherheit eines Verschlüsselungsverfahrens untersucht, kann man zwischen verschiedenen Angriffstypen unterscheiden.

- In einem *Angriff mit bekanntem Geheimtext* (ciphertext only attack) kennt der Angreifer nur einen Teil der Geheimtexte.
- Eine bessere Situation für den Angreifer ist ein *Angriff mit bekanntem Klartext* (known plaintext attack). Bei diesem Angriff kennt er einen Teil des Geheimtextes und ein dazu passendes Stück Klartext.
- Noch günstiger ist es für einen Angreifer, wenn er sich Klartexte seiner Wahl verschlüsseln lassen kann. Man spricht dabei von einem *Angriff mit gewähltem Klartext* (chosen plaintext attack).
- Der stärkste Angriff ist der *Angriff mit gewähltem Geheimtext* (chosen ciphertext attack), bei der sich der Angreifer Geheimtexte seiner Wahl entschlüsseln lassen kann.

Bei den letzten beiden Angriffen unterscheidet man, ob der Angreifer die Klar- bzw. Geheimtexte vorher wählen muss (*direkter* Angriff) oder ob der Angreifer abhängig von den bisherigen Ergebnissen die weiteren Texte wählen kann (*adaptiver* Angriff).

Ein Kryptosystem gewährleistet die höchste Sicherheit, wenn ein adaptiver Angriff mit gewähltem Geheimtext dem Angreifer keine Informationen über den Klartext oder den verwendeten Schlüssel liefert.

2.2.3 Public-Key-Infrastruktur (PKI)

Trotz der Vereinfachung des Schlüsselmanagements gegenüber symmetrischen Verfahren sind dennoch ausgearbeitete Verfahren zur Erzeugung, Zuordnung, Ausgabe, Beglaubigung, Verteilung und Sperrung von Schlüsseln erforderlich.

Zur Gewährleistung der Integrität und Vertrauenswürdigkeit benötigt man ein komplexes Managementsystem, eine sogenannte *Public-Key-Infrastruktur (PKI)*. Eine PKI besitzt eine Zertifizierungsstelle, die einen öffentlichen Schlüssel eines Teilnehmers mit seinem Namen in einem *Zertifikat* verbindet, nachdem sich dieser gegenüber dieser Zertifizierungsstelle identifiziert hat (siehe z. B. [Sch05b] oder [BNS05]). Nur so kann sichergestellt werden, dass der öffentliche Schlüssel auch wirklich zum entsprechenden Teilnehmer gehört.

Weitere Aufgaben einer PKI sind z. B. die sogenannte *Rückrufliste*, in der kompromittierte Schlüssel gespeichert werden, aber auch Zeitstempeldienste, die Daten durch digitale Signaturen verifizierbar mit einem Zeitpunkt verbinden.

Für manche bei elektronischen Wahlen eingesetzten kryptografischen Beweise (vgl. Abschnitt 3.5) ist es wichtig, dass ein Teilnehmer über seinen eigenen privaten Schlüssel verfügt. Auch diese Sicherstellung der Kenntnis des eigenen privaten Schlüssels kann Aufgabe einer PKI sein.

2.2.4 Diskreter Logarithmus

Taher ElGamal entwickelte 1984 ein asymmetrisches Verschlüsselungsverfahren [Elg85] basierend auf dem Diskreten Logarithmus.

Diskreter-Logarithmus-Annahme

Gegeben sei eine Primzahl p von k Bit Länge, ein Generator g der Gruppe $\mathbb{Z}_p^*$ und ein Wert $a \in_R \mathbb{Z}$. Nun kann man zwar den Wert g^a effizient berechnen, es ist aber bisher kein effizienter Algorithmus X bekannt, der den Wert a der *diskreten Logarithmusfunktion* (Umkehrfunktion) aus g^a bestimmt. Das heißt, für jedes X gibt es eine vernachlässigbare Funktion ν und ein $k_0 \in \mathbb{N}$, so dass für alle $k \geq k_0$ gilt:

$$P[X(k, p, g, g^a) = a | a \in_R \mathbb{Z}] \leq \nu(k).$$

Unter der Annahme, dass es keinen effizienten Algorithmus gibt, ist die diskrete Exponentialfunktion $\mathbb{N} \to G$, $a \to g^a$ also eine Einwegfunktion.

ElGamal-Verschlüsselung

Bei der ElGamal-Verschlüsselung betrachtet man eine endliche Untergruppe G von $\mathbb{Z}_p^*$ mit Primzahlordnung $|G| = q$, wobei p eine Primzahl ist und $q|p-1$. Es sei g ein erzeugendes Element dieser Untergruppe, d. h. $G =< g >$, und es sei $m \in G$ eine zu verschlüsselnde Nachricht. Im Folgenden sind alle Multiplikationen Gruppenoperationen innerhalb von G. Bei einer ElGamal-Verschlüsselung besitzt der Empfänger ein Schlüsselpaar, bestehend aus dem geheimen Schlüssel s, der gemäß einer Gleichverteilung aus $\mathbb{Z}_q$ ausgewählt wird und dem öffentlichen Schlüssel $h = g^s$. Der öffentliche Schlüssel h ist dem Sender der Nachricht bekannt, so dass dieser seine Nachricht m verschlüsselt, indem er einen zufälligen Wert $\alpha \in \mathbb{Z}_q$ wählt und dem Empfänger das Paar $(x, y) = (g^\alpha, h^\alpha m)$ als Chiffretext übermittelt. Dieser kann das Chiffrat entschlüsseln, indem er mit seinem geheimen Schlüssel s den Wert $m = y \cdot x^{-s}$ berechnet.

Die ElGamal-Verschlüsselung wird oft allgemeiner dargestellt, indem der Geheimtext über eine Chiffre f unter einem symmetrischen Schlüssel h^α erhalten wird. Da bei elektronischen Wahlen zur Auszählung eine homomorphe Funktion (siehe Abschnitt 2.2.5) wünschenswert ist, wird in dieser Arbeit nur die Variante mit $f(m) = h^\alpha m$ verwendet.

Ein Angreifer kann den geheimen Schlüssel s aus $h = g^s$ unter der Diskreter-Logarithmus-Annahme nicht berechnen. Man weiß allerdings nicht, ob es eine Möglichkeit gibt, aus g^α und $h = g^s$ den zur Entschlüsselung notwendigen Wert $g^{\alpha s}$ zu berechnen. Die Vermutung, dass auch das nicht möglich ist, wird in der Diffie-Hellman-Berechnungsannahme formuliert.

Diffie-Hellman-Berechnungsannahme

Gegeben seien eine Primzahl p der Länge k Bit und eine Untergruppe G von $\mathbb{Z}_p^*$ mit Primzahlordnung $|G| = q$. Des Weiteren sei g ein erzeugendes Element von G sowie a und b zwei zufällig gewählte Elemente aus $\mathbb{Z}_q$. Dann ist es für hinreichend große Primzahlen p und q praktisch unmöglich, ohne Kenntnis von a oder b aus g^a und g^b den Wert g^{ab} zu berechnen. Formal bedeutet das, dass es für jeden effizienten Algorithmus X eine vernachlässigbare Funktion ν und ein $k_0 \in \mathbb{N}$ gibt, so dass für alle $k \geq k_0$ gilt:

$$P[X(k, p, q, g, g^a, g^b) = g^{ab} | a, b \in_R \mathbb{Z}] \leq \nu(k).$$

Falls diese Annahme nicht zuträfe, wäre es möglich, g^{ab} aus g^a und g^b zu berechnen. Bezüglich der Verschlüsselung mit ElGamal bedeutete dies dann, dass eine Nachricht $(x, y) := (g^\alpha, h^\alpha m)$ mit $h := g^s$ leicht zu entschlüsseln wäre. Man könnte dann $g^{\alpha s} = h^\alpha$ aus g^α und g^s ohne Kenntnis von s berechnen und die Nachricht $m = y \cdot h^{-\alpha}$ entschlüsseln.

Kann ein Angreifer diskrete Logarithmen berechnen, so kann er auch die Diffie-Hellman-Berechnungsannahme brechen. Somit ist die Diffie-Hellman-Berechnungsannahme schwächer als die Diskrete-Logarithmus-Annahme.

Diffie-Hellman-Entscheidungsannahme

Man vermutet sogar, dass ein Angreifer nicht einmal merken würde, dass er den korrekten Wert g^{ab} aus g^a und g^b errechnet hat. Dies ist die sogenannte *Diffie-Hellman-Entscheidungsannahme*.

Gegeben seien eine Primzahl p der Länge k Bit, eine zyklische Untergruppe G von $\mathbb{Z}_p^*$ mit Primzahlordnung q, ein erzeugendes Element g von G sowie drei Elemente g^a, g^b und g^c aus G. Dann gibt es keinen effizienten Algorithmus X, der entscheiden kann, ob $g^{ab} \equiv g^c \pmod p$ gilt. Das heißt, für jedes X gibt es eine vernachlässigbare Funktion ν und ein $k_0 \in \mathbb{N}$, so dass für alle $k \geq k_0$ gilt:

$$P[X(k, p, g, g^a, g^b, g^c) = 1 | a, b \in_R \mathbb{Z}, g^c = g^{ab}] \leq \frac{1}{2} + \nu(k).$$

2.2.5 Homomorphe Verschlüsselung

Bei einer homomorphen Verschlüsselungsfunktion können die Chiffretexte miteinander so verknüpft werden, dass die sich daraus ergebende Entschlüsselung einer Verknüpfung der Klartexte entspricht.

Bei elektronischen Wahlverfahren werden die Stimmen zur Geheimhaltung einzelner Wahlentscheidungen verschlüsselt. Homomorphe Verschlüsselungsfunktionen sind für elektronische Wahlen besonders gut geeignet, da man das Auszählungsergebnis erhalten kann, ohne einzelne Stimmen zu entschlüsseln. Die verschlüsselten Stimmen werden addiert und nur die Summe der Stimmen wird entschlüsselt. Dadurch wird die Anonymität einzelner Wähler gewährleistet.

Definition 2.1 (Homomorphe Verschlüsselung)
Sei M die Menge aller Klartexte und C die Menge der Chiffretexte, so dass M eine Gruppe bezüglich der Operation $\oplus$ und C eine Gruppe bezüglich der Operation $\otimes$ ist. Man nennt ein Verschlüsselungsschema $(\oplus, \otimes)$-*homomorph*, wenn für alle Verschlüsselungsfunktionen E, Schlüssel k, Klartexte $m_1, m_2 \in M$ und für zwei Geheimtexte $e_1 \in \{E_{k,r}(m_1) \,|r \text{ Zufallszahl}\}$ und $e_2 \in \{E_{k,r'}(m_2) \,|r' \text{ Zufallszahl}\}$ die *Verknüpfung* $e_1 \otimes e_2$ dieser Chiffretexte wieder ein Chiffretext e ist, der eine Verschlüsselung der Verknüpfung der einzelnen Klartexte darstellt, d. h. es gilt $e \in \{E_{k,\tilde{r}}(m_1 \oplus m_2) \,|\tilde{r} \text{ Zufallszahl}\}$.

Beispiel 2.1 (Homomorphe ElGamal-Verschlüsselung)
Der Raum der Klartexte ist bei der ElGamal-Verschlüsselung eine Gruppe $(G, \cdot)$ mit Primzahl-Gruppenordnung $|G| = q$. Der Raum der Chiffretexte $G \times G$ ist ebenfalls eine Gruppe bezüglich komponentenweiser Multiplikation.
Gegeben seien zwei Klartexte m_1 und m_2 und die zugehörigen Chiffretexte

$$e_1 := (g^{\alpha_1}, h^{\alpha_1} m_1) \text{ und } e_2 := (g^{\alpha_2}, h^{\alpha_2} m_2)\,.$$

Dann ist die komponentenweise Multiplikation der Chiffretexte

$$e_1 e_2 = (g^{\alpha_1} g^{\alpha_2}, h^{\alpha_1} h^{\alpha_2} m_1 m_2)$$

eine Verschlüsselung der multiplizierten Klartexte $m_1 \cdot m_2$.

2.2.6 Homomorphie bezüglich Addition bei ElGamal

Das originale ElGamal-System ist homomorph bezüglich Multiplikation. Bei elektronischen Wahlen möchte man die Homomorphie ausnutzen, um das Auszählungsergebnis als Summe der verschlüsselten Stimmen zu erhalten. Eine Möglichkeit, Homomorphie bezüglich Addition zu erreichen, ist die Darstellung der möglichen Nachrichten durch erzeugende Gruppenelemente g_i von G. Dem zweifachen Senden einer Nachricht mit primitivem Element g_i, also einer Multiplikation zweier Chiffretexte $(g^{\alpha_1}, h^{\alpha_1} g_i)$ und $(g^{\alpha_2}, h^{\alpha_2} g_i)$, entspricht die Addition im Exponenten $\left(g_i g_i = g_i^{1+1}\right)$. Die Multiplikation der Chiffretexte entspricht der Addition der auf die Nachricht entfallenen Anzahlen.

Die Verschlüsselung einer Nachricht v_i mit einem zufälligen Wert α_i ist dann das Wertepaar $(x_i, y_i) = (g^{\alpha_i}, h^{\alpha_i} v_i)$. Bei n verschlüsselten Nachrichten ergibt sich die komponentenweise Multiplikation zu

$$\left(\prod_{i=1}^{n} x_i, \prod_{i=1}^{n} y_i\right) = \left(g^{\alpha}, h^{\alpha} \cdot \prod_{i=1}^{n} v_i\right) \text{ mit } \alpha := \sum_{i=1}^{n} \alpha_i.$$

Betrachtet man K verschiedene Nachrichten, so kann man diese auch durch die Menge $M = \{1, n, n^2, \ldots, n^{K-1}\}$ darstellen. Dabei ist $n > 1$ die Anzahl aller Sender von Nachrichten. Anschaulich werden die Nachrichten als Stellenwerte in einem Zahlensystem zur Basis n dargestellt. Da die Anzahl der Nachrichten pro Nachrichtentyp höchstens n ist, kommt es nur dann zu einer Stellenüberschreitung, wenn alle n Sender die gleiche Nachricht schicken. Andernfalls beschreiben die Ziffern in diesem Stellensystem die Anzahlen der entsprechenden Nachrichten.

Um hier Homomorphie bezüglich Addition zu erreichen, wählt man einen weiteren Erzeuger γ der Gruppe G. Die ursprüngliche Nachricht v_i wird als Nachricht γ^{v_i} dargestellt. Einer Multiplikation zweier Chiffretexte $(g^{\alpha_1}, h^{\alpha_1}\gamma^{v_1})$ und $(g^{\alpha_2}, h^{\alpha_2}\gamma^{v_2})$ mit den Nachrichten v_1 und v_2 entspricht dann die Addition der Nachrichten im Exponenten: $\gamma^{v_1}\gamma^{v_2} = \gamma^{v_1+v_2}$.

2.3 Non-malleable ElGamal

In Public-Key-Kryptosystemen möchte man verhindern, dass ein Angreifer aus einem Geheimtext einen ähnlichen Geheimtext erzeugen und versenden kann, so dass zwischen den Klartexten ein Zusammenhang besteht. Das erreicht man mit dem Einsatz von non-malleable Verschlüsselungen, die in verschiedenen Bereichen Anwendung finden.

Bei MIX-Netzen (siehe Abschnitt 3.8) ist es wichtig, dass unehrliche Benutzer keine Kopien oder korrelierte Chiffretexte von verschlüsselten Nachrichten ehrlicher Benutzer versenden können. Andernfalls könnten sie (mit einer gewissen Wahrscheinlichkeit) die Entschlüsselung der angegriffenen Nachricht bestimmen, indem sie Wiederholungen in der Ausgabeliste zählen oder Korrelationen bestimmen. Es ist daher notwendig, ein sogenanntes non-malleable Verschlüsselungsschema zu verwenden.

Bei demokratischen Wahlen möchte man, dass die Wähler ihre Stimmen unabhängig von der Stimmentscheidung der anderen Wähler abgeben. Daher sind beispielsweise Wahlwerbung im Wahllokal und Veröffentlichungen von Zwischenergebnissen oder Hochrechnungen vor Ablauf der Wahlphase verboten. Bei elektronischen Wahlen bedeutet dies, dass man darüber hinaus keine verschlüsselten Stimmen abgeben darf, deren Klartexte zu denen anderer Stimmen in einer dem Sender bekannten Relation stehen. Auch an dieser Stelle sind also Kryptosysteme erforderlich, die eine non-malleable Verschlüsselung aufweisen.

Definition 2.2 (non-malleable)
Gegeben sei ein in einem Public-Key-Verschlüsselungschema mit Sicherheitsparameter k erstellter Geheimtext $E(m)$ zu einem zufälligen Klartext m. Es sei R eine beliebige, effizient überprüfbare Relation. Man betrachte die folgenden zwei Szenarios.

1. Ein Angreifer kennt einen Geheimtext $E(m)$, aber nicht den dazugehörigen Klartext m. Er versucht, einen von $E(m)$ verschiedenen Geheimtext $E(m')$ zu erzeugen, so dass die Klartexte m und m' in der ihm bekannten Relation R stehen.

2. Ein Angreifer kennt $E(m)$ nicht und versucht einen zu $E(m)$ verschiedenen Geheimtext zu erzeugen, so dass die Klartexte m und m' in der ihm bekannten Relation R stehen.

Ein Kryptosystem ist *non-malleable* (*unverformbar*), wenn für jede Relation R eine vernachlässigbare Funktion ν existiert, so dass sich die Erfolgswahrscheinlichkeiten für den Angreifer in beiden oben beschriebenen Szenarios nur in $\nu(k)$ unterscheiden.

2.3.1 Grundidee der non-malleable ElGamal-Verschlüsselung

Der Grundgedanke einer non-malleable ElGamal-Verschlüsselung ist, dass der Sender S einen Zero-Knowledge-Beweis[2] liefert, dass er den zur Verschlüsselung verwendeten zufälligen Wert kennt. Dazu muss der Zero-Knowledge-Beweis selbst non-malleable sein, d. h. er muss den gewählten Namen des Senders oder eine andere eindeutig dem Sender zuzuordnende Information ID_S enthalten, damit die Spur zum Sender S zurückverfolgt werden kann (vgl. Abschnitt 3.2). Amit Sahai hat in [Sah99] gezeigt, dass das Konzept der non-malleability unter dieser Voraussetzung äquivalent zur semantischen Sicherheit unter einem adaptiven Angriff mit gewähltem Geheimtext ist.

Lemma 2.2
Das ElGamal-Verschlüsselungsschema ist unter der Diffie-Hellman-Entscheidungsannahme semantisch sicher gegen einen Angriff mit adaptiv gewähltem Klartext.

Beweis:
Der Beweis ist in [TY98] nachzulesen. □

Satz 2.3
Wenn der Sender einer ElGamal-Nachricht einen Zero-Knowledge-Beweis liefert, dass er die zur Verschlüsselung verwendete Zufallszahl kennt, ist dieses ElGamal-Verschlüsselungsschema non-malleable unter der Diffie-Hellman-Entscheidungsannahme.

Beweis:
Gegeben sei der öffentliche Schlüssel PK zum Sicherheitsparameter k. Man betrachte die zwei folgenden Szenarios:

- Ein Sender X kennt den Geheimtext $E(m)$ zu einem ihm unbekannten Klartext m.
  Mit einer Erfolgswahrscheinlichkeit p_1 gelingt es ihm, einen Geheimtext $E(m') \neq E(m)$ zusammen mit einem Beweis der Kenntnis der zur Verschlüsselung verwendeten Zufallszahl zu generieren, so dass m' und m in einer ihm bekannten, effizient überprüfbaren Relation R stehen.

  Da der Sender X in einem Zero-Knowledge-Beweis gezeigt hat, dass er die zur Verschlüsselung von $E(m')$ verwendete Zufallszahl kennt, kann er den ElGamal-Geheimtext $E(m')$ entschlüsseln und erhält m'. Über die ihm bekannte Relation R, in der m' und m stehen, kann er nun Informationen $h(m)$ über den Klartext m ermitteln.

[2] In einem Zero-Knowledge-Beweis erfährt der Verifier während des Beweises nur, dass der Prover einen Beweis für die Behauptung kennt. Zero-Knowledge-Beweise werden in Kapitel 3 behandelt.

Die Erfolgswahrscheinlichkeit $P[X(k, E(m), PK) = h(m)]$ für X, diese Informationen $h(m)$ zu ermitteln kann aber größer als p_1 sein, da es möglicherweise noch andere Wege gibt, um an die Informationen $h(m)$ zu gelangen. Es sei p'_1 die Wahrscheinlichkeit dafür, dass X auf anderen Wegen an $h(m)$ gelangt. Dann ist $P[X(k, E(m), PK) = h(m)] = p_1 + p'_1$ bzw. $p_1 = P[X(k, E(m), PK) = h(m)] - p'_1$.

- Der Sender X kennt weder m noch den Geheimtext $E(m)$. Er erzeugt mit einer Erfolgswahrscheinlichkeit p_2 einen Geheimtext $E(m') \neq E(m)$ zusammen mit einem Beweis der Kenntnis der zur Verschlüsselung verwendeten Zufallszahl, so dass m' und m in der ihm bekannten Relation R stehen.

  Da er die Zufallszahl kennt, die zur Verschlüsselung verwendet wurde, kann er wie im ersten Szenarium $E(m')$ zum Wert m' entschlüsseln. Über die ihm bekannte effizient überprüfbare Relation R, in der m' und m stehen, kann er nun die gleichen Informationen $h(m)$ über den Klartext m ermitteln wie im ersten Szenarium.

  Die Erfolgswahrscheinlichkeit $P[X(k, PK) = h(m)]$ für X, diese Informationen $h(m)$ nun ohne Kenntnis von $E(m)$ zu ermitteln, kann wiederum größer als p_2 sein, da es andere Wege geben kann, um an die Informationen $h(m)$ zu gelangen. Es sei p'_2 die Wahrscheinlichkeit dafür, dass X auf anderen Wegen an $h(m)$ gelangt. Dann ist $P[X(k, PK) = h(m)] = p_2 + p'_2$ bzw. $p_2 = P[X(k, PK) = h(m)] - p'_2$.

  Für die Wahrscheinlichkeit p'_2, über andere Wege $h(m)$ zu bestimmen, gilt $p'_2 \leq p'_1$, da X im zweiten Szenarium den Wert $E(m)$ nicht kennt.

Da das ElGamal-Verschlüsselungsverfahren unter der Diffie-Hellman-Entscheidungsannahme semantisch sicher gegen einen Angriff mit gewähltem Klartext ist, unterscheiden sich die Wahrscheinlichkeiten $P[X(k, E(m), PK) = h(m)]$ und $P[X(k, PK) = h(m)]$ nur in einer vernachlässigbaren Funktion $\nu(k)$. Die Differenz der Erfolgswahrscheinlichkeiten $p_1 - p_2$ kann nicht negativ sein, da X im zweiten Szenarium weniger Informationen und somit höchstens die gleiche Erfolgswahrscheinlichkeit wie im ersten Szenarium hat.

$$\begin{aligned} p_1 - p_2 &= (P[X(k, E(m), PK) = h(m)] - p'_1) - (P[X(k, PK) = h(m)] - p'_2) \\ &= \underbrace{(P[X(k, E(m), PK) = h(m)] - P[X(k, PK) = h(m)])}_{=\nu(k)} + \underbrace{(p'_2 - p'_1)}_{\leq 0} \\ &\leq \nu(k) \end{aligned}$$

Folglich ist der Unterschied vernachlässigbar und das Verschlüsselungsverfahren non-malleable.

□

2.3.2 Non-malleable ElGamal-Verschlüsselung

Gegeben sei ein ElGamal-Schema, d. h. eine zyklische Gruppe G und ein erzeugendes Element g. Es sei s der geheime und $h = g^s$ der öffentliche ElGamal-Schlüssel des Empfängers. Um zu überprüfen, dass der Sender die zur Verschlüsselung verwendete Zufallszahl u kennt, wird das Schnorr-Signaturschema (vgl. Abschnitt 2.5.4) verwendet. Innerhalb des Signaturschemas

betrachtet man den Wert g^u der ElGamal-Nachricht $(g^u, h^u m)$ als öffentlichen Schlüssel mit zugehörigem geheimen Schlüssel u (siehe [TY98]). Die zu signierende Nachricht besteht im Prinzip aus der ursprünglichen ElGamal-Nachricht $(g^u, h^u m)$ und der Identität des Absenders ID_S der Nachricht.

$\mathcal{H}$ sei eine Hashfunktion, die einen Wert c (Frage bzw. Challenge genannt) liefert (vgl. Abschnitt 3.3).

Es sei $c = u \cdot \mathcal{H}\left(g, g^u, h^u m, g^{u'}, ID_S\right) + u' \bmod q$ mit einem zufälligen[3] Wert $u' \in_R \mathbb{Z}_q$. Dann ist die verschlüsselte non-malleable ElGamal-Nachricht:

$$E(m) = \left(g^u, h^u m, g^{u'}, c, ID_S\right).$$

2.3.3 Verifikation und Entschlüsselung

Der Empfänger der Nachricht $E(m) = \left(g^u, h^u m, g^{u'}, c, ID_S\right)$ akzeptiert diese nur, wenn sie die folgende Gleichung erfüllt:

$$g^c = (g^u)^{\mathcal{H}\left(g, g^u, h^u m, g^{u'}, ID_S\right)} \cdot g^{u'}.$$

Dann extrahiert er die Komponenten g^u und $h^u m$ der eigentlichen ElGamal-Nachricht und entschlüsselt:

$$(g^u)^{-s}\, h^u m = g^{-su} g^{su} m = m.$$

Satz 2.4
Das oben vorgestellte erweiterte ElGamal-Verschlüsselungsschema ist non-malleable im Random-Oracle-Modell.

Beweis:
Die ElGamal-Nachricht im erweiterten ElGamal-Verschlüsselungsschema entspricht exakt der Nachricht im Standard-ElGamal-Schema. Da die zusätzlich erzeugten Werte als Zero-Knowledge-Beweis keine Informationen enthalten, außer dass der Sender die zur Verschlüsselung verwendete Zufallszahl kennt, ergibt sich die semantische Sicherheit des vorgestellten Schemas unter der Diffie-Hellman-Entscheidungsannahme aus Lemma 2.2.

Das in dem Verschlüsselungsschema enthaltene Beweissystem ist ein nicht-interaktiver Zero-Knowledge-Beweis, basierend auf dem Schnorr-Signaturschema (siehe Abschnitt 2.5.4). Der Sender kann entsprechend dem Beweis von Satz 2.3 den von ihm erzeugten Geheimtext entschlüsseln. Diese Entschlüsselung erstellt er in diesem nicht-interaktiven Fall mittels einer Oracle-Replay-Technik[4] (vgl. [PS96]). Nach Satz 2.3 ist das Verschlüsselungsschema somit non-malleable.

□

[3] Im Folgenden bedeutet $w \in_R W$, dass w aus W gemäß einer Gleichverteilung ausgewählt wird.

[4] Die Oracle-Replay-Technik basiert auf dem sogenannten *Forking-Lemma* (*Gabelungslemma*), das in [PS96] vorgestellt wird. Die Behandlung dieses Lemma ist jedoch zu umfangreich, um in dieser Arbeit ausgeführt zu werden.

2.4 Modifiziertes ElGamal-Verschlüsselungsverfahren

Das in diesem Abschnitt vorgestellte modifizierte ElGamal-Verschlüsselungsverfahren ist eine vereinfachte Version des Cramer-Shoup-Kryptosystems (vgl. [CS98]). Es bietet im Unterschied zu Cramer-Shoup in der Grundversion lediglich semantische Sicherheit unter einem Angriff mit gewähltem Klartext (vgl. Abschnitt 2.2). Jedoch kann es durch die Ergänzung des Zero-Knowledge-Beweises der zur Verschlüsselung verwendeten Zufallszahl (vgl. Satz 2.3) zu einem Verfahren ausgebaut werden, das semantisch sicher gegen Angriffe mit adaptiv gewählten Geheimtexten ist.

Gegeben sei eine endliche Untergruppe G von $\mathbb{Z}_p^*$ mit Primzahlordnung $|G| = q$, wobei p eine Primzahl ist und $q|p-1$. Die Werte g_1 und g_2 seien erzeugende Elemente dieser Untergruppe und es sei $m \in G$ eine zu verschlüsselnde Nachricht. Alle Multiplikationen sind Gruppenoperationen innerhalb von G. Der Empfänger wählt zwei Zufallszahlen s_1, s_2 als seinen geheimen Schlüssel gemäß einer Gleichverteilung aus $\mathbb{Z}_q$ und berechnet $h = g_1^{s_1} g_2^{s_2}$. Der öffentliche Schlüssel ist dann g_1, g_2, h. Der Sender verschlüsselt die Nachricht m, indem er einen zufälligen Wert $\alpha \in_R \mathbb{Z}_q$ wählt und dem Empfänger das Tripel

$$(g_1^\alpha, g_2^\alpha, h^\alpha m)$$

als Chiffretext übermittelt. Dieser kann das Chiffrat entschlüsseln, indem er mit seinem geheimen Schlüssel den Wert

$$(g_1^\alpha)^{-s_1} \cdot (g_2^\alpha)^{-s_2} \cdot h^\alpha m = g_2^{-\alpha s_2} g_1^{-\alpha s_1} g_1^{\alpha s_1} g_2^{\alpha s_2} m = m$$

berechnet.

2.4.1 Kommutative Verschlüsselung

Für das in Kapitel 6 erstellte erpressungsresistente Wahlsystem wird die Verwendung eines kommutativen Verschlüsselungsverfahrens gefordert. Eine Verschlüsselungsverfahren ist kommutativ, wenn sowohl die Reihenfolge der Verschlüsselungen als auch die Reihenfolge der Entschlüsselungen nicht relevant ist. In diesem Abschnitt wird daher eine kommutative Variante des modifizierten ElGamal-Verschlüsselungsverfahrens vorgestellt, die im Wahlsystem in Abschnitt 6.2 verwendet wird.

Gegeben sei eine endliche Untergruppe G von $\mathbb{Z}_p^*$ mit Primzahlordnung $|G| = q$, wobei p eine Primzahl ist und $q|p-1$. Die Werte g_1 und g_2 seien erzeugende Elemente dieser Untergruppe und es sei $m \in G$ eine zu verschlüsselnde Nachricht. Alle Multiplikationen sind Gruppenoperationen innerhalb der kommutativen Gruppe G.

Zum geheimen Schlüssel $s_1, s_2 \in_R \mathbb{Z}_q$ sei $g_1, g_2, h = g_1^{s_1} g_2^{s_2}$ der zugehörige öffentliche Schlüssel, und $g_1, g_2, h' = g_1^{s'_1} g_2^{s'_2}$ sei der öffentliche Schlüssel zum geheimen Schlüssel $s'_1, s'_2 \in_R \mathbb{Z}_q$.

Es sei

$$(g_1^\alpha, g_2^\alpha, h^\alpha m) \text{ bzw.}$$
$$(g_1^\beta, g_2^\beta, h'^\beta m)$$

der mit Hilfe der Zufallszahlen $\alpha \in_R \mathbb{Z}_q$ bzw. $\beta \in_R \mathbb{Z}_q$ erstellte Chiffretext der Nachricht m. Multipliziert man nun die dritte Komponente des ersten Chiffretextes mit h'^β bzw. des zweiten

Geheimtextes mit h^α und ergänzt g_1^β, g_2^β bzw. g_1^α, g_2^α, so erhält man die folgenden Tupel:

$$(g_1^\alpha, g_2^\alpha, g_1^\beta, g_2^\beta, h'^\beta h^\alpha m)$$
$$(g_1^\beta, g_2^\beta, g_1^\alpha, g_2^\alpha, h^\alpha h'^\beta m)$$

Die Tupel sind unter den beiden öffentlichen Schlüsseln verschlüsselte Geheimtexte der Nachricht m und unterscheiden sich aufgrund der Kommutativität in G lediglich in der Anordnung der Werte g_1^α, g_2^α und g_1^β, g_2^β. Abgesehen von dieser Anordnung kann man also die Reihenfolge der Verschlüsselungen vertauschen.

Es sei nun umgekehrt ein solcher Geheimtext $(g_1^\beta, g_2^\beta, g_1^\alpha, g_2^\alpha, h^\alpha h'^\beta m)$ gegeben. Bei Kenntnis des geheimen Schlüssels s_1, s_2 kann man $h'^\beta m = (g_1^\alpha)^{-s_1} \cdot (g_2^\alpha)^{-s_2} \cdot h^\alpha \cdot h'^\beta m$ bzw. bei Kenntnis von s_1', s_2' kann man $h^\alpha m = (g_1^\beta)^{-s_1'} \cdot (g_2^\beta)^{-s_2'} \cdot h^\alpha \cdot h'^\beta m$ berechnen. Das heißt, man kann den gegebenen Geheimtext zur unter g_1, g_2, h' modifiziert ElGamal-verschlüsselten Nachricht

$$(g_1^\beta, g_2^\beta, h'^\beta m),$$

aber auch zur unter g_1, g_2, h modifiziert ElGamal-verschlüsselten Nachricht

$$(g_1^\alpha, g_2^\alpha, h^\alpha m)$$

entschlüsseln.

Die beiden Verschlüsselungen sind somit kommutativ, d. h. man kann sowohl die Reihenfolge der Verschlüsselungen als auch der Entschlüsselungen vertauschen.

2.5 Digitale Signaturen

Eine digitale Signatur soll ein elektronisches Pendant zur handschriftlichen Unterschrift darstellen.

2.5.1 Digitales Signaturschema

Ein digitales Signaturschema ist ein Tripel (G, S, V) dreier probabilistischer Algorithmen.
Die Schlüsselerzeugung G erstellt bei Eingabe eines Sicherheitsparameters k ein Schlüsselpaar, bestehend aus dem öffentlichen Schlüssel PK und dem privaten Schlüssel SK.
Der Signaturalgorithmus S berechnet unter Eingabe des Sicherheitsparameters k und geheimen Schlüssels SK die Signatur $sig = S(k, m, SK)$ einer Nachricht $m \in \{0,1\}^k$.
Der Verifikationsalgorithmus V erhält als Eingabe k, PK, sig und m und gibt 'wahr' oder 'falsch' aus. Wenn PK zu SK passt, ist die Ausgabe $V(k, PK, m, S(k, SK, m))$ ='wahr'. Ein Angreifer, der weder SK noch die Signatur zu einer Nachricht m kennt, darf nur mit einer vernachlässigbaren Wahrscheinlichkeit einen Wert sig' erzeugen können, so dass $V(k, PK, m, sig')$ ='wahr' ist. Man unterscheidet vier Angriffsarten, je nachdem welche Informationen der Angreifer hat (siehe Abschnitt 2.5.2).

Der private Schlüssel wird also mit einer Signaturfunktion auf das zu signierende Dokument so angewendet, dass jeder mit einer Verifikationsfunktion unter dem öffentlichen Schlüssel die Signatur überprüfen kann.

Die rechtliche Grundlage digitaler Signaturen ist durch das Signaturgesetz [Sig97] und [Sig01] gegeben. Zur eindeutigen Zuordnung der öffentlichen Signaturschlüssel zu ihren Benutzern sieht das Signaturgesetz eine vertrauenswürdige Instanz zur Zertifizierung der Signaturschlüssel vor (siehe Abschnitt 2.2.3).

Man kann eine digitale Signatur und das Zertifikat mit einem gläsernen Tresor und eingraviertem Namen des Besitzers vergleichen. Nur der Besitzer des Tresors verfügt über den Schlüssel und kann Nachrichten in diesen Tresor einschließen. Die Nachricht kann aber von jedem gelesen und anhand der Gravur des Tresors eindeutig dem Besitzer zugeordnet werden. Jeder kann sich sicher sein, dass nur dieser die Nachricht in den Tresor gelegt haben kann.

2.5.2 Angriffstypen auf digitale Signaturen

Entsprechend den Angriffstypen auf Verschlüsselunsverfahren kann man auch bei digitalen Signaturen vier Angriffsarten unterscheiden.

- Der schwächste Angriff ist der *Angriff ohne bekannte Signaturen* (key only attack). Hier kennt der Angreifer lediglich den öffentlichen Schlüssel des Signierers.
- Kennt der Angreifer den öffentlichen Schlüssel und mehrere Paare aus Nachricht und zugehöriger Signatur, so spricht man von einem *Angriff mit bekannten Signaturen* (known signature attack).
- Ein Angriff, bei dem der Angreifer nicht nur den öffentlichen Schlüssel kennt, sondern selbst Nachrichten wählen kann, zu denen der Signierer korrekte Signaturen erzeugt, heißt *Angriff mit gewählten Nachrichten* (chosen message attack).
- Kann ein Angreifer die zu signierenden Nachrichten in Abhängigkeit des Angriffverlaufs nacheinander wählen, so ist das ein *adaptiver Angriff mit gewählten Nachrichten* (adaptive chosen message attack).

2.5.3 Mögliche Erfolgsstufen eines Angriffs

Ein Angriff auf ein Signaturschema kann zu unterschiedlichen Erfolgsstufen führen.

- Kann ein Angreifer zu einer nicht notwendigerweise von ihm selbst gewählten Nachricht eine Signatur erstellen, so spricht man von *existentieller Fälschbarkeit* (existential forgery).
- Wenn der Angreifer zu einer oder mehreren selbst gewählten Nachrichten Signaturen fälschen kann, so nennt man das *selektive Fälschbarkeit* (selective forgery).
- Gelingen diese Fälschungen dem Angreifer zu jeder beliebigen Nachricht, so heißt diese Erfolgsstufe *universelle Fälschbarkeit* (universal forgery).
- Schließlich könnte es einem Angreifer noch gelingen, den privaten Signaturschlüssel zu berechnen. Diese *Kompromittierung des Schlüssels* bezeichnet man als total break.

2.5.4 Das Schnorr-Signaturschema [Sch91]

Es sei G eine Gruppe und M der Raum der signierbaren Nachrichten. Schnorr-Signaturen basieren auf einer Hashfunktion $\mathcal{H} : G \times M \to \mathbb{Z}_q$. Der private Schlüssel des Signierers sei x, der öffentliche $h = g^x$.

Zu einer gegebenen Nachricht $m \in M$ wählt der Signierer eine Zahl $r \in_R \mathbb{Z}_q$, berechnet g^r und $c := \mathcal{H}(g^r, m)$ sowie $z := r + cx \bmod q$. Als Signatur sendet er das Tripel (m, c, z).
Die Signatur ist gültig, wenn $g^z h^{-c} = g^{r+cx} h^{-c} = g^r$ und daher $\mathcal{H}(g^z h^{-c}, m) = c$ gilt.

Unter kryptografischen Standardannahmen sind Schnorr-Signaturen bei einem Angriff mit adaptiv gewählten Nachrichten nicht existentiell fälschbar. Nähere Angaben und weitere Sicherheitsuntersuchungen von Schnorr-Signaturen sind in der Veröffentlichung von Schnorr und Jakobsson [SJ99] zu finden.

Bad officials are elected by
good citizens who do not vote.

George Jean Nathan

Kapitel 3

Kryptografische Bausteine

In diesem Kapitel werden die in den Wahlprotokollen der Kapitel 5 und 6 verwendeten kryptografischen Bausteine und Komponenten, wie z. B. Zero-Knowledge-, Designated-Verifier-, Witness-Indistinguishable-Beweise oder MIX-Netze vorgestellt und analysiert.

3.1 Interaktive Beweise, Beweissystem

Beweissysteme dienen unter anderem dazu, den korrekten Protokollablauf zu gewährleisten, indem Teilnehmer anderen Teilnehmern oder Beobachtern beweisen, dass sie das Protokoll korrekt befolgt haben. Im Zusammenhang mit elektronischen Wahlen kann beispielsweise ein Wähler beweisen, eine korrekte Wahloption verschlüsselt zu haben.

3.1.1 Aufbau und Eigenschaften von Beweissystemen

In einem Beweissystem versucht der Prover (Beweisführer), dem Verifier (Verifizierer) die Korrektheit einer Aussage zu beweisen.

Definition 3.1 (Beweissystem)
Gegeben sei ein Alphabet Σ und eine Sprache L - eine Teilmenge der Menge Σ^* aller endlichen Bitstrings. Ein Beweissystem für eine Behauptung $\ell \in L$ ist ein Paar (Prover,Verifier) von Turing-Maschinen, das durchführbar und korrekt ist. Die Eigenschaften der Durchführbarkeit und Korrektheit seien dabei wie folgt definiert.

Definition 3.2 (Durchführbarkeit (completeness))
Ein Beweissystem heißt *durchführbar*, wenn es für jede Aussage ℓ aus der Sprache L ein $N_0 \in \mathbb{N}$ und eine vernachlässigbare Funktion ν gibt, so dass mit $|\ell| = N \geq N_0$ gilt:

$$P[\text{Verifier akzeptiert}] \geq 1 - \nu(N).$$

Man sagt also, dass ein interaktiver Beweis, in dem der Verifier den Beweis immer akzeptiert, wenn der Prover ehrlich ist und beide das Protokoll korrekt befolgen, durchführbar ist.

Definition 3.3 (Korrektheit (soundness))
Ein Beweissystem heißt *korrekt*, wenn es für jede Aussage $\ell \notin L$ ein $N_0 \in \mathbb{N}$ und eine vernachlässigbare Funktion μ gibt, so dass mit $|\ell| = N \geq N_0$ gilt:

$$P[\text{Verifier lehnt ab}] \geq 1 - \mu(N).$$

Das bedeutet, dass man einen Beweissystem korrekt nennt, wenn ein unehrlicher Prover, der versucht eine falsche Behauptung zu beweisen, mit hoher Wahrscheinlichkeit scheitert.

Die Kommunikation kann interaktiv zwischen den Teilnehmern ablaufen. Es gibt aber auch Beweissysteme, bei denen der Prover den Beweis ohne Eingabe des Verifiers erstellen kann. Bei interaktiven Beweisen wird ein Challenge-Response-Protokoll durchgeführt, in dem der Prover eine Antwort (Response) auf die vom Verifier geschickte Frage (Challenge) schicken muss. Die Gültigkeit des Beweises kann am Ende vom Verifier akzeptiert oder abgelehnt werden.

Definition 3.4 (Interaktiver Beweis)
Gegeben sei eine Sprache L, also eine Teilmenge der Menge Σ^* aller endlichen Bitstrings. Ein interaktiver Beweis für eine Behauptung $\ell \in L$ ist ein Paar (Prover,Verifier) interaktiver Turing-Maschinen, das durchführbar und korrekt ist.

3.2 Zero-Knowledge-Beweise

Eine spezielle Form von Beweissystemen sind sogenannte Zero-Knowledge-Beweise.

3.2.1 Simulierbarkeit, Zero-Knowledge-Eigenschaft

In einem Zero-Knowledge-Beweis erfährt der Verifier während des Beweises nur, dass der Prover einen Beweis für die Behauptung kennt.

Der Beweis muss sich von einem Simulator, der das Geheimnis selbst nicht kennt, effizient, also mit einer in der Eingabelänge polynomiellen Laufzeit, simulieren lassen.

Definition 3.5 (Zero-Knowledge-Beweis)
Gegeben sei eine Sprache L, ein interaktiver Beweis (Prover,Verifier) für die Behauptung $\ell \in L$ und die Protokollansicht *view* des Verifiers. Dann nennt man (Prover,Verifier) einen (perfekten) Zero-Knowledge-Beweis, wenn es eine polynomielle Turing-Maschine M (den Simulator) gibt, so dass für alle $\ell \in L$ die Ausgaben von view und M identisch verteilt sind.

Das bedeutet, dass man eine zur tatsächlichen Protokolldurchführung identische Protokollansicht ohne Kenntnis des Beweises für $\ell \in L$ effizient erstellen können muss. Diese Simulierbarkeit des Protokollablaufs ist die *Zero-Knowledge-Eigenschaft*.

3.2.2 Interaktiver Beweis der Gleichheit K diskreter Logarithmen

David Chaum et al. stellten 1987 eine Lösung zu folgendem Problem vor (vgl. [CEG87]). Gegeben sei eine Gruppe G mit Primzahlordnung $|G| = q$, einem Erzeuger g von G und einem Wert

$\alpha \in \mathbb{Z}_q$. Der Prover veröffentlicht einen Wert $\beta := g^\alpha$. Er möchte den Verifier davon überzeugen, dass er den Wert α kennt, ohne jedoch α zu verraten. Der folgende interaktive Beweis stellt eine Verallgemeinerung des ursprünglichen Problems dar. Ein Prover kennt eine Lösung für K diskrete Logarithmen in einer Gruppe G mit Primzahlordnung $|G| = q$. Das bedeutet, dass er einen Wert $\alpha \in \mathbb{Z}_q$ kennt, so dass $g_i^\alpha = \beta_i$ für $i = 1, 2, \ldots, K$ gilt. Der Prover will also beweisen, dass $\alpha = \log_{g_1}(\beta_1) = \ldots = \log_{g_K}(\beta_K)$ gilt, ohne α zu offenbaren. Der interaktive Beweis besteht aus den folgenden Schritten.

1. Der Prover wählt einen Wert $a \in_R \mathbb{Z}_q$ und sendet die Werte $b_i := g_i^a$ für $i = 1, 2, \ldots, K$ an den Verifier.
2. Der Verifier wählt ein Bit $c \in_R \{0, 1\}$ und übermittelt es an den Prover.
3. Der Prover berechnet $r := a + c \cdot \alpha \bmod q$ und sendet r an den Verifier.
4. Der Verifier überprüft, ob die Gleichungen $g_i^r \stackrel{?}{=} b_i \beta_i^c$ für $i = 1, 2, \ldots, K$ erfüllt sind.

Wenn der Prover den Wert α nicht kennt, könnte er c vorher mit Wahrscheinlichkeit $\frac{1}{2}$ raten und die Werte $g_i^r \beta_i^{-c}$ als b_i an den Verifier senden. Daher werden die oben beschriebenen Schritte N-mal durchgeführt. N ist hierbei der Sicherheitsparameter. Mit jeder Durchführung halbiert sich diese Betrugswahrscheinlichkeit.

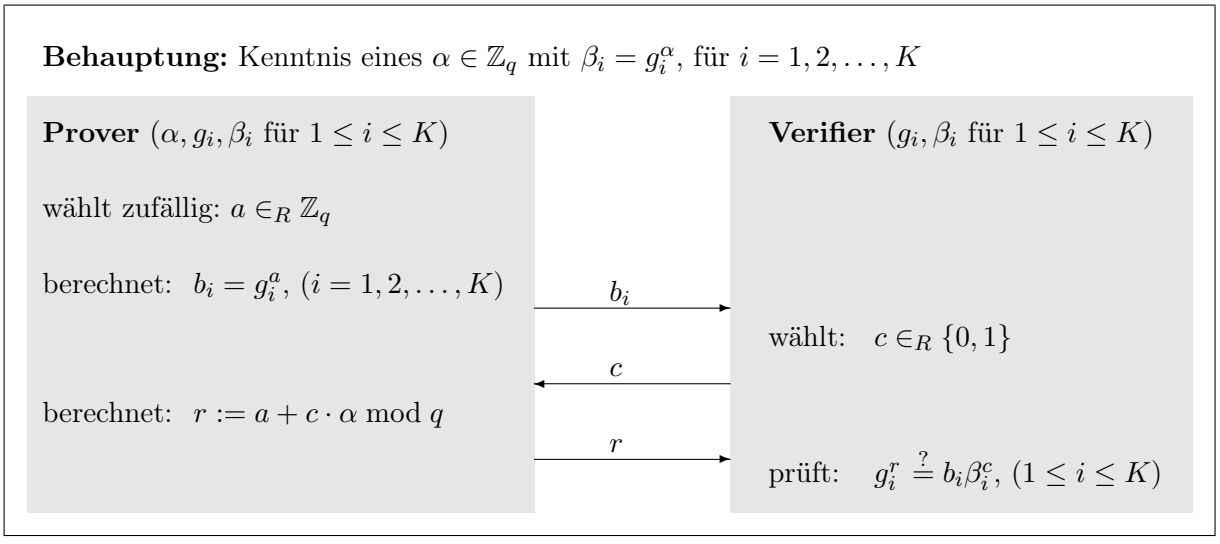

Abbildung 3.1: Interaktiver Beweis der Gleichheit K diskreter Logarithmen.

Erläuterung: In der Abbildung und den folgenden Abbildungen von Protokollen sind in den Klammern hinter der am Protokoll beteiligten Parteien die der Partei bekannten Werte angegeben.

Satz 3.1
Der interaktive Beweis der Gleichheit K diskreter Logarithmen besitzt die Zero-Knowledge-Eigenschaft. Er ist durchführbar und korrekt.

Beweis:

- Der interaktive Beweis ist *durchführbar*. Wenn sich alle Teilnehmer korrekt verhalten, der Wert α Lösung aller K diskreten Logarithmen $\log_{g_i}(\beta_i)$ für $i = 1, 2, \ldots, K$ ist, und der Prover diese Lösung α kennt, kann er immer eine korrekte Antwort auf die Challenge des Verifiers geben. Der Prover berechnet die Antwort r auf die Frage c durch $r = a + c \cdot \alpha \bmod q$, so dass $g_i^r = g_i^{a+c\cdot\alpha} = g_i^a \cdot (g_i^\alpha)^c = b_i \cdot \beta_i^c$ gilt.

- Das Protokoll ist *korrekt*. Kennt der Prover den Wert α der diskreten Logarithmen nicht oder nicht alle diskreten Logarithmen besitzen den gleichen Wert α, so kann der Prover in jeder Durchführung die Antwort r nur für einen der zwei möglichen Werte von c berechnen. Um erfolgreich zu betrügen, muss er c vorher erraten, einen Wert $r \in_R \mathbb{Z}_q$ wählen und dann im ersten Schritt des Beweises die Werte $b_i := g_i^r \beta_i^{-c}$ zum Verifier senden.

  In jedem Fall kann der Prover nur auf eines der beiden möglichen Bits eine korrekte Antwort berechnen. Angenommen, er könnte für beide möglichen Werte von c korrekte Antworten r_0 und r_1 berechnen, so könnte er auch die Lösungen[1] α_i berechnen, denn es gilt:

  $$\left.\begin{array}{l} g_i^{r_0} = b_i \cdot (g_i^{\alpha_i})^0 = b_i \\ g_i^{r_1} = b_i \cdot (g_i^{\alpha_i})^1 = b_i \cdot g_i^{\alpha_i} \end{array}\right\} \Rightarrow g_i^{r_1 - r_0} = g_i^{r_1} g_i^{-r_0} = b_i \cdot g_i^{\alpha_i} b_i^{-1} = g_i^{\alpha_i} \Rightarrow r_1 - r_0 = \alpha_i.$$

  Angenommen, es gilt $\alpha_j := \log_{g_j}(\beta_j) \neq \log_{g_k}(\beta_k) =: \alpha_k$ für $j, k \in \{1, 2, \ldots, K\}$, also nicht alle diskreten Logarithmen sind gleich, dann muss für die vom Prover an den Verifier geschickten Werte b_j und b_k das Folgende gelten, damit der Verifier die Gleichungen $g_i^r = b_i \beta_i^c$ (für $i = 1, 2, \ldots, K$) akzeptieren kann:

  $$\begin{array}{lll} b_j = g_j^r & \text{und } b_k = g_k^r & \text{für } c = 0, \\ b_j = g_j^{r-\alpha_j} & \text{und } b_k = g_k^{r-\alpha_k} & \text{für } c = 1. \end{array}$$

  Da $\alpha_j \neq \alpha_k$ ist, gilt $\log_{g_j}(b_j) = \log_{g_k}(b_k)$ für $c = 0$ und $\log_{g_j}(b_j) \neq \log_{g_k}(b_k)$ für $c = 1$. Der Prover kann also auch in dem Fall, dass nicht alle diskreten Logarithmen den gleichen Wert besitzen, nur dann eine korrekte Antwort r auf die Frage c des Verifiers geben, wenn er diese Challenge c vorausahnt.

  Also kann der Prover in jeder Durchführung mit einer Wahrscheinlichkeit von $\frac{1}{2}$ betrügen. Die Wahrscheinlichkeit, dass sein Betrugsversuch bei N Durchführungen mindestens einmal aufgedeckt wird, ist dann $1 - 2^{-N}$. Somit ist das Protokoll korrekt.

- Das Protokoll besitzt die Zero-Knowledge-Eigenschaft. Es lässt sich von einem polynomiell zeitbeschränkten Simulator, der eine Lösung α oder verschiedene Lösungen α_i $(i = 1, 2, \ldots, K)$ nicht kennt, simulieren.

  1. Der Simulator wählt einen Wert $d \in_R \{0, 1\}$, einen Wert $r \in_R \mathbb{Z}_q$, berechnet zunächst die K Werte $b_i := g_i^r \beta_i^{-d}$, $(i = 1, 2, \ldots, K)$, und sendet diese b_i an den Verifier.
  2. Der Verifier wählt $c \in_R \{0, 1\}$ und übermittelt es an den Simulator.

[1] Der Prover könnte für jedes $i = 1, 2, \ldots, K$ eine Lösung α_i berechnen, die natürlich auch alle gleich sein könnten.

3. Der Simulator vergleicht d mit c.
   - Ist $d = c$, so sendet er r an den Verifier.
     Dieser stellt fest, dass

     $$g_i^r = b_i \cdot \beta_i^c \text{ für } i = 1, 2, \ldots, K$$

     gilt[2], und die Simulation war erfolgreich[3].
   - Ist $c \neq d$, so wird die Runde gelöscht.

Da r gemäß einer Gleichverteilung aus $\mathbb{Z}_q$ ausgewählt wird, ist auch $r - d \cdot \alpha$ über $\mathbb{Z}_q$ gleichverteilt und die Werte $b_i = g_i^r \beta_i^{-d} = g_i^{r-d\cdot\alpha}$ des Simulators haben die gleiche Verteilung wie die Werte $b_i = g_i^a$ für $a \in \mathbb{Z}_q$ des Provers. Die Werte c und d sind beide unabhängig voneinander gewählt. Die Wahrscheinlichkeit, dass $d = c$ gilt, ist $\frac{1}{2}$. Da die Runden für $c \neq d$ gelöscht werden, besitzen c und d in den Runden für $c = d$ die gleiche Verteilung. Somit besitzen auch die Werte $r = a + c \cdot \alpha$ des Provers bzw. $r = a + d \cdot \alpha$ die gleiche Verteilung.

Der Simulator benötigt voraussichtlich $2N$ Runden, um N Durchführungen erfolgreich zu simulieren. Dies ist also eine effiziente Simulation.

□

3.2.3 Zero-Knowledge-Beweis $\mathbf{ZK}(\alpha, \beta : \mathbf{d} = \mathbf{g}^\alpha \mathbf{h}^\beta, \tilde{\mathbf{x}}_\mathbf{1} = \mathbf{x}_\mathbf{1}^\alpha, \tilde{\mathbf{x}}_\mathbf{2} = \mathbf{x}_\mathbf{2}^\alpha, \tilde{\mathbf{y}} = \mathbf{y}^\beta)$

Gegeben sei eine Gruppe G mit Primzahlordnung $|G| = q$ und Erzeuger g, x_1, x_2, y von G. In diesem Zero-Knowledge-Beweis behauptet der Prover, Werte $\alpha, \beta \in \mathbb{Z}_q$ zu kennen, so dass die Gleichungen $d = g^\alpha h^\beta$, $\tilde{x}_1 = x_1^\alpha$, $\tilde{x}_2 = x_2^\alpha$ und $\tilde{y} = y^\beta$ erfüllt sind.
Die Werte $d, g, h, x_1, \tilde{x}_1, x_2, \tilde{x}_2, y, \tilde{y}$ sind Prover und Verifier bekannt. Der interaktive Beweis besteht aus den folgenden Schritten.

1. Der Prover wählt Werte $a, a' \in_R \mathbb{Z}_q$ und sendet $b_{1,1} = x_1^a$, $b_{2,1} = x_2^a$, $b_2 = y^{a'}$ und $b_3 = g^a h^{a'}$ an den Verifier.

2. Der Verifier wählt ein Bit $c \in_R \{0, 1\}$ und übermittelt es an den Prover.

3. Der Prover berechnet $r = a + c \cdot \alpha$ und $r' = a' + c \cdot \beta$ und sendet die Werte an den Verifier.

4. Der Verifier überprüft, ob die Gleichungen

   $$\begin{aligned} x_1^r &\stackrel{?}{=} b_{1,1}\tilde{x}_1^c, \\ x_2^r &\stackrel{?}{=} b_{2,1}\tilde{x}_2^c, \\ y^{r'} &\stackrel{?}{=} b_2\tilde{y}^c, \\ g^r h^{r'} &\stackrel{?}{=} b_3 \cdot d^c \end{aligned}$$

   erfüllt sind.

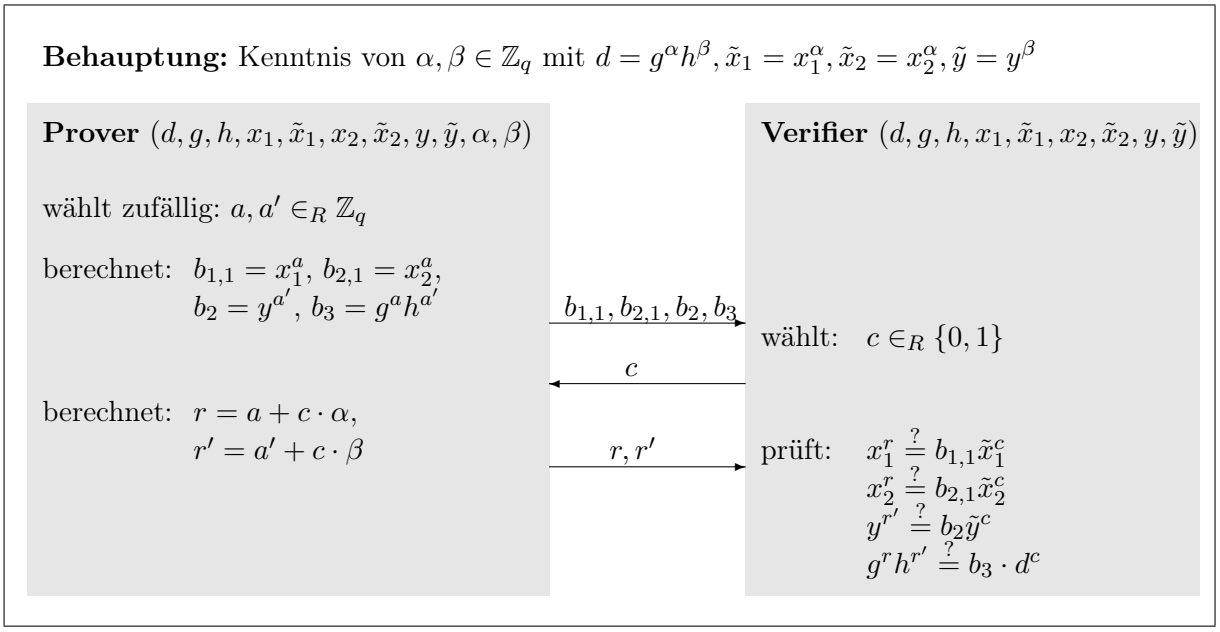

Abbildung 3.2: ZK$(\alpha, \beta : d = g^\alpha h^\beta, \tilde{x}_1 = x_1^\alpha, \tilde{x}_2 = x_2^\alpha, \tilde{y} = y^\beta)$.

Diese Schritte werden $N-$mal durchgeführt. N ist der Sicherheitsparameter.

Satz 3.2
Der interaktive Beweis ZK$(\alpha, \beta : d = g^\alpha h^\beta, \tilde{x}_1 = x_1^\alpha, \tilde{x}_2 = x_2^\alpha, \tilde{y} = y^\beta)$ ist durchführbar, korrekt und besitzt die Zero-Knowledge-Eigenschaft.

Beweis:

- Der interaktive Beweis ist *durchführbar*. Wenn sich alle Teilnehmer korrekt verhalten und der Prover die Werte α, β kennt, so dass $d = g^\alpha h^\beta$, $\tilde{x}_1 = x_1^\alpha$, $\tilde{x}_2 = x_2^\alpha$, und $\tilde{y} = y^\beta$) gilt, kann er immer korrekt auf die Challenge des Verifiers antworten. Der Prover berechnet die Antworten r, r' auf die Frage c durch $r = a + c \cdot \alpha$ und $r' = a' + c \cdot \beta$, so dass die folgenden Gleichungen erfüllt sind:
  $x_1^r = x_1^{a+c\cdot\alpha} = x_1^a \cdot (x_1^\alpha)^c = b_{1,1} \cdot \tilde{x}_1^c$,
  $x_2^r = x_2^{a+c\cdot\alpha} = x_2^a \cdot (x_2^\alpha)^c = b_{2,1} \cdot \tilde{x}_2^c$,
  $y^{r'} = y^{a'+c\cdot\beta} = y^{a'} \cdot (y^\beta)^c = b_2 \cdot \tilde{y}^c$ und
  $g^r h^{r'} = g^{a+c\cdot\alpha} h^{a'+c\cdot\beta} = g^a g^{c\cdot\alpha} h^{a'} h^{c\cdot\beta} = g^a h^{a'} (g^\alpha h^\beta)^c = b_3 d^c$.

- Das Protokoll ist *korrekt*. Kennt der Prover den Wert α oder den Wert β nicht, so kann er in jeder Durchführung die Antworten r und r' nur für einen der zwei möglichen Werte von c berechnen. Um erfolgreich zu betrügen, muss er c vorher erraten, Werte $r, r' \in_R \mathbb{Z}_q$ wählen und dann im ersten Schritt des Beweises die Werte $b_{1,1} = x_1^r \tilde{x}_1^{-c}$, $b_{2,1} = x_2^r \tilde{x}_2^{-c}$, $b_2 = y^{r'} \tilde{y}^{-c}$ und $b_3 = g^r h^{r'} d^{-c}$ zum Verifier senden.

[2] Denn für $c = d$ gilt: $g_i^r = g_i^r \beta_i^{-d} \cdot \beta_i^c = b_i \cdot \beta_i^c$.

[3] Da der Simulator die Werte α_i, für $i = 1, 2, \ldots, K$, nicht kennt, die Simulation aber dennoch ohne diese Werte α_i durchführen kann, ist es für die Simulation nicht relevant, ob die α_i alle den gleichen Wert haben.

In jedem Fall kann der Prover nur auf eines der beiden möglichen Bits korrekte Antworten berechnen. Angenommen, er könnte für beide möglichen Werte von c korrekte Antworten r_0, r_0' bzw. r_1, r_1' berechnen, ohne α oder β zu kennen. Dann gilt:

$$\begin{array}{lll} x_1^{r_0} & = b_{1,1} & \\ x_2^{r_0} & = b_{2,1} & \text{aber auch} \\ y^{r_0'} & = b_2 & \\ g^{r_0} h^{r_0'} & = b_3 & \end{array} \qquad \begin{array}{ll} x_1^{r_1} & = b_{1,1} \cdot \tilde{x}_1 \\ x_2^{r_1} & = b_{2,1} \cdot \tilde{x}_2 \\ y^{r_1'} & = b_2 \cdot \tilde{y} \\ g^{r_1} h^{r_1'} & = b_3 \cdot d \end{array}$$

Ersetzt man in den Gleichungen auf der rechten Seite $b_{1,1}$, $b_{2,1}$, b_2 bzw. b_3, so erhält man:

$$\begin{array}{ll} x_1^{r_1} & = x_1^{r_0} \cdot \tilde{x}_1 \\ x_2^{r_1} & = x_2^{r_0} \cdot \tilde{x}_2 \\ y^{r_1'} & = y^{r_0'} \cdot \tilde{y} \\ g^{r_1} h^{r_1'} & = g^{r_0} h^{r_0'} \cdot d \end{array} \qquad \text{also} \qquad \begin{array}{ll} x_1^{r_1 - r_0} & = \tilde{x}_1 \\ x_2^{r_1 - r_0} & = \tilde{x}_2 \\ y^{r_1' - r_0'} & = \tilde{y} \\ g^{r_1 - r_0} h^{r_1' - r_0'} & = d \end{array}$$

Das heißt, dass der Prover $\alpha = r_1 - r_0$ und dann auch $\beta = r_1' - r_0'$ berechnen kann. Dies ist ein Widerspruch zur Annahme, dass er α oder β nicht kennt.

Also kann der Prover in jeder Durchführung mit einer Wahrscheinlichkeit von $\frac{1}{2}$ betrügen. Die Wahrscheinlichkeit, dass sein Betrugsversuch bei N Durchführungen mindestens einmal aufgedeckt wird, ist dann $1 - 2^{-N}$.

- Das Protokoll besitzt die Zero-Knowledge-Eigenschaft. Es lässt sich von einem polynomiell zeitbeschränkten Simulator, der die Lösung α oder β nicht kennt, simulieren.

  1. Der Simulator wählt einen Wert $c' \in_R \{0,1\}$, Werte $r, r' \in_R \mathbb{Z}_q$, berechnet zunächst die Werte $b_{1,1} = x_1^r \tilde{x}_1^{-c'}$, $b_{2,1} = x_2^r \tilde{x}_2^{-c'}$, $b_2 = y^{r'} \tilde{y}^{-c'}$, $b_3 = g^r h^{r'} d^{-c'}$ und sendet diese an den Verifier.
  2. Der Verifier wählt $c \in_R \{0,1\}$ und sendet es an den Simulator.
  3. Der Simulator vergleicht c' mit c.
     - Ist $c' = c$, so sendet er r, r' an den Verifier. Dieser stellt fest, dass die Verifikationsgleichungen
     $$\begin{array}{lcl} x_1^r & = & b_{1,1} \tilde{x}_1^c, \\ x_2^r & = & b_{2,1} \tilde{x}_2^c, \\ y^{r'} & = & b_2 \tilde{y}^c, \\ g^r h^{r'} & = & b_3 \cdot d^c \end{array}$$
     erfüllt sind, und die Simulation war erfolgreich.
     - Ist jedoch $c' \neq c$, so wird die Runde gelöscht.

Da r, r' gemäß einer Gleichverteilung aus $\mathbb{Z}_q$ ausgewählt werden, sind auch $r - c' \cdot \alpha$ und $r' - c' \cdot \beta$ über $\mathbb{Z}_q$ gleichverteilt. Die Werte $b_{1,1} = x_1^r \tilde{x}_1^{-c'} = x_1^{r - c' \cdot \alpha}$ des Simulators und die Werte $b_{1,1} = x_1^a$ für $a \in \mathbb{Z}_q$ des Provers besitzen somit die gleiche Verteilung. Analog weisen die Werte $b_{2,1} = x_2^r \tilde{x}_2^{-c'} = x_2^{r - c' \cdot \alpha}$ des Simulators und die Werte $b_{2,1} = x_2^a$ des Provers die gleiche Verteilung auf. Gleiches gilt für die Werte $b_2 = y^{r'} \tilde{y}^{-c'} = y^{r' - c' \cdot \beta}$ des Simulators

und $b_2 = y^{a'}$ für $a' \in \mathbb{Z}_q$ des Provers sowie für die Werte $b_3 = g^r h^{r'} d^{-c'} = g^{r-c'\cdot\alpha} h^{r'-c'\cdot\beta}$ des Simulators und $b_3 = g^a h^{a'}$ des Provers. Die Werte c und c' sind beide unabhängig voneinander gewählt. Die Wahrscheinlichkeit, dass $c' = c$ gilt, ist $\frac{1}{2}$. Da die Runden für $c \neq c'$ gelöscht werden, besitzen c und c' in den Runden für $c = c'$ die gleiche Verteilung. Somit besitzen auch die Werte $r = a + c \cdot \alpha$ des Provers und $r = a + c' \cdot \alpha$ sowie die Werte $r' = a' + c \cdot \beta$ des Provers und $r' = a' + c' \cdot \beta$ die gleiche Verteilung.

Der Simulator benötigt voraussichtlich $2N$ Runden, um N Durchführungen erfolgreich zu simulieren. Dies ist also eine effiziente Simulation. □

3.3 Parallele Ausführung und Fiat-Shamir-Heuristik

Wird die Challenge des Verifiers in einem interaktiven Beweis als String statt als Bit gewählt, so bleibt die Zero-Knowledge-Eigenschaft nur gegenüber einem ehrlichen Verifier erhalten, der die Challenge zufällig und unabhängig von den erhaltenen Werten wählt (siehe z. B. [GK96]). Man nennt diese schwächere Eigenschaft *Honest-Verifier-Zero-Knowledge*.

Man kann darüber hinaus einen 3-Runden-Beweis auch ohne Interaktion mit einem Verifier durchführen, wenn die Frage (Challenge) des Verifiers vom Prover durch eine Hashfunktion erzeugt wird. Dazu kann man nach [FS86] eine kryptografische Hashfunktion verwenden, die die Werte des Provers aus dem ersten Schritt und die dem Verifier (und Prover) bekannten Werte (sogenanntes Environment) als Eingabewerte verwendet.
Diese Idee von Fiat und Shamir, dass die Hashfunktion das zufällige Auswählen simuliert, wurde von Bellare und Rogaway mittels des Random-Oracle-Modells in [BR93] genauer formalisiert. Aufgrund der Einwegeigenschaft der Hashfunktion kennt der Prover die Challenge erst, wenn er die Commitments gebildet und in die Hashfunktion eingegeben hat. Wegen der Kollisionsresistenz der Hashfunktion kann er diese auch nicht mehr ändern.

3.4 Nicht-interaktiver Zero-Knowledge-Beweis der non-malleability für zwei Prover und zwei Verschlüsselungen

In den Wahlsystemen, bei denen der Wähler seine Stimme zusammen mit einem Observer erzeugt, muss er den Zero-Knowledge-Beweis, die zur Verschlüsselung verwendeten Zufallszahlen zu kennen, ebenfalls gemeinsam mit dem Observer durchführen. Dazu muss das in Abschnitt 2.3 vorgestellte Verfahren für zwei Prover, also den Wähler als Prover 1 und den Observer als Prover 2, erweitert werden. Darüber hinaus wird die Wahlberechtigung des Wählers bei dem in Kapitel 6 vorgestellten erpressungsresistenten Wahlsystem ebenfalls non-malleable verschlüsselt und in die abgegebene Stimme integriert.

In diesem neuen nicht-interaktiven Zero-Knowledge-Beweis für zwei Prover und zwei Verschlüsselungen erhält der Observer als Prover 2 nur verschlüsselte Werte des Wählers (Prover 1). Die Werte, die der Observer erzeugt, können später im tatsächlichen Wahlprotokoll noch durch Designated-Verifier-Beweise (vgl. Abschnitt 3.5) durch den Wähler verifiziert werden.

3.4.1 Verschlüsselung

Gegeben sei eine endliche Untergruppe G von $\mathbb{Z}_p^*$ mit Primzahlordnung $|G| = q$, wobei p eine Primzahl ist und $q|p-1$ gilt. Die Werte g und γ seien erzeugende Elemente von G. Der geheime Schlüssel sei s und $h = g^s$ sei der öffentliche ElGamal-Schlüssel des Empfängers.

Wie in Abschnitt 2.3 wird das Schnorr-Signaturschema (vgl. Abschnitt 2.5.4) benutzt, um zu überprüfen, dass der Sender die zur Verschlüsselung verwendete Zufallszahl u kennt.

$\mathcal{H}$ sei eine kryptografische Hashfunktion, die einen Wert c (Frage bzw. Challenge genannt) liefert. Ebenso kann man anstelle der Hashfunktion einen echten Zufallszahlengenerator einsetzen. Prover 1 wählt Zufallszahlen $a, a' \in_R \mathbb{Z}_q$ und berechnet eine probabilistische Verschlüsselung $(g^a, h^a m)$ der Nachricht m sowie den Wert $g^{a'}$, die er an Prover 2 sendet. Dieser wählt Zufallszahlen $b, b', c, c' \in_R \mathbb{Z}_q$, berechnet eine Wiederverschlüsselung[4] $(g^{a+b}, h^{a+b}m)$ und eine Verschlüsselung $(g^c, h^c\sigma)$ der Nachricht σ. Diese Nachricht σ ersetzt das Identifikationsmerkmal aus Abschnitt 2.3.
Er berechnet außerdem die Werte

$$g^{a'+b'+c'} \quad \text{und} \quad (b+c) \cdot \mathcal{H}(g, g^{a+b}, h^{a+b}m, g^{a'+b'+c'}, g^c, h^c\sigma) + (b'+c').$$

Die berechneten Werte

$$(g^{a+b}, h^{a+b}m), (g^c, h^c\sigma), g^{a'+b'+c'} \quad \text{und} \quad (b+c) \cdot \mathcal{H}(g, g^{a+b}, h^{a+b}m, g^{a'+b'+c'}, g^c, h^c\sigma) + (b'+c')$$

schickt er zurück an Prover 1, der

$$d := (a+b+c) \cdot \mathcal{H}(g, g^{a+b}, h^{a+b}m, g^{a'+b'+c'}, g^c, h^c m) + (a'+b'+c')$$

berechnet.
Die verschlüsselte ElGamal-Nachricht, die sowohl im Hinblick auf σ als auch auf m non-malleable ist, ist dann

$$E(m) = (g^{a+b}, h^{a+b}m, g^{a'+b'+c'}, d, g^c, h^c\sigma).$$

3.4.2 Entschlüsselung und Verifikation

Der Empfänger dieser Nachricht $E(m)$ extrahiert die Komponenten g^{a+b}, $h^{a+b}m$ und g^c, $h^c\sigma$ der ElGamal-Nachrichten. Er akzeptiert die Nachricht nur, wenn sie folgende Gleichung erfüllt:

$$g^d = (g^{a+b} \cdot g^c)^{\mathcal{H}(g, g^{a+b}, h^{a+b}m, g^{a'+b'+c'}, g^c, h^c\sigma)} \cdot g^{a'+b'+c'}.$$

Dann entschlüsselt er:

$$(g^{a+b})^{-s} h^{a+b} m = g^{-s(a+b)} g^{s(a+b)} m = m \quad \text{und} \quad (g^c)^{-s} h^c \sigma = g^{-sc} g^{(sc)} \sigma = \sigma.$$

[4] Eine Wiederverschlüsselung ist eine erneute Verschlüsselung eines Geheimtextes unter dem gleichen öffentlichen Schlüssel, aber mit einer anderen Zufallszahl. Es entsteht aufgrund der Homomorphie ein Geheimtext unter dem verwendeten öffentlichen Schlüssel mit der Summe der Zufallszahlen. Dazu muss der Geheimtext nicht entschlüsselt werden. Die Wiederverschlüsselung kann also insbesondere auch ohne Kenntnis des geheimen Schlüssels berechnet werden.

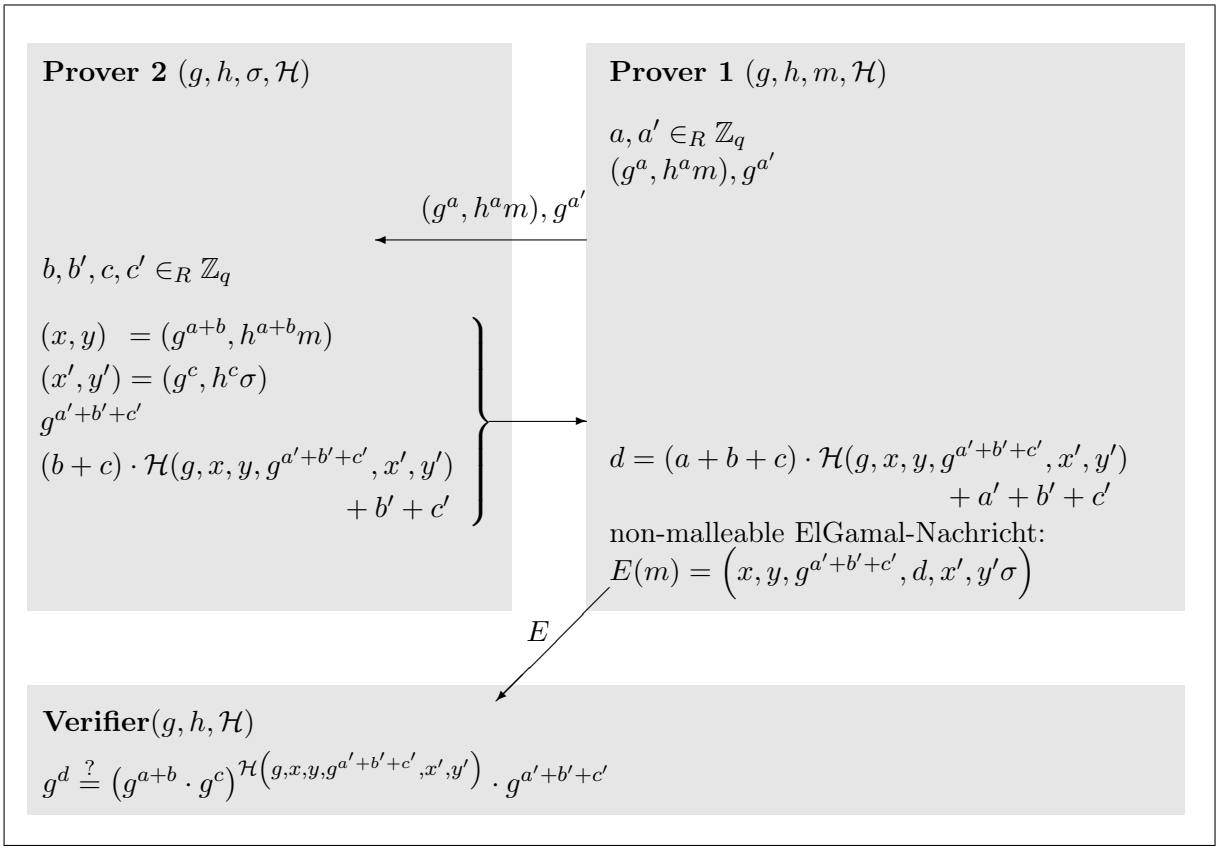

Abbildung 3.3: NIZK-Beweis der non-malleability für zwei Prover und zwei Verschlüsselungen.

Satz 3.3
Das vorgestellte erweiterte Verschlüsselungsschema ist non-malleable unter einem adaptiven Chosen-Ciphertext-Angriff im Random-Oracle-Modell.

Beweis:
Mit $u := a+b+c$ und $u' := a'+b'+c'$ entspricht die Verschlüsselung, Entschlüsselung und Verifikation der aus Abschnitt 2.3. O.B.d.A. lässt sich die Rolle von Nachricht m und σ auch vertauschen, so dass der nicht-interaktive Zero-Knowledge-Beweis sowohl die Kenntnis der Zufallszahl $a + b$, als auch der Zufallszahl c erfordert. Die Prover können also gemeinsam den von ihnen erzeugten Geheimtext entschlüsseln. Diese Entschlüsselung erstellen sie in diesem nicht-interaktiven Fall mittels einer Oracle-Replay-Technik (vgl. [PS96]). Nach Satz 2.3 ist das Verschlüsselungsschema somit non-malleable.

□

3.5 Designated-Verifier-Beweise

In vielen Beweisen ist es von Bedeutung, dass nur ein bestimmter Verifier zu der Überzeugung gelangen kann, dass der Beweis korrekt ist. Bei elektronischen Wahlen kann es beispielsweise entscheidend sein, dass nur der Wähler selbst weiß, welche vom Observer verschlüsselte Wahloption sich hinter einem Geheimtext verbirgt.

Zur Lösung setzt man sogenannte *Designated-Verifier-Beweise* ein. Ein Verifier in einem Beweis heißt *Designated-Verifier*, wenn niemand außer diesem Teilnehmer durch den Beweis überzeugt werden kann. Dahinter steht der folgende Gedanke. Der Prover A beweist dem Verifier B - anstelle der zu zeigenden Aussage θ - die Aussage „θ ist wahr oder ich bin B".

Daher ist die Designated-Verifier-Eigenschaft eines Beweises auch dann noch erfüllt, wenn der Designated-Verifier geheime Informationen an Dritte weitergibt. Die folgende Definition der Designated-Verifier-Eigenschaft basiert auf der Definition in [JSI96].

Definition 3.6 (Designated-Verifier-Eigenschaft)
Gegeben seien drei Parteien, ein Prover A, ein Designated-Verifier B und eine dritte Partei C. Es sei $(\text{Prot}_A, \text{Prot}_B)$ ein Beweis, in dem A dem Verifier B eine Aussage θ beweist.

Ein Beweis $(\text{Prot}_A, \text{Prot}_B)$ für eine Aussage θ besitzt die *Designated-Verifier-Eigenschaft*, wenn für jedes Protokoll $(\text{Prot}_A, \text{Prot}'_B, \text{Prot}_C)$, in dem B unter Verwendung von $(\text{Prot}_A, \text{Prot}_B)$ gegenüber C die Aussage θ „beweist", ein anderes Protokoll $(\text{Prot}''_B, \text{Prot}_C)$ existiert, so dass B die Berechnungen von Prot''_B effizient selbst durchführen und C nicht zwischen den Protokollansichten $(\text{Prot}_A, \text{Prot}'_B, \text{Prot}_C)$ und $(\text{Prot}''_B, \text{Prot}_C)$ unterscheiden kann.

3.5.1 Designated-Verifier-Wiederverschlüsselungsbeweis

Definition 3.7 (Witness)
Ein *witness* (*Zeuge*) einer Behauptung ist eine Information, die verwendet werden kann, um die Behauptung zu beweisen.

Beispielsweise ist der Diskrete Logarithmus a Zeuge für die Kenntnis des Diskreten Logarithmus von g^a.

Ein effizientes Protokoll, das für gegebene Chiffretexte e und e' und einem Zeugen, dass e' eine Wiederverschlüsselung von e darstellt, die Existenz eines solchen Zeugen so beweist, dass nur der Designated-Verifier seine Korrektheit überprüfen kann, wird *Designated-Verifier-Wiederverschlüsselungsbeweis* genannt.

3.5.2 Ein Designated-Verifier-Wiederverschlüsselungsbeweis basierend auf einer ElGamal-Verschlüsselung

Gegeben sei eine multiplikative Gruppe G mit Primzahlordnung $q = |G|$. Der Wert g sei ein erzeugendes Element von G und $z_V \in \mathbb{Z}_q$.

In diesem Abschnitt wird ein Designated-Verifier-Wiederverschlüsselungsbeweis beschrieben, der auf der Technik aus [JSI96] basiert. Anschließend werden die Eigenschaften des Protokolls analysiert.

Der Prover kann geheim beweisen, dass (x', y') eine Wiederverschlüsselung von (x, y) mit ξ als Zeugen ist, d. h. $(x', y') = (g^\xi x, h^\xi y)$ gilt. Der geheime Schlüssel des Verifiers wird mit z_V bezeichnet, und der dazugehörige öffentliche Schlüssel ist durch $h_V = g^{z_V}$ gegeben. Das Protokoll setzt voraus, dass der Verifier seinen geheimen Schlüssel kennt. Wenn diese Eigenschaft

nicht durch die zugrundeliegende Public-Key-Infrastruktur sichergestellt wird, muss ein Protokoll herangezogen werden, das diese Eigenschaft garantiert (siehe Beispiel 3.1).

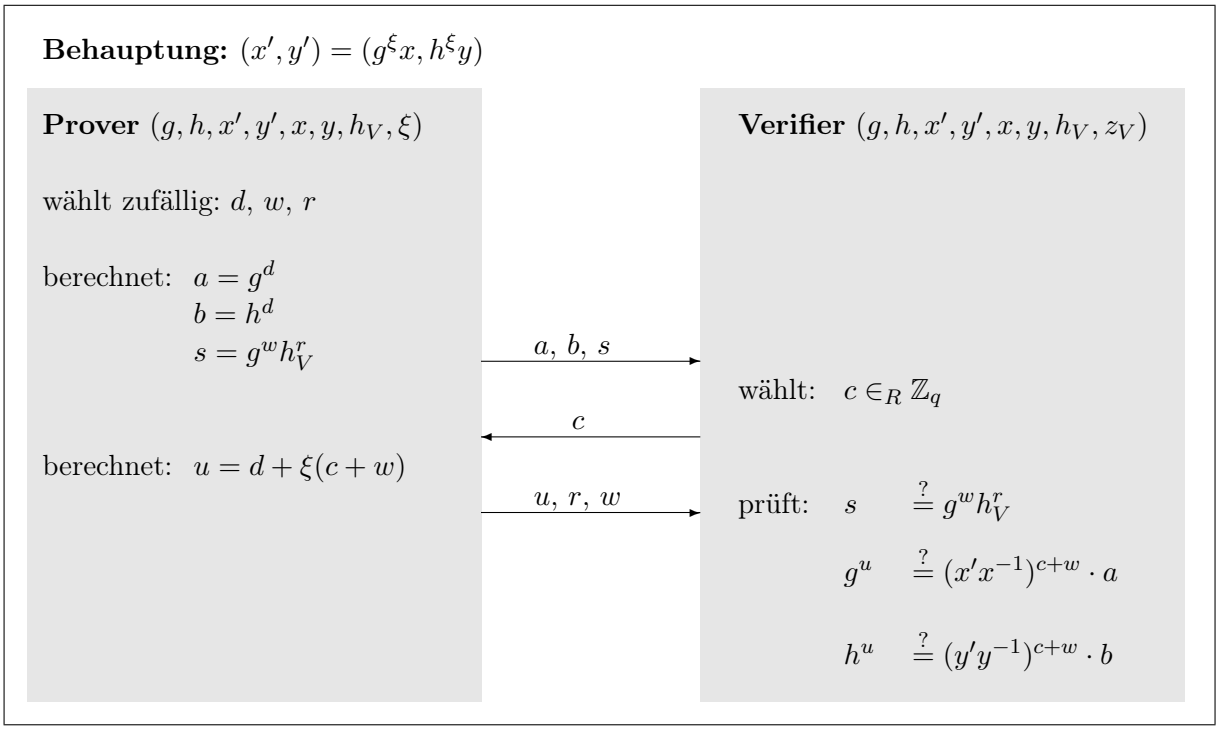

Abbildung 3.4: 3-Runden Designated-Verifier-Wiederverschlüsselungsbeweis.

1. Der Prover wählt d, w und r zufällig und berechnet

   $$a = g^d,\ b = h^d \text{ und } s = g^w h_V^r.$$

   Er sendet diese Werte zum Verifier. Die Werte legen den Prover auf d, w, bzw. r fest. Jedoch ist s eine Festlegung für $w + z_V r$, und der Verifier kann sein Wissen von z_V benutzen, um beliebige Werte w' und r' zu wählen, die die Gleichung $w' + z_V r' = w + z_V r$ erfüllen.

2. Der Verifier wählt eine Zufallszahl (Challenge) $c \in_R \mathbb{Z}_q$ und übermittelt sie an den Prover.

3. Der Prover berechnet $u = d + \xi(c + w)$ und sendet w, r, u zum Verifier.

4. Der Verifier überprüft, ob die folgenden Gleichungen gelten:

   $$\begin{aligned} s &\stackrel{?}{=} g^w h_V^r \\ g^u &\stackrel{?}{=} \left(x'x^{-1}\right)^{c+w} \cdot a \\ h^u &\stackrel{?}{=} \left(y'y^{-1}\right)^{c+w} \cdot b \end{aligned}$$

Satz 3.4
Dieser interaktive Beweis der Wiederverschlüsselung von (x, y) ist durchführbar, korrekt und besitzt die Designated-Verifier-Eigenschaft.

Beweis:

- Der interaktive Beweis ist *durchführbar*. Wenn sich alle Teilnehmer korrekt verhalten, und der Prover (x', y') wirklich als Wiederverschlüsselung von (x, y) erhalten hat, kann er immer eine korrekte Antwort u auf die Challenge c geben. Es gilt dann $s = g^w h_V^r$.
  Der Prover berechnet die Antwort u auf die Frage c durch $u = d + \xi(c + w)$, so dass sowohl

  $$g^u = g^{d+\xi(c+w)} = g^d \cdot \left(g^\xi x \cdot x^{-1}\right)^{c+w} = a \cdot \left(x' \cdot x^{-1}\right)^{c+w}$$

  als auch

  $$h^u = h^{d+\xi(c+w)} = h^d \cdot \left(h^\xi y \cdot y^{-1}\right)^{c+w} = b \cdot \left(y' \cdot y^{-1}\right)^{c+w}$$

  gilt.

- Das Protokoll ist *korrekt*. Ist $g^\xi x \neq x'$ oder $g^\xi y \neq y'$, so kann der Prover die Antwort u nur für den passenden Wert c berechnen. Um erfolgreich zu betrügen, muss er c vorher erraten, Werte $u, w \in_R \mathbb{Z}_q$ wählen und dann im ersten Schritt des Beweises die Werte $a := \left(x \cdot (x')^{-1}\right)^{c+w} g^u$, $b := \left(y \cdot (y')^{-1}\right)^{c+w} h^u$ und $s = g^w h_V^r$ zum Verifier senden.

  In jedem Fall kann der Prover nur auf einen der q möglichen Werte für c eine korrekte Antwort u geben. Angenommen, er könnte für zwei verschiedene Werte c und c' einen korrekten Wert u als Antwort geben, so würde gelten:

  $$\left.\begin{array}{l} g^u = (x \cdot (x')^{-1})^{c+w} \\ g^u = (x \cdot (x')^{-1})^{c'+w} \end{array}\right\} \Rightarrow (x \cdot (x')^{-1})^{c+w} = (x \cdot (x')^{-1})^{c'+w} \Rightarrow 1 = (x \cdot (x')^{-1})^{c'-c}$$

  $$\underset{\text{da } x \neq x'}{\Rightarrow} 0 = c' - c \Rightarrow c' = c,$$

  $$\left.\begin{array}{l} h^u = (y \cdot (y')^{-1})^{c+w} \\ h^u = \left(y \cdot (y')^{-1}\right)^{c'+w} \end{array}\right\} \Rightarrow (y \cdot (y')^{-1})^{c+w} = (y \cdot (y')^{-1})^{c'+w} \Rightarrow 1 = (y \cdot (y')^{-1})^{c'-c}$$

  $$\underset{\text{da } y \neq y'}{\Rightarrow} 0 = c' - c \Rightarrow c' = c.$$

  Dann wären die Werte c und c' gleich.

  Der Prover kann also mit einer Wahrscheinlichkeit von $\frac{1}{q}$ betrügen. Die Wahrscheinlichkeit, dass sein Betrugsversuch aufgedeckt wird, ist dann $1 - q^{-1}$.

- Der Beweis erfüllt die *Designated-Verifier-Eigenschaft*. Der Verifier, der den geheimen Schlüssel z_V kennt, für den $g^{z_V} = h_V$ gilt, kann den obigen Beweis für beliebige (x, y) und $(\tilde{x}, \tilde{y})$ generieren. Der Verifier muss den Beweis nicht nur wie ein Simulator zum Nachweis der Zero-Knowledge-Eigenschaft simulieren können. Er kann gegenüber Dritten nicht einfach Szenen löschen. Daher wird der Beweis auch bezüglich des geheimen Schlüssels des Designated-Verifier durchgeführt.

  Entscheidend ist, dass der Wert s den Designated-Verifier *nicht* auf w und r festlegt. Er wählt zufällige Werte α, β und $\tilde{u}$ und berechnet dann $\tilde{s} = g^\beta$, $\tilde{a} = g^{\tilde{u}} \left(\tilde{x} x^{-1}\right)^{-\alpha}$ und $\tilde{b} = h^{\tilde{u}} \left(\tilde{y} y^{-1}\right)^{-\alpha}$. Diese Werte sendet er an den Dritten, dem er den Beweis übertragen möchte, und er erhält $\tilde{c} \in_R \mathbb{Z}_q$ von ihm zurück. Daraufhin berechnet er $\tilde{w} = \alpha - \tilde{c} \bmod q$

und $\tilde{r} = (\beta - \tilde{w})z_V^{-1} \bmod q$ und setzt $(\tilde{w}, \tilde{r}, \tilde{u})$ als Beweis. Dieser Beweis ist vom Dritten ununterscheidbar zur ursprünglichen Protokollansicht, denn er besteht die Verifikation:

$$\begin{aligned} \tilde{s} &= g^{\beta} = g^{\alpha-\tilde{c}} g^{\beta-(\alpha-\tilde{c})} = g^{\alpha-\tilde{c}} g^{z_V(\beta-\tilde{w})z_V^{-1}} = g^{\tilde{w}} h_V^{\tilde{r}}, \\ g^{\tilde{u}} &= \left(\tilde{x}x^{-1}\right)^{\alpha} \cdot g^{\tilde{u}} \left(\tilde{x}x^{-1}\right)^{-\alpha} = \left(\tilde{x}x^{-1}\right)^{\tilde{c}+\alpha-\tilde{c}} \cdot g^{\tilde{u}} \left(\tilde{x}x^{-1}\right)^{-\alpha} = \left(\tilde{x}x^{-1}\right)^{\tilde{c}+\tilde{w}} \cdot \tilde{a}, \\ h^{\tilde{u}} &= \left(\tilde{y}y^{-1}\right)^{\alpha} \cdot h^{\tilde{u}} \left(\tilde{y}y^{-1}\right)^{-\alpha} = \left(\tilde{y}y^{-1}\right)^{\tilde{c}+\alpha-\tilde{c}} \cdot h^{\tilde{u}} \left(\tilde{y}y^{-1}\right)^{-\alpha} = \left(\tilde{y}y^{-1}\right)^{\tilde{c}+\tilde{w}} \cdot \tilde{b}. \end{aligned}$$

Der Designated-Verifier kann also Dritten gegenüber für beliebige $(\tilde{x}, \tilde{y})$ „beweisen", dass dies eine Wiederverschlüsselung von (x, y) ist.

□

3.5.3 Nicht-interaktiver Designated-Verifier-Wiederverschlüsselungsbeweis

In diesem Abschnitt wird ein Designated-Verifier-Wiederverschlüsselungbeweis vorgestellt, der auf dem in Abschnitt 3.5.2 beschriebenen basiert und unter Benutzung der Fiat-Shamir-Heuristik (siehe Abschnitt 3.3 und [FS86]) nun nicht-interaktiv durchgeführt werden kann.
Der Beweis kann dann vom Prover ohne Interaktion des Verifiers erstellt werden:

1. Der Prover berechnet a, b und s wie im interaktiven Beweis.
2. Dann berechnet der Prover die Zufallszahl (Challenge) $c = \mathcal{H}(E||a||b||s)$, wobei $a||b||s$ die Aneinanderkettung von a, b und s darstellt, $E = (x||y||x'||y')$ die sogenannte *Umgebung* (*environment*) und $\mathcal{H}$ eine kryptografische Hashfunktion ist.
3. Für diese Frage c berechnet der Prover u. Der Vektor (x, w, r, u) ist der Beweis.[5]
4. Der Verifier testet, ob $c \stackrel{?}{=} \mathcal{H}\left(E \left|\left| g^u(x'x^{-1})^{-(c+w)} \right|\right| h^u(y'y^{-1})^{-(c+w)} \left|\right| g^w h_V^r\right)$ gilt.

Satz 3.5
Der nicht-interaktive Beweis ist durchführbar, korrekt und besitzt die Designated-Verifier-Eigenschaft.

Beweis:

- Der Beweis ist *durchführbar*. Wenn sich alle Teilnehmer korrekt verhalten, und der Prover (x', y') wirklich als Wiederverschlüsselung von (x, y) erhalten hat, kann er immer eine korrekte Antwort u auf die Challenge c geben. Der Prover berechnet wie im interaktiven Beweis (Satz 3.4) die Antwort $u = d + \xi(c + w)$, so dass gilt:

$$\begin{aligned} s &= g^w h_V^r, \\ a &= g^u \left(x'x^{-1}\right)^{-c-w}, \\ b &= h^u \left(y'y^{-1}\right)^{-c-w}. \end{aligned}$$

  Daher ist $c = \mathcal{H}(E||a||b||s)$, wie es vom Prover angegeben wurde.

[5] Diese Konstruktion ist effizienter als diejenige, die in [JSI96] beschrieben wird. Dort benötigt man ein 5-Tupel als Beweis.

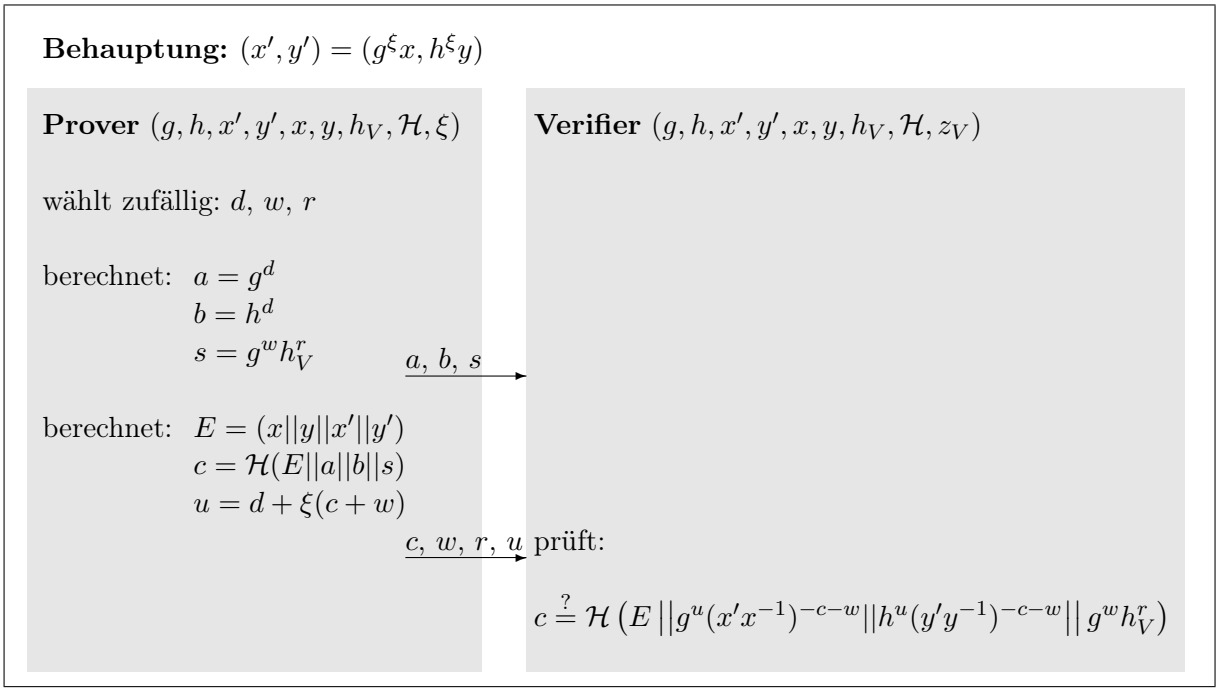

Abbildung 3.5: Nicht-interaktiver Designated-Verifier-Wiederverschlüsselungsbeweis.

- Das Protokoll ist *korrekt*. Ist $g^\xi x \neq x'$ oder $g^\xi y \neq y'$, so kann der Prover die Antwort u nur für einen passenden der q möglichen Werte von c berechnen. Die Korrektheit dieses Beweises ergibt sich unmittelbar aus dem Beweis der Korrektheit zu Satz 3.4. Da der Prover den Wert von c durch die Einweg-Hashfunktion nicht vor der Wahl der Werte d, w und r bestimmen kann, besteht nur eine Wahrscheinlichkeit von q^{-1}, dass er c vorher rät.

- Der Beweis erfüllt die *Designated-Verifier-Eigenschaft*. Kennt der Verifier den geheimen Schlüssel z_V, für den $g^{z_V} = h_V$ gilt, so kann er den obigen Beweis für beliebige (x, y) und $(\tilde{x}, \tilde{y})$ generieren.

  Der Wert s legt den Verifier *nicht* auf w und r fest. Der Verifier wählt zufällige Werte α, β und $\tilde{u}$ und berechnet (mit $E = (x||y||\tilde{x}||\tilde{y})$):

$$\begin{aligned}
\tilde{a} &= g^{\tilde{u}} \left(\tilde{x}x^{-1}\right)^{-\alpha}, \\
\tilde{b} &= h^{\tilde{u}} \left(\tilde{y}y^{-1}\right)^{-\alpha}, \\
\tilde{s} &= g^{\beta}, \\
\tilde{c} &= \mathcal{H}\left(E||\tilde{a}||\tilde{b}||\tilde{s}\right), \\
\tilde{w} &= \alpha - \tilde{c} \bmod q, \\
\tilde{r} &= (\beta - \tilde{w}) z_V^{-1} \bmod q.
\end{aligned}$$

  Dann setzt er $(\tilde{c}, \tilde{w}, \tilde{r}, \tilde{u})$ als Beweis. Der Beweis besteht die Verifikation, denn es gilt:

$$\begin{aligned}
\tilde{s} &= g^{\alpha-\tilde{c}} g^{\beta-(\alpha-\tilde{c})} = g^{\beta} = g^{\alpha-\tilde{c}} g^{z_V(\beta-\tilde{w})z_V^{-1}} = g^{\tilde{w}} h_V^{\tilde{r}}, \\
\tilde{a} &= g^{\tilde{u}} \left(\tilde{x}x^{-1}\right)^{-\alpha} = g^{\tilde{u}} \left(\tilde{x}x^{-1}\right)^{-\tilde{c}-\tilde{w}}, \\
\tilde{b} &= h^{\tilde{u}} \left(\tilde{y}y^{-1}\right)^{-\alpha} = h^{\tilde{u}} \left(\tilde{y}y^{-1}\right)^{-\tilde{c}-\tilde{w}},
\end{aligned}$$

so dass (mit $E = (x||y||\tilde{x}||\tilde{y})$) gilt:

$$\tilde{c} = \mathcal{H}(E||g^{\tilde{u}} \left(\tilde{x}x^{-1}\right)^{-\tilde{c}-\tilde{w}} ||h^{\tilde{u}} \left(\tilde{y}y^{-1}\right)^{-\tilde{c}-\tilde{w}} ||g^{\tilde{w}}h_V^{\tilde{r}}).$$

Der Verifier kann also für beliebige $(\tilde{x}, \tilde{y})$ „beweisen", dass dies eine Wiederverschlüsselung von (x, y) ist.

□

3.6 Witness-Indistinguishable-Beweise

Führt man Zero-Knowledge-Protokolle parallel aus, senden Prover und Verifier also gleich mehrere Nachrichten in einer Runde, bzw. wählt der Verifier die Challenge nicht als einzelnes Bit, sondern als String, so bleibt die Zero-Knowledge-Eigenschaft im Allgemeinen nicht erhalten.

Darüber hinaus gibt es Protokolle, die schon in der nicht-parallelen Ausführung keine effiziente Simulierbarkeit besitzen, die aber dennoch gewisse Sicherheitsanforderungen beweisbar erfüllen sollen.

Alternativ zu den Zero-Knowledge-Beweisen wurde daher das Konzept des *Witness-Hiding*-Protokolls von Feige und Shamir eingeführt [FS90].

Definition 3.8 (wesentlich verschieden)
Gegeben seien zwei Zeugen ξ, ξ' einer Behauptung unter dem Sicherheitsparameter k. Die Zeugen nennt man *wesentlich verschieden*, wenn ein effizienter Algorithmus B und eine vernachlässigbare Funktion ν für alle Angreifer A existiert, so dass gilt:

$$P[A(k, \xi) = \xi'] \leq P[B(k) = \xi'] + \nu(k).$$

Zwei Zeugen einer Behauptung heißen also wesentlich verschieden, wenn es mit Kenntnis einer der beiden Zeugen genauso schwer ist, den anderen zu berechnen, wie ohne Kenntnis eines Zeugen.

Definition 3.9 (witness-indistinguishable)
Ein Beweissystem (Prover,Verifier) heißt *witness-indistinguishable* (kurz *WI*), wenn der Verifier für alle Paare von wesentlich verschiedenen Zeugen ξ_1, ξ_2 für die zu beweisende Behauptung die Protokolldurchführung, bei der der Prover ξ_1 verwendet hat, nicht von der Protokolldurchführung, bei der der Prover ξ_2 verwendet hat, unterscheiden kann.

Somit hat ein Beweissystem die Witness-Indistinguishable-Eigenschaft, wenn für Prover die Protokollansichten bei der Verwendung zweier wesentlich verschiedener Zeugen ununterscheidbar sind.

3.6.1 Interaktiver 3-Runden Witness-Indistinguishable-Beweis der 1-von-n_L-ElGamal-Wiederverschlüsselung für zwei Prover

Mit dem folgenden neuen Witness-Indistinguishable-Beweisprotokoll können zwei Prover durch eine Interaktion mit dem Verifier über drei Runden beweisen, dass sie gemeinsam für eine

ElGamal-verschlüsselte Nachricht $(\tilde{x}_i, \tilde{y}_i)$ eine Wiederverschlüsselung (x'_k, y'_k) in der Liste der n_L ElGamal-verschlüsselten Nachrichten $(x'_1, y'_1) \dots (x'_{n_L}, y'_{n_L})$ erstellt haben, ohne diese konkret anzugeben, d. h. ohne k zu bestimmen[6]. Das Protokoll basiert auf den Verfahren, die in [CDS94], [CFSY96] und [CGS97] vorgestellt wurden und stellt eine Erweiterung auf n_L Geheimtexte und zwei hintereinander durchgeführte Wiederverschlüsselungen durch zwei verschiedene Instanzen (zwei Prover) dar. Es sei (x'_k, y'_k) eine Wiederverschlüsselung von $(\tilde{x}_i, \tilde{y}_i)$, und die zur Wiederverschlüsselung benutzte Zufälligkeit (der Zeuge) sei ξ, d. h. es gelte $(x'_k, y'_k) = (g^\xi \tilde{x}_i, h^\xi \tilde{y}_i)$. Hierbei besteht die Wiederverschlüsselung eigentlich aus zwei nacheinander erfolgten Wiederverschlüsselungen mit Zeugen a und b, d. h. $\xi = a + b$ und $(x'_k, y'_k) = (g^b g^a \tilde{x}_i, h^b h^a \tilde{y}_i)$. Der Zeuge a ist Prover 1 bekannt, während Prover 2 den Zeugen b kennt[7].

1. (a) Prover 2 wählt einen Wert $w \in_R \mathbb{Z}_q$, berechnet $u = g^w$ und $t = h^w$ und sendet u und t an Prover 1.

   (b) Prover 1 wählt ein $v \in_R \mathbb{Z}_q$ sowie die Werte $d_\ell, r_\ell \in_R \mathbb{Z}_q$ für $1 \leq \ell \leq n_L$, $\ell \neq k$. Anschließend berechnet er

$$\alpha_\ell = \left(x'_\ell \tilde{x}_i^{-1}\right)^{d_\ell} g^{r_\ell} \text{ für } \ell \neq k,$$

$$\beta_\ell = \left(y'_\ell \tilde{y}_i^{-1}\right)^{d_\ell} h^{r_\ell} \text{ für } \ell \neq k,$$

$$\alpha_k = u \cdot g^v,$$

$$\beta_k = t \cdot h^v.$$

   Dann schickt er die Werte α_ℓ, β_ℓ für $1 \leq \ell \leq n_L$ an den Verifier.

   Man beachte, dass sie die Prover auf d_ℓ und r_ℓ für alle $\ell = 1, \dots, n_L$, außer für $\ell = k$, festlegen. α_k und β_k legen die Prover nur auf einen Wert $\xi d_k + r_k$ fest, da $\alpha_k = g^{\xi d_k + r_k}$ und $b = h^{\xi d_k + r_k}$ gilt. Das bedeutet, dass die Prover immer noch d_k und r_k nach dieser Runde geeignet wählen können.

2. Der Verifier wählt eine Challenge $c \in_R \mathbb{Z}_q$ und sendet diese an Prover 1.

3. (a) Prover 1 setzt $d_k = c - \sum_{\substack{\ell=1 \\ \ell \neq k}}^{n_L} d_\ell$ und sendet d_k an Prover 2.

   (b) Prover 2, der seinen Teil b des Zeugens ξ kennt, berechnet $r = w - b d_k$ und übermittelt r an Prover 1.

   (c) Nun kann Prover 1 mit Hilfe der Kenntnis von a den Wert $r_k = v - a d_k + r$ berechnen. Er sendet die Werte d_ℓ, r_ℓ für $1 \leq \ell \leq n_L$ an den Verifier.

[6] Man kann für die Verwendung in einem Wahlprotokoll o.B.d.A. annehmen, dass für jede gültige Wahloption $m_i \in \mathbf{L}$ eine deterministische Standardverschlüsselung $(\tilde{x}_i, \tilde{y}_i)$ existiert, so dass deutlich ist, welches m_i zu einem gegebenen Chiffretext $(\tilde{x}_i, \tilde{y}_i)$ gehört. Die auf diese Weise deterministisch verschlüsselten Stimmen können beispielsweise entschlüsselt werden, indem man alle gültigen Stimmen aus $\mathbf{L}$ durchprobiert.

[7] Im Rahmen eines Wahlprotokolls mit Observer (vgl. Kapitel 6) ist Prover 1 der Wähler, während Prover 2 dem Observer entspricht.

Der Verifier prüft, ob die folgenden Gleichungen erfüllt sind:

$$c \stackrel{?}{=} \sum_{\ell=1}^{n_L} d_\ell,$$

$$\alpha_\ell \stackrel{?}{=} \left(x'_\ell \tilde{x}_i^{-1}\right)^{d_\ell} g^{r_\ell} \text{ für } 1 \le \ell \le n_L,$$

$$\beta_\ell \stackrel{?}{=} \left(y'_\ell \tilde{y}_i^{-1}\right)^{d_\ell} h^{r_\ell} \text{ für } 1 \le \ell \le n_L.$$

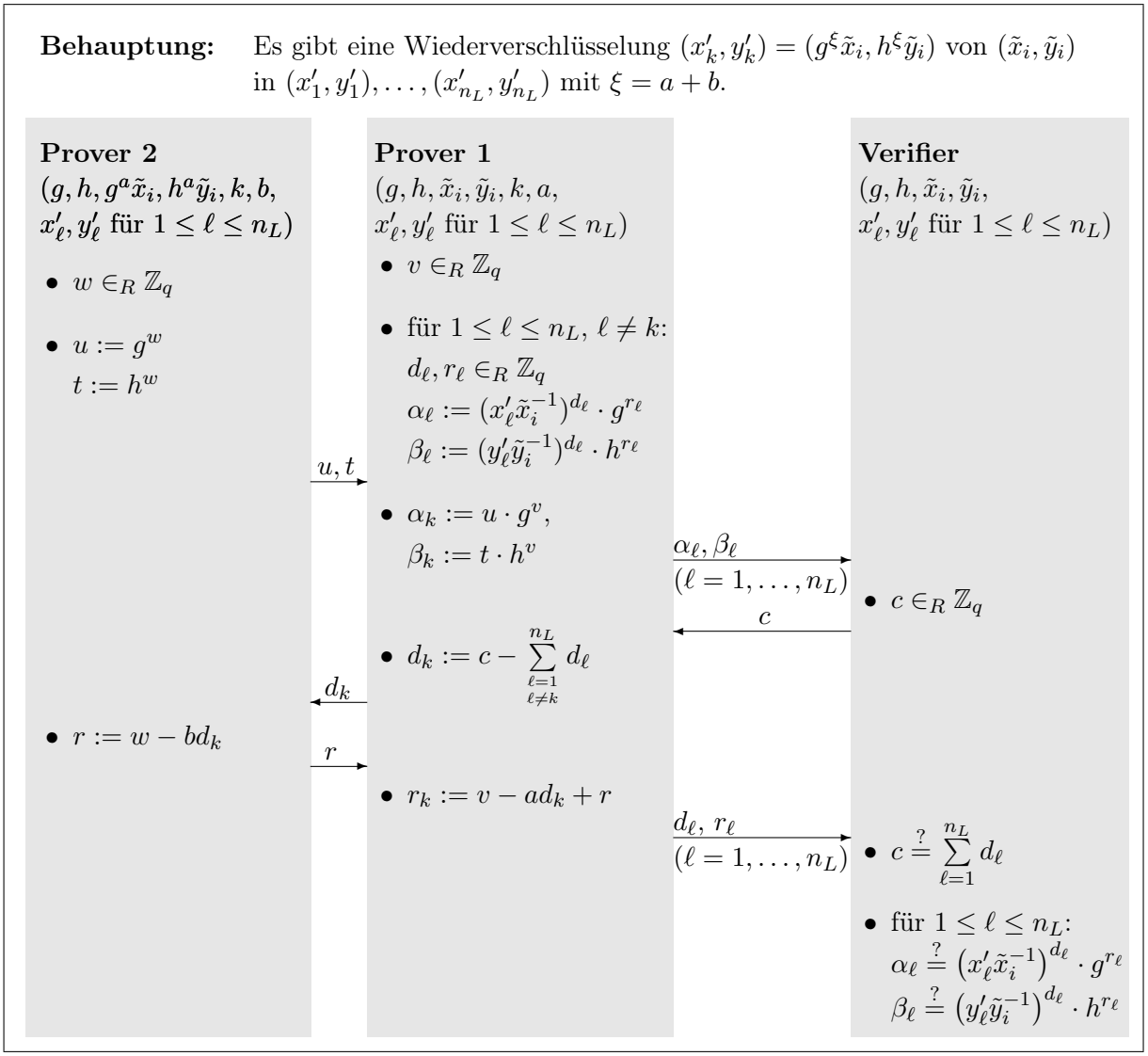

Abbildung 3.6: Interaktiver 3-Runden 2-Prover WI-Wiederverschlüsselungsbeweis.

Satz 3.6

Der interaktive Beweis ist durchführbar, korrekt und witness-indistinguishable.

Beweis:

- Zunächst wird hier die *Durchführbarkeit* des interaktiven Beweises gezeigt.
  Wenn sich alle Teilnehmer korrekt verhalten, und $(g^{\xi}\tilde{x}_i, h^{\xi}\tilde{y}_i) \in \{(x'_1, y'_1), \ldots, (x'_{n_L}, y_{n_L})\}$ gilt, können die Prover immer eine korrekte Antwort auf die Challenge c des Verifiers geben. Prover 1 wählt den Wert d_k gerade so, dass $c = \sum_{\ell=1}^{n_L} d_\ell$ gilt. Die Werte α_ℓ und β_ℓ sind für $\ell \neq k$ ebenfalls von Prover 1 so gewählt, dass sie den vom Verifier zu überprüfenden Gleichungen entsprechen.
  Da des Weiteren

$$\begin{aligned}\alpha_k &= u \cdot g^v = g^w \cdot g^v = g^{w+v} = g^{(a+b)d_k-(a-b)d_k+w+v} \\ &= g^{\xi d_k+(v-ad_k+(w-bd_k))} = g^{\xi d_k+r_k} = \left(x'_k\tilde{x}_i^{-1}\right)^{d_k} g^{r_k}\end{aligned}$$

  und

$$\begin{aligned}\beta_k &= t \cdot h^v = h^w \cdot h^v = h^{w+v} = h^{(a+b)d_k-(a-b)d_k+w+v} \\ &= h^{\xi d_k+(v-ad_k+(w-bd_k))} = h^{\xi d_k+r_k} = \left(y'_k\tilde{y}_i^{-1}\right)^{d_k} h^{r_k}\end{aligned}$$

  gilt, bestehen auch α_k und β_k die Überprüfung durch den Verifier.

- Um zu zeigen, dass der Beweis *korrekt* ist, muss man zeigen, dass die Prover keine Möglichkeit haben, auf eine Challenge c des Verifiers korrekt zu antworten, wenn es keine Wiederverschlüsselung in der betrachteten Liste der Geheimtexte gibt.
  Um erfolgreich zu betrügen, müssten die Prover die Challenge vorher erraten und dann zufällige Werte $d_1, \ldots, d_{n_L}$ und $r_1, \ldots, r_{n_L}$ so wählen, dass die Gleichungen des Verifiers erfüllt sind. Die Prover können aber nur auf einen der q möglichen Werte für c eine korrekte Antwort geben.

  Angenommen, die Prover könnten auf zwei verschiedene Werte c und c' korrekt antworten, dann müssten sie für mindestens ein $j \in \{1, \ldots, n_L\}$ die Werte d_j und r_j so verändern, dass für die angepassten Werte $d'_j \neq d_j$ und r'_j gilt:

$$c' = d_j + \sum_{\substack{\ell=1 \\ \ell \neq j}}^{n_L} d_\ell.$$

  Außerdem müsste dann gelten:

$$\alpha_j = \left(x'_j\tilde{x}_i^{-1}\right)^{d_j} \cdot g^{r_j} \stackrel{!}{=} \left(x'_j\tilde{x}_i^{-1}\right)^{d'_j} \cdot g^{r'_j}, \tag{3.1}$$

$$\beta_j = \left(y'_j\tilde{y}_i^{-1}\right)^{d_j} \cdot h^{r_j} \stackrel{!}{=} \left(y'_j\tilde{y}_i^{-1}\right)^{d'_j} \cdot h^{r'_j}. \tag{3.2}$$

  Da $x'_j\tilde{x}_i^{-1} = g^{\xi_j}$ für ein ξ_j, und $y'_j\tilde{y}_i^{-1} = h^{\tilde{\xi}_j}$ für ein $\tilde{\xi}_j$ gilt, folgt aus den Gleichungen (3.1) und (3.2)

$$\begin{aligned}g^{\xi_j \cdot d_j + r_j} &= g^{\xi_j \cdot d'_j + r'_j}, \\ h^{\tilde{\xi}_j \cdot d_j + r_j} &= h^{\tilde{\xi}_j \cdot d'_j + r'_j}.\end{aligned}$$

Also müssen die beiden folgenden Gleichungen erfüllt sein:

$$\xi_j \cdot d_j + r_j = \xi_j \cdot d'_j + r'_j, \tag{3.3}$$
$$\tilde{\xi}_j \cdot d_j + r_j = \tilde{\xi}_j \cdot d'_j + r'_j. \tag{3.4}$$

Stellt man nun Gleichung (3.3) nach ξ_j und Gleichung (3.4) nach $\tilde{\xi}_j$ um, so erkennt man, dass die beiden Werte ξ_j und $\tilde{\xi}_j$ gleich sein müssen:

$$\xi_j = (r'_j - r_j)(d_j - d'_j)^{-1} = \tilde{\xi}_j.$$

Also gilt:

$$x'_j = \tilde{x}_i g^{\xi_j},$$
$$y'_j = \tilde{y}_i h^{\xi_j}.$$

Das bedeutet, dass (x'_j, y'_j) eine Verschlüsselung von $(\tilde{x}_i, \tilde{y}_i)$ unter dem Zeugen ξ_j ist. Dies ist ein Widerspruch zur Annahme, dass es keine Wiederverschlüsselung in der betrachteten Liste gibt.

Also kann der Prover mit einer Wahrscheinlichkeit von $\frac{1}{q}$ betrügen. Die Wahrscheinlichkeit, dass sein Betrugsversuch aufgedeckt wird, ist dann $1 - q^{-1}$.

- Es bleibt noch zu zeigen, dass der Beweis die *Witness-Indistinguishable-Eigenschaft* erfüllt.

  Es sei $(x'_{k'}, y'_{k'})$, $(k' \neq k)$, eine weitere Wiederverschlüsselung von $(\tilde{x}_i, \tilde{y}_i)$ unter dem zur Wiederverschlüsselung benutzten, von ξ wesentlich verschiedenen Zeugen $\xi' = a' + b'$, dann gilt

  $$(x'_{k'}, y'_{k'}) = (g^{\xi'} \tilde{x}_i, h^{\xi'} \tilde{y}_i).$$

  Prover 2 verhält sich wie oben beschrieben, wählt $w' \in_R \mathbb{Z}_q$, berechnet $u = g^{w'}$ und $t = g^{w'}$ und sendet diese Werte an Prover 1.
  Dieser wählt die Werte $d_\ell, r_\ell \in_R \mathbb{Z}_q$ (für $1 \leq \ell \leq n_L$ und $\ell \neq k'$) sowie $v' \in_R \mathbb{Z}_q$, berechnet $\alpha_1, \ldots, \alpha_{n_L}$ und $\beta_1, \ldots, \beta_{n_L}$ analog zum oben angegebenen Protokoll und übermittelt die α_i und β_i für $i = 1, \ldots, n_L$ zum Verifier. Damit legt sich Prover 1 auf d_ℓ und r_ℓ für alle $\ell = 1, \ldots, n_L$, außer für k', fest.

  Nachdem Prover 1 vom Verifier die Challenge $c \in_r \mathbb{Z}_q$ erhalten hat, wählt er

  $$d_{k'} = c - \sum_{\substack{\ell=1 \\ \ell \neq k'}}^{n_L} d_\ell$$

  und sendet c und $d_{k'}$ an Prover 2. Dieser antwortet mit $r' = w' - b' d_{k'}$.

  Prover 1 berechnet nun $r_{k'} = v' - a' d_{k'} + r'$ und sendet die Werte d_ℓ und r_ℓ für $1 \leq \ell \leq n_L$ zum Verifier, der die im Protokoll beschriebene Verifikation vornimmt.

Der Verifier kann nicht unterscheiden, welchen Zeugen die Prover verwenden, da

$$\left(x'_{k'}\tilde{x}_i^{-1}\right)^{d_{k'}} g^{r_{k'}} = \left(g^{\xi'}\right)^{d_{k'}} g^{r_{k'}} = g^{\xi'\cdot d_{k'}+r_{k'}} = g^{(a'+b')d_{k'}+v'-a'd_{k'}+w'-b'd_{k'}} = g^{w'+v'} = \alpha_{k'}$$

und

$$\left(y'_{k'}\tilde{y}_i^{-1}\right)^{d_{k'}} h^{r_{k'}} = \left(h^{\xi'}\right)^{h_{k'}} h^{r_{k'}} = h^{\xi'\cdot d_{k'}+r_{k'}} = h^{(a'+b')d_{k'}+v'-a'd_{k'}+w'-b'd_{k'}} = h^{w'+v'} = \beta_{k'}$$

gelten.

□

Das vorgeschlagene Protokoll ist ein interaktiver Witness-Indistinguishable-Beweis in drei Kommunikationsrunden zwischen den Provern und dem Verifier. Unter Benutzung der Fiat-Shamir-Heuristik (siehe Abschnitt 3.3 und [FS86]) kann der Beweis in einen nicht-interaktiven Beweis umgewandelt werden.

3.6.2 Nicht-interaktiver Witness-Indistinguishable-Beweis der 1-von-n_L-ElGamal-Wiederverschlüsselung für zwei Prover

Es sei $\mathcal{H}$ eine kryptografische Hashfunktion. Der Beweis verläuft dann wie folgt.

1. (a) Prover 2 wählt einen Wert $w \in_R \mathbb{Z}_q$, berechnet $u = g^w$ und $t = h^w$ und sendet u und t an Prover 1.

   (b) Prover 1 wählt einen Wert $v \in_R \mathbb{Z}_q$ sowie die Werte $d_\ell, r_\ell \in_R \mathbb{Z}_q$ für $1 \leq \ell \leq n_L$, $\ell \neq k$.
   Anschließend berechnet er die α_ℓ, β_ℓ für $1 \leq \ell \leq n_L$ wie im interaktiven Beweis und schickt sie an den Verifier.

2. (a) Prover 1 berechnet nun die „Challenge“ $c = \mathcal{H}(E||\alpha_1||\ldots||\alpha_{n_L}||\beta_1||\ldots||\beta_{n_L})$ selbst mit Hilfe einer kryptografischen Hashfunktion $\mathcal{H}$, wobei $x||y$ die Aneinanderkettung von x und y, und $(\tilde{x}_i||\tilde{y}_i||x'_1||y'_1||\ldots||x'_{n_L}||y'_{n_L})$ das Environment E ist.

   (b) Nun setzt Prover 1 $d_k = c - \sum_{\substack{\ell=1 \\ \ell\neq k}}^{n_L} d_\ell$ und sendet d_k an Prover 2.

   (c) Prover 2, der seinen Teil b des Zeugens ξ kennt, berechnet $r = w - bd_k$ und schickt r an Prover 1.

   (d) Nun kann Prover 1 mit Hilfe der Kenntnis von a den Wert $r_k = v - ad_k + r$ berechnen. Er schickt die Werte d_ℓ, r_ℓ für $1 \leq \ell \leq n_L$ an den Verifier.

Der Verifier prüft, ob die folgende Gleichung erfüllt ist:

$$\sum_{\ell=1}^{n_L} d_\ell \stackrel{?}{=} \mathcal{H}(E||(x'_1\tilde{x}_i^{-1})^{d_1} g^{r_1}||\ldots||(x'_{n_L}\tilde{x}_i^{-1})^{d_{n_L}} g^{r_{n_L}}||(y'_1\tilde{y}_i^{-1})^{d_1} h^{r_1}||\ldots||(y'_{n_L}\tilde{y}_i^{-1})^{d_{n_L}} h^{r_{n_L}})$$

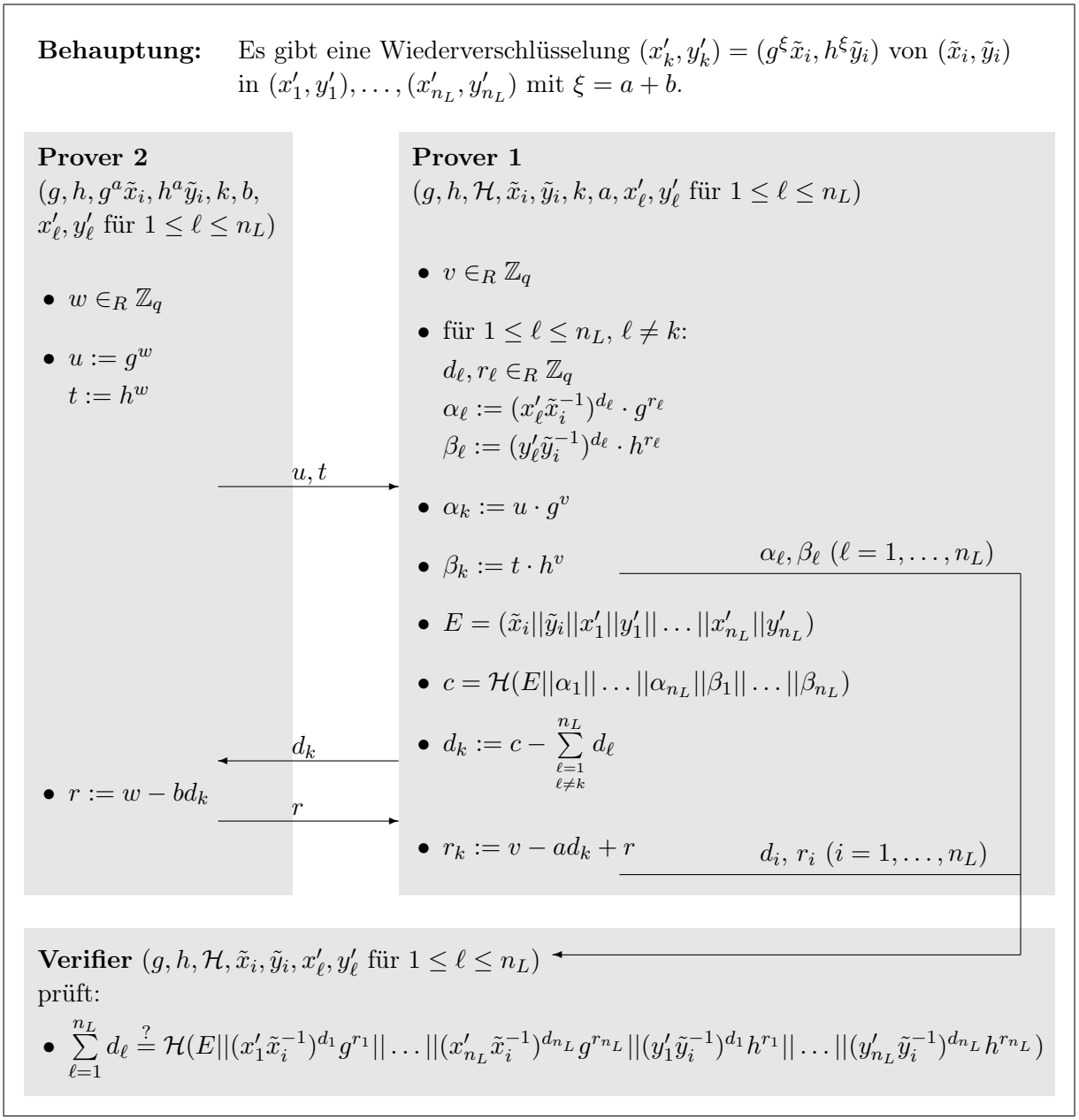

Abbildung 3.7: Nicht-interaktiver 2-Prover WI-Wiederverschlüsselungsbeweis.

Satz 3.7

Der nicht-interaktive Beweis der Existenz einer Wiederverschlüsselung ist durchführbar, korrekt und besitzt die Witness-Indistinguishable-Eigenschaft.

Beweis:

- Der Witness-Indistinguishable-Beweis ist *durchführbar*. Wenn sich alle Teilnehmer korrekt verhalten und wirklich $(g^\xi \tilde{x}_i, h^\xi \tilde{y}_i)$ in $\{(x'_1, y'_1), \ldots, (x'_{n_L}, y'_{n_L})\}$ liegt, kann der Prover immer eine korrekte „Antwort" auf c geben.
  Der Prover wählt d_k und r_k, so dass $c = \sum_{\ell=1}^{n_L} d_\ell$ und $\xi d_k + r_k = w + v$ gilt.

Da bereits die einzelnen Eingaben in die kryptografische Hashfunktion gleich den erforderlichen Werten sind (vgl. Beweis der Durchführbarkeit in Satz 3.6), besteht auch die Ausgabe der Hashfunktion die Überprüfung durch den Verifier.

- Das Protokoll ist *korrekt*. Die Korrektheit des Protokolls ergibt sich aus der interaktiven Form (Satz 3.6) unter Berücksichtigung der kryptografischen Einweg-Hashfunkion. Durch die Fiat-Shamir-Heuristik ist es auch bei Einsatz dieser Hashfunktion einem unehrlichen Prover nur dann möglich korrekt zu antworten, wenn er c vorher rät. Also kann der Prover mit einer Wahrscheinlichkeit von $\frac{1}{q}$ betrügen. Die Wahrscheinlichkeit, dass sein Betrugsversuch aufgedeckt wird, ist dann $1 - q^{-1}$.

- Der Beweis erfüllt die *Witness-Indistinguishable-Eigenschaft*. Wie im interaktiven Beweis (siehe Satz 3.6) kann der Verifier nicht unterscheiden, welchen Zeugen der Prover für den Beweis benutzt. Es ist unwichtig, ob die Challenge Ausgabe einer kryptografischen Hashfunktion ist oder vom Verifier gesendet wird. Da es schon bei den einzelnen Eingaben in die kryptografische Hashfunktion für den Verifier unmöglich ist zu unterscheiden, welchen Zeugen der Prover benutzt, besteht auch die Ausgabe der Hashfunktion die Überprüfung durch den Verifier. Damit besitzt auch die nicht-interaktive Variante die Witness-Indistinguishable-Eigenschaft.

□

3.6.3 Nicht-interaktiver Witness-Indistinguishable-Beweis für zwei Prover der 1-von-n_L-Wiederverschlüsselung im modifizierten-ElGamal-Schema

In diesem nicht-interaktiven Beweis zeigen zwei Prover, dass sie gemeinsam für eine modifiziert-ElGamal-verschlüsselte Nachricht $(\tilde{x}_{1,i}, \tilde{x}_{2,i}, \tilde{y}_i)$ eine Wiederverschlüsselung $(x'_{1,k}, x'_{2,k}, y'_k)$ in der Liste der n_L modifiziert-ElGamal-verschlüsselten Nachrichten $(x'_{1,1}, x'_{2,1}, y'_1) \dots (x'_{1,n_L}, x'_{2,n_L}, y'_{n_L})$ erstellt haben, ohne diese konkret anzugeben, d. h. ohne das k zu bestimmen. Das Protokoll basiert auf dem in Abschnitt 3.6.2 vorgestellten Beweis und stellt eine Erweiterung für Protokolle dar, in denen die modifizierte ElGamal-Verschlüsselung verwendet wird. Es sei $(x'_{1,k}, x'_{2,k}, y'_k)$ eine Wiederverschlüsselung von $(\tilde{x}_{1,i}, \tilde{x}_{2,i}, \tilde{y}_i)$, und die zur Wiederverschlüsselung benutzte Zufälligkeit (der Zeuge) sei ξ, d. h. es gelte $(x'_{1,k}, x'_{2,k}, y'_k) = (g_1^{\xi}\tilde{x}_{1,i}, g_2^{\xi}\tilde{x}_{2,i}, h^{\xi}\tilde{y}_i)$. Wie in den Abschnitten 3.6.1 und 3.6.2 besteht die Wiederverschlüsselung eigentlich aus zwei nacheinander erfolgten Wiederverschlüsselungen mit den Zeugen a und b, d. h. $\xi = a + b$ und

$$(x'_{1,k}, x'_{2,k}, y'_k) = (g_1^b g_1^a \tilde{x}_{1,i}, g_2^b g_2^a \tilde{x}_{2,i}, h^b h^a \tilde{y}_i).$$

Der Zeuge a ist Prover 1 bekannt, während Prover 2 den Zeugen b kennt. Es sei $\mathcal{H}$ eine kryptografische Hashfunktion.

1. (a) Prover 2 wählt einen Wert $w \in_R \mathbb{Z}_q$, berechnet $u_1 = g_1^w, u_2 = g_2^w$ und $t = h^w$ und sendet u_1, u_2 und t an Prover 1.

   (b) Prover 1 wählt einen Wert $v \in_R \mathbb{Z}_q$ sowie die Werte $d_\ell, r_\ell \in_R \mathbb{Z}_q$ für $1 \leq \ell \leq n_L$, $\ell \neq k$.

Anschließend berechnet er:

$$\alpha_{1,\ell} = \left(x'_{1,\ell}\tilde{x}_{1,i}^{-1}\right)^{d_\ell} g_1^{r_\ell} \text{ für } \ell \neq k,$$

$$\alpha_{2,\ell} = \left(x'_{2,\ell}\tilde{x}_{2,i}^{-1}\right)^{d_\ell} g_2^{r_\ell} \text{ für } \ell \neq k,$$

$$\beta_\ell = \left(y'_\ell \tilde{y}_i^{-1}\right)^{d_\ell} h^{r_\ell} \text{ für } \ell \neq k,$$

$$\alpha_{1,k} = u_1 \cdot g_1^v,$$

$$\alpha_{2,k} = u_2 \cdot g_2^v,$$

$$\beta_k = t \cdot h^v.$$

und schickt diese Werte an den Verifier.

2. (a) Prover 1 berechnet nun die „Challenge“

$$c = \mathcal{H}(E||\alpha_{1,1}|| \ldots ||\alpha_{1,n_L}||\alpha_{2,1}|| \ldots ||\alpha_{2,n_L}||\beta_1|| \ldots ||\beta_{n_L})$$

mit Hilfe einer kryptografischen Hashfunktion $\mathcal{H}$ selbst, wobei $a||b$ die Aneinanderkettung von a und b, und $(\tilde{x}_{1,i}||\tilde{x}_{2,i}||\tilde{y}_i||x'_{1,1}||x'_{2,1}||y'_1|| \ldots ||x'_{1,n_L}||x'_{2,n_L}||y'_{n_L})$ das „Environment“ E ist.

(b) Nun setzt Prover 1 $d_k = c - \sum_{\substack{\ell=1 \\ \ell \neq k}}^{n_L} d_\ell$ und schickt d_k an Prover 2.

(c) Prover 2, der seinen Teil b des Zeugens ξ kennt, berechnet $r = w - bd_k$ und sendet r an Prover 1.

(d) Nun kann Prover 1 mit Hilfe der Kenntnis von a den Wert $r_k = v - ad_k + r$ berechnen. Er schickt die Werte d_ℓ, r_ℓ für $1 \leq \ell \leq n_L$ an den Verifier.

Der Verifier prüft, ob die folgende Gleichung erfüllt ist:

$$\sum_{\ell=1}^{n_L} d_\ell \stackrel{?}{=} \mathcal{H}(E||(x'_{1,1}\tilde{x}_{1,i}^{-1})^{d_1} g_1^{r_1}|| \ldots ||(x'_{n_L}\tilde{x}_{1,i}^{-1})^{d_{n_L}} g_1^{r_{n_L}}||(x'_{1,1}\tilde{x}_{2,i}^{-1})^{d_1} g_2^{r_1}|| \ldots ||(x'_{n_L}\tilde{x}_{2,i}^{-1})^{d_{n_L}} g_2^{r_{n_L}} ||(y'_1\tilde{y}_i^{-1})^{d_1} h^{r_1}|| \ldots ||(y'_{n_L}\tilde{y}_i^{-1})^{d_{n_L}} h^{r_{n_L}})$$

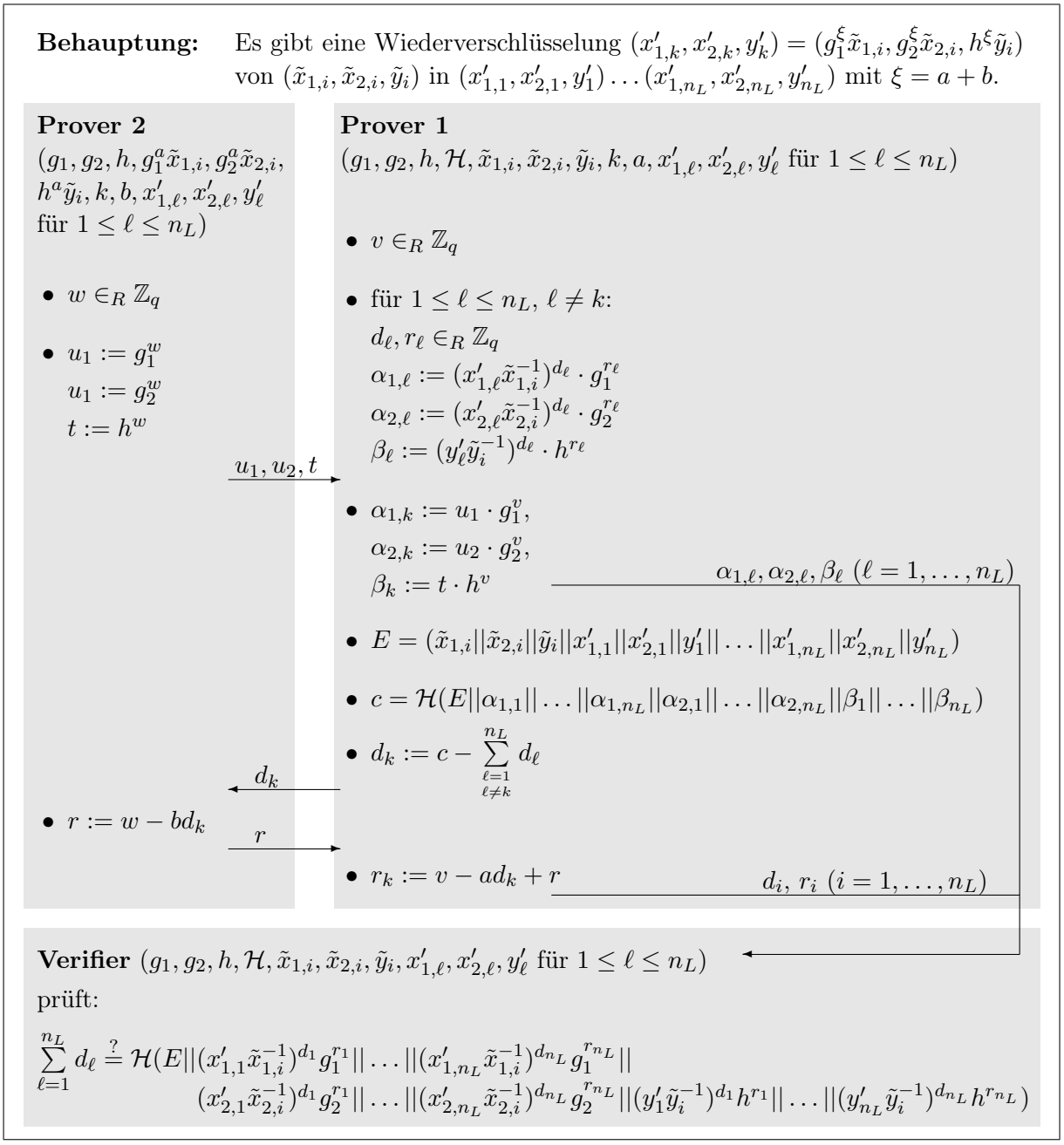

Abbildung 3.8: Nicht-interaktiver 2-Prover WI-Wiederverschlüsselungsbeweis.

Satz 3.8
Der nicht-interaktive Beweis der Existenz einer Wiederverschlüsselung ist durchführbar, korrekt und besitzt die Witness-Indistinguishable-Eigenschaft.

Beweis:

- Der Beweis ist *durchführbar*. Wenn sich alle Teilnehmer korrekt verhalten und tatsächlich $(g_1^\xi \tilde{x}_{1,i}, g_2^\xi \tilde{x}_{2,i}, h^\xi \tilde{y}_i)$ in $\{(x'_{1,1}, x'_{2,1}, y'_1), \dots, (x'_{1,n_L}, x'_{2,n_L}, y'_{n_L})\}$ liegt, kann der Prover immer

eine korrekte „Antwort" auf c geben. Der Prover wählt d_k und r_k, so dass $c = \sum_{\ell=1}^{n_L} d_\ell$ und $\xi d_k + r_k = w + v$ gilt.
Die Werte $\alpha_{1,\ell}, \alpha_{2,\ell}$ und β_ℓ sind für $\ell \neq k$ ebenfalls von Prover 1 so gewählt, dass sie den vom Verifier zu überprüfenden Gleichungen entsprechen:

$$\begin{aligned}
\alpha_{1,k} &= u_1 \cdot g_1^v = g_1^w \cdot g_1^v = g_1^{w+v} = g_1^{(a+b)d_k-(a-b)d_k+w+v} \\
&= g_1^{\xi d_k+(v-ad_k+(w-bd_k))} = g_1^{\xi d_k+r_k} = \left(x'_{1,k}\tilde{x}_{1,i}^{-1}\right)^{d_k} g_1^{r_k}, \\
\alpha_{1,k} &= u_2 \cdot g_2^v = g_2^w \cdot g_2^v = g_2^{w+v} = g_2^{(a+b)d_k-(a-b)d_k+w+v} \\
&= g_2^{\xi d_k+(v-ad_k+(w-bd_k))} = g_2^{\xi d_k+r_k} = \left(x'_{1,k}\tilde{x}_{1,i}^{-1}\right)^{d_k} g_2^{r_k}, \\
\beta_k &= t \cdot h^v = h^w \cdot h^v = h^{w+v} = h^{(a+b)d_k-(a-b)d_k+w+v} \\
&= h^{\xi d_k+(v-ad_k+(w-bd_k))} = h^{\xi d_k+r_k} = \left(y'_k\tilde{y}_i^{-1}\right)^{d_k} h^{r_k}.
\end{aligned}$$

Da schon die einzelnen Eingaben in die kryptografische Hashfunktion gleich den erforderlichen Werten sind, besteht auch die Ausgabe der Hashfunktion die Überprüfung durch den Verifier.

- Das Protokoll ist *korrekt*. Die Prover haben keine Möglichkeit, auf c korrekt zu „antworten", wenn es keine Wiederverschlüsselung in der betrachteten Liste der Geheimtexte gibt.
  Um erfolgreich zu betrügen, müssten die Prover die Challenge vorher erraten und dann zufällige Werte $d_1, \ldots, d_{n_L}$ und $r_1, \ldots, r_{n_L}$ so wählen, dass die Gleichungen des Verifiers erfüllt sind. Die Prover können aber nur auf einen der q möglichen Werte für c eine korrekte „Antwort" geben.

  Angenommen, die Prover könnten auf zwei verschiedene Werte c und c' korrekt antworten, dann müssten sie für mindestens ein $j \in \{1, \ldots, n_L\}$ die Werte d_j und r_j so verändern, dass für die angepassten Werte $d'_j \neq d_j$ und r'_j gilt:

$$c' = d_j + \sum_{\substack{\ell=1 \\ \ell \neq j}}^{n_L} d_\ell.$$

  Außerdem müsste dann gelten:

$$\alpha_{1,j} = \left(x'_{1,j}\tilde{x}_{1,i}^{-1}\right)^{d_j} \cdot g_1^{r_j} \stackrel{!}{=} \left(x'_{1,j}\tilde{x}_{1,i}^{-1}\right)^{d'_j} \cdot g_1^{r'_j}, \tag{3.5}$$

$$\alpha_{2,j} = \left(x'_{2,j}\tilde{x}_{2,i}^{-1}\right)^{d_j} \cdot g_2^{r_j} \stackrel{!}{=} \left(x'_{2,j}\tilde{x}_{2,i}^{-1}\right)^{d'_j} \cdot g_2^{r'_j}, \tag{3.6}$$

$$\beta_j = \left(y'_j\tilde{y}_i^{-1}\right)^{d_j} \cdot h^{r_j} \stackrel{!}{=} \left(y'_j\tilde{y}_i^{-1}\right)^{d'_j} \cdot h^{r'_j}. \tag{3.7}$$

  Da $x'_{1,j}\tilde{x}_{1,i}^{-1} = g_1^{\xi_j}$ für ein ξ_j und $x'_{2,j}\tilde{x}_{2,i}^{-1} = g_2^{\xi_j}$ und $y'_j\tilde{y}_i^{-1} = h^{\tilde{\xi}_j}$ für ein $\tilde{\xi}_j$ gilt, folgt aus den Gleichungen (3.5), (3.6) und (3.7)

$$\begin{aligned}
g_1^{\xi_j \cdot d_j + r_j} &= g_1^{\xi_j \cdot d'_j + r'_j}, \\
g_2^{\xi_j \cdot d_j + r_j} &= g_2^{\xi_j \cdot d'_j + r'_j}, \\
h^{\tilde{\xi}_j \cdot d_j + r_j} &= h^{\tilde{\xi}_j \cdot d'_j + r'_j}.
\end{aligned}$$

Also müssen die beiden folgenden Gleichungen erfüllt sein:

$$\xi_j \cdot d_j + r_j = \xi_j \cdot d'_j + r'_j, \tag{3.8}$$
$$\tilde{\xi}_j \cdot d_j + r_j = \tilde{\xi}_j \cdot d'_j + r'_j. \tag{3.9}$$

Stellt man nun Gleichung (3.8) nach ξ_j und Gleichung (3.9) nach $\tilde{\xi}_j$ um, so erkennt man, dass die beiden Werte ξ_j und $\tilde{\xi}_j$ gleich sein müssen:

$$\xi_j = (r'_j - r_j)(d_j - d'_j)^{-1} = \tilde{\xi}_j.$$

Also gilt:

$$\begin{aligned} x'_{1,j} &= \tilde{x}_{1,i} g_1^{\xi_j}, \\ x'_{2,j} &= \tilde{x}_{2,i} g_2^{\xi_j}, \\ y'_j &= \tilde{y}_i h^{\xi_j}. \end{aligned}$$

Das bedeutet, dass $(x'_{1,j}, x'_{2,j}, y'_j)$ eine Verschlüsselung von $(\tilde{x}_{1,i}, \tilde{x}_{2,i}, \tilde{y}_i)$ unter dem Zeugen ξ_j ist. Dies ist ein Widerspruch zur Annahme, dass es keine Wiederverschlüsselung in der betrachteten Liste gibt.
Die Korrektheit des nicht-interaktiven Beweises ergibt sich dann unter Berücksichtigung der kryptografischen Einweg-Hashfunkion. Durch die Fiat-Shamir-Heuristik ist es auch bei Einsatz dieser Hashfunktion einem unehrlichen Prover nur dann möglich korrekt zu antworten, wenn er c vorher rät. Also kann der Prover mit einer Wahrscheinlichkeit von $\frac{1}{q}$ betrügen. Die Wahrscheinlichkeit, dass sein Betrugsversuch aufgedeckt wird, ist dann $1 - q^{-1}$.

- Der Beweis erfüllt die *Witness-Indistinguishable-Eigenschaft.*
  Es sei $(x'_{1,k'}, x'_{2,k'}, y'_{k'})$, mit $k' \neq k$, eine weitere Wiederverschlüsselung von $(\tilde{x}_{1,i}, \tilde{x}_{2,i}, \tilde{y}_i)$ unter dem zur Wiederverschlüsselung benutzten, von ξ wesentlich verschiedenen Zeugen $\xi' = a' + b'$. Dann gilt $(x'_{1,k'}, x'_{2,k'}, y'_{k'}) = (g^{\xi'} \tilde{x}_{1,i}, g^{\xi'} \tilde{x}_{2,i}, h^{\xi'} \tilde{y}_i)$.

  Prover 2 verhält sich wie oben beschrieben, wählt $w' \in_R \mathbb{Z}_q$, berechnet $u_1 = g_1^{w'}$, $u_2 = g_2^{w'}$ und $t = g^{w'}$ und übermittelt diese Werte an Prover 1. Dieser wählt die Werte $d_\ell, r_\ell \in_R \mathbb{Z}_q$ (für $1 \leq \ell \leq n_L$ und $\ell \neq k'$) sowie $v' \in_R \mathbb{Z}_q$, berechnet $\alpha_{1,1}, \ldots, \alpha_{1,n_L}$, $\alpha_{2,1}, \ldots, \alpha_{2,n_L}$ und $\beta_1, \ldots, \beta_{n_L}$ analog zum oben angegebenen Protokoll und sendet die $\alpha_{1,i}, \alpha_{2,i}$ und β für $i = 1, \ldots, n_L$ zum Verifier. Damit legt sich Prover 1 auf d_ℓ und r_ℓ für alle $\ell = 1, \ldots, n_L$, außer für k', fest.

  Nachdem Prover 1 vom Verifier die Challenge $c \in_r \mathbb{Z}_q$ erhalten hat, wählt er $d_{k'} = c - \sum_{\substack{\ell=1 \\ \ell \neq k'}}^{n_L} d_\ell$ und sendet c und $d_{k'}$ zu Prover 2. Dieser sendet $r' = w' - b' d_{k'}$ zurück.

  Prover 1 berechnet nun $r_{k'} = v' - a' d_{k'} + r'$ und sendet die Werte d_ℓ und r_ℓ für $1 \leq \ell \leq n_L$ zum Verifier, der die im Protokoll beschriebene Verifikation vornimmt.

Der Verifier kann nicht unterscheiden, welchen Zeugen die Prover verwenden, weil

$$\left(x'_{1,k'}\tilde{x}_{1,i}^{-1}\right)^{d_{k'}} g_1^{r_{k'}} = \left(g_1^{\xi'}\right)^{d_{k'}} g_1^{r_{k'}} = g_1^{\xi' \cdot d_{k'} + r_{k'}} = g_1^{(a'+b')d_{k'}+v'-a'd_{k'}+w'-b'd_{k'}} = g_1^{w'+v'} = \alpha_{1,k'}$$

und

$$\left(x'_{2,k'}\tilde{x}_{2,i}^{-1}\right)^{d_{k'}} g_2^{r_{k'}} = \left(g_2^{\xi'}\right)^{d_{k'}} g_2^{r_{k'}} = g_2^{\xi' \cdot d_{k'} + r_{k'}} = g_2^{(a'+b')d_{k'}+v'-a'd_{k'}+w'-b'd_{k'}} = g_2^{w'+v'} = \alpha_{2,k'}$$

und

$$\left(y'_{k'}\tilde{y}_i^{-1}\right)^{d_{k'}} h^{r_{k'}} = \left(h^{\xi'}\right)^{h_{k'}} h^{r_{k'}} = h^{\xi' \cdot d_{k'} + r_{k'}} = h^{(a'+b')d_{k'}+v'-a'd_{k'}+w'-b'd_{k'}} = h^{w'+v'} = \beta_{k'}$$

gelten.

Da es schon bei den einzelnen Eingaben in die kryptografische Hashfunktion für den Verifier unmöglich ist, zu unterscheiden, welchen Zeugen der Prover benutzt, besteht auch die Ausgabe der Hashfunktion die Überprüfung durch den Verifier. Damit besitzt auch diese nicht-interaktive Variante die Witness-Indistinguishable-Eigenschaft.

□

3.7 Geheimnisteilungsverfahren

Bei elektronischen Wahlen ist es wünschenswert, dass niemand alleine im Wahlausschuss die Stimmen einzelner Wähler entschlüsseln und somit nachvollziehen kann, was derjenige gewählt hat. Das Vertrauen sollte verteilt werden. Im Hinblick auf elektronische Wahlen bedeutet das, dass der Wahlausschuss aus mehreren Personen besteht und dass die Registrierung und die Auszählung von mehreren Personen durchgeführt wird. Zu diesem Zweck kann man sogenannte *Geheimnisteilungsverfahren* oder *Secret-Sharing-Schemes* verwenden.

Bei einem Geheimnisteilungsverfahren kann nur eine Teilgruppe oder die gesamte Gruppe, auf die das Geheimnis aufgeteilt wurde, das Geheimnis rekonstruieren. Eine Möglichkeit, die Zugriffsstruktur auf das Geheimnis zu regeln, sind sogenannte *Schwellenschemata* (*threshold-schemes*).

3.7.1 (t, n)-Schwellenschemata

In einem (t, n)-Schwellenschema wird das Geheimnis auf n Personen aufgeteilt. Nur eine Gruppe von t oder mehr Personen kann zusammen das Geheimnis rekonstruieren. Den Wert t nennt man *Schwellenwert* des Verfahrens.

Adi Shamir stellte 1979 ein solches Protokoll vor, [Sha79]. In diesem Protokoll soll ein Geheimnis s auf n Personen so aufgeteilt werden, dass nur t oder mehr Personen s rekonstruieren können. Dazu wählt man ein Polynom $f \in K[x]$ vom Grad $t-1$ über einem endlichen Körper K mit zufälligen Koeffizienten $a_1, \ldots, a_{t-1}$ und $a_0 = s$. Die Teilgeheimnisse sind n verschiedene Punkte $(x_i, f(x_i))$ mit $x_i \neq 0$ für $1 \leq i \leq n$. Zur Vereinfachung setzt man meist $x_i = i$ und erhält somit die Teilgeheimnisse $s_i = f(i)$ für $1 \leq i \leq n$.

Sind weniger als t Punkte von f bekannt, so erhält man ein Gleichungssystem, das nicht eindeutig lösbar ist. Bei t oder mehr gegebenen Punkten lässt sich das Polynom mit der Interpolationsformel von Lagrange wie folgt eindeutig bestimmen.

3.7.2 Lagrange-Interpolation

Für die Stützstellen x_i $(1 \leq i \leq n)$ erfüllt das *Lagrange-Interpolationspolynom*

$$L_n(x) = \sum_{j=1}^{n} \ell_j(x) \cdot f(x_j)$$

mit den *Lagrange-Faktoren*

$$\ell_j(x) = \prod_{\substack{k=1 \\ k \neq j}}^{n} (x - x_k) (x_j - x_k)^{-1}$$

die Interpolationsbedingung $L_n(x_i) = f(x_i)$, da gilt:

$$\ell_j(x_i) = \prod_{\substack{k=1 \\ k \neq j}}^{n} (x_i - x_k) (x_j - x_k)^{-1} = \begin{cases} 1 & : \quad i = j \\ 0 & : \quad i \neq j \end{cases}$$

Außerdem ist $\text{Grad}(L_n(x)) \leq n$. Das bedeutet, dass $L_n(x)$ - aufgrund der eindeutigen Lösbarkeit des aus der Interpolationsbedingung hervorgehenden linearen Gleichungssystems - das eindeutige Interpolationspolynom ist.

Mit Hilfe der Lagrange-Interpolation kann also eine Gruppe bestehend aus mindestens t Personen das Interpolationspolynom und somit das aufgeteilte Geheimnis bestimmen.

3.7.3 Verifizierbare (t, n)-Geheimnisteilungsverfahren

In verifizierbaren Geheimnisteilungsverfahren können die Teilnehmer feststellen, ob die Teilgeheimnisse insofern korrekt sind, dass jeweils t Teilgeheimnisse das gleiche Geheimnis erzeugen.

Definition 3.10 (t-Konsistenz)
Gegeben sei ein (t, n)-Geheimnisteilungsverfahren. Es sei M die Menge aller möglichen Teilgeheimnisse des Systems. Eine Menge $Q \subseteq M$ heißt t-konsistent, wenn jede t-elementige Teilmenge von Q das gleiche Geheimnis rekonstruiert.

Lemma 3.9
Die n Werte $s_1, \ldots, s_n$ sind genau dann t-konsistent bezüglich des Geheimnisteilungsverfahrens von Shamir (siehe Abschnitt 3.7.1), wenn die Punkte $(1, s_1), (2, s_2), \ldots, (n, s_n)$ ein Polynom f mit $\text{Grad}(f) \leq t - 1$ interpolieren.

Beweis:

„⇒" Angenommen, $s_1, \ldots, s_n$ sind t-konsistent. Die t Punkte $(1, s_1), (2, s_2), \ldots, (t, s_t)$ interpolieren dann ein Polynom f mit $\text{Grad}(f) \leq t - 1$ und $f(0) = s$.
Betrachtet man stattdessen die Werte $(2, s_2), \ldots, (t, s_t), (i, s_i)$ mit $i > t$, so interpolieren auch diese Werte ein Polynom $\tilde{f}$ mit $\text{Grad}(\tilde{f}) \leq t - 1$ und $\tilde{f}(0) = s$, da $s_1, \ldots, s_n$ t-konsistent sind. Da $\text{Grad}(f) \leq t - 1$ und $\text{Grad}(\tilde{f}) \leq t - 1$ ist und beide Polynome die t

Punkte $(0, s), (2, s_2), (3, s_3), \ldots, (t, s_t)$ besitzen, folgt, dass die Polynome gleich sind.
Da also die $t+1$ Punkte $(1, s_1), (2, s_2), (3, s_3), \ldots, (t, s_t), (i, s_i)$ noch immer dasselbe Polynom f interpolieren, folgt per Induktion, dass auch die n Punkte $(1, s_1), (2, s_2), \ldots, (n, s_n)$ das Polynom f interpolieren.

„⇐" Angenommen, die n Punkte $(1, s_1), (2, s_2), \ldots, (n, s_n)$ interpolieren ein Polynom f mit $\text{Grad}(f) \leq t-1$, dann genügt auch jede beliebige Teilmenge von t Punkten, um f eindeutig zu bestimmen.
Das bedeutet, dass die Werte $(1, s_1), (2, s_2), \ldots, (n, s_n)$ t-konsistent sind.

□

Beispiel 3.1 (Sicherstellung der Kenntnis eines geheimen Schlüssels)
Es sei G eine Gruppe mit Primzahlordnung q, $G =< g >$ und $s_V \in \mathbb{Z}_q$ der geheime ElGamal-Schlüssel des Teilnehmers V.

Für die Durchführung von Designated-Verifier-Beweisen ist es essentiell, dass der Verifier - in Wahlsystemen der Wähler - seinen eigenen geheimen Schlüssel kennt (siehe Abschnitt 3.5). Es ist jedoch denkbar, dass nicht der Wähler, sondern ein Erpresser oder Stimmenkäufer den geheimen Schlüssel s_V kennt.

Das folgende Protokoll aus [HS00] kann Teil der Schlüsselregistrierung innerhalb einer Public-Key-Infrastruktur oder auch Teil des Wahlprotokolls sein. Es basiert auf dem Geheimnisteilungsverfahren von Paul Feldman, [Fel87]. Der Besitzer des geheimen Schlüssels teilt diesen auf und sendet die Anteile an die Autoritäten. Diese überprüfen, ob er zu h_V gehört und übermitteln die Anteile über einen sicheren Kanal zum Teilnehmer V. Das Protokoll beweist also, dass ein Teilnehmer V seinen geheimen ElGamal-Schlüssel s_V kennt, der zu seinem öffentlichen ElGamal-Schlüssel h_V gehört (so dass $g^{s_V} = h_V$ gilt).

1. Der Teilnehmer V teilt seinen geheimen Schlüssel s_V unter den Personen $A_1, \ldots, A_n$ mittels des Geheimnisteilungsverfahrens von Feldman auf:
   Er wählt zufällige Koeffizienten $a_1, \ldots, a_{t-1}$ gemäß einer Gleichverteilung aus und erhält so ein zufälliges Polynom $f_V(x) = s_V + a_1 x + \ldots + a_{t-1} x^{t-1}$ vom Grad $t-1$.

2. V sendet den Anteil $s_j = f_V(j)$ geheim, beispielsweise[8] verschlüsselt mit dem öffentlichen Schlüssel von A_j, zu A_j, für $j = 1, \ldots, n$.

   Der Teilnehmer V wird auf die Koeffizienten des Polynoms festgelegt, indem er $c_i = g^{a_i}$ für $i = 1, \ldots, t-1$ veröffentlicht.

3. Jede Person A_j verifiziert mit der folgenden Gleichung, ob der empfangene Anteil s_j tatsächlich auf dem festgelegten Polynom $f_V(\cdot)$ liegt:

$$g^{s_j} = h_V \cdot c_1^j \cdots c_{t-1}^{j^{t-1}} = g^{s_V} \cdot g^{a_1 j} \cdots g^{a_{t-1} j^{t-1}} = g^{f_V(j)}$$

[8] Alternativ kann auch A_j innerhalb eines Wahlprotokolls mit sicherem Kanal von A_j zu V zunächst Daten über diesen Kanal zu V schicken, mit denen V dann den Anteil mit dem One-Time-Pad verschlüsseln kann.

4. Damit sichergestellt ist, dass der Wähler wirklich s_V kennt und nicht eine weitere Person die Anteile an A_j verschickt hat, senden alle A_j ihren Anteil über einen sicheren Kanal zum Teilnehmer V.

3.7.4 (t, n)-Schwellenverschlüsselungsschemata

Mit Geheimnisteilungsverfahren können beliebige Geheimnisse aufgeteilt werden. So kann man den geheimen Schlüssel s eines asymmetrischen Verschlüsselungsverfahrens in n Teilgeheimnisse aufteilen, so dass eine Nachricht nur entschlüsselt werden kann, wenn t oder mehr Besitzer eines Teilgeheimnisses zusammenarbeiten.

Auf elektronische Wahlen bezogen bedeutet das, dass nur eine Gruppe von t oder mehr Personen A_j, $j = 1, \ldots, n$, des Wahlausschusses eine verschlüsselte Nachricht entschlüsseln kann, wenn man das Schlüsselpaar (s, h) so konstruiert, dass jede Person des Wahlausschusses einen Anteil s_j von s in einem (t, n)-Geheimnisteilungsverfahren erhält. Nur t oder mehr Autoritäten können dann eine unter h verschlüsselte Stimme entschlüsseln.

3.7.5 Ein (t, n)-Schwellensystem mit ElGamal

Wenn eine Person, ein sogenannter Dealer, das Geheimnis aufteilt, so kennt er den geheimen Schlüssel. Das bedeutet, dass sich dann aber bei elektronischen Wahlen das Vertrauen, das einer einzelnen Person im Wahlausschuss entgegenzubringen wäre, auf eine andere Person, den Dealer, verlagert. Wünschenswert ist ein System, das ohne eine solche *vertrauenswürdige dritte Partei*, auch *Trusted Third Party* genannt, auskommt.

Torben Pedersen hat gezeigt (siehe [Ped91]), dass man einen geheimen Schlüssel so aufteilen kann, dass einzelne an der Schlüsselerzeugung beteiligte Personen nichts über den Schlüssel aussagen können. Außerdem muss der aufgeteilte geheime Schlüssel s nicht explizit berechnet werden. Der zugrundeliegende Klartext kann entschlüsselt werden, ohne dass ein Teilnehmer den geheimen Schlüssel kennt. Der Schlüssel kann daher mehrfach verwendet werden.

Um auch die Verschlüsselung so zu gestalten, dass nur eine Gruppe von t oder mehr Personen A_j, $(j = 1, \ldots, n)$, eine verschlüsselte Nachricht entschlüsseln kann, wird das Schlüsselpaar (s, h) so konstruiert, dass jede Person A_j einen Anteil s_j von s in einem (t, n)-Geheimnisteilungssystem erhält und öffentlich auf diesen Anteil über $h_j = g^{s_j}$ festgelegt wird.
Die Vorgehensweise wird im Folgenden beschrieben:

1. Jede Person A_j, $(j = 1, \ldots, n)$, veröffentlicht $\beta_j := g^{\alpha_j}$ mit einem gemäß einer Gleichverteilung aus $\mathbb{Z}_q$ zufällig ausgewählten Wert α_j.

2. Jeder Teilnehmer A_j beweist, dass er tatsächlich α_j kennt, wie es in Abschnitt 3.2.2 beschrieben wird.

3. Jeder Teilnehmer A_j wählt ein Polynom f_j mit Koeffizienten $a_{i,j} \in_R \mathbb{Z}_q$, $(i = 1, \ldots, t-1)$, mit $\text{Grad}(f_j) = t - 1$ und $f_j(0) = \alpha_j$, also $a_{0,j} := \alpha_j$.

4. Nun veröffentlichen alle A_j die Werte[9] $g^{a_{i,j}}$ für $i = 1, \ldots, t-1$.

[9] Der Wert $g^{a_{0,j}}$ wurde bereits im ersten Schritt veröffentlicht.

5. Nachdem alle Teilnehmer die Werte veröffentlicht haben, sendet A_j, $(j = 1, \ldots, n)$, den Wert $s_{j,k} := f_j(k)$ mit einer Signatur über einen sicheren Kanal zur Person A_k, für $k = 1, \ldots, n$, $k \neq j$.

6. Jeder Teilnehmer A_k überprüft die Signatur und Korrektheit des Wertes $s_{j,k}$, indem er feststellt, ob $g^{s_{j,k}} = \prod_{i=0}^{t-1} g^{a_{i,j}k^i}$ gilt, denn für korrekte $s_{j,k}$ gilt

$$g^{s_{j,k}} = g^{f_j(k)} = g^{\sum_{i=0}^{t-1} a_{i,j}k^i} = \prod_{i=0}^{t-1} g^{a_{i,j}k^i}.$$

7. Jede Person A_k berechnet ihr Teilgeheimnis s_k als Summe aller $s_{j,k}$ und veröffentlicht $h_k := g^{s_k}$ als zugehörigen öffentlichen Teilgeheimnisschlüssel.

Bei einer Zusammenarbeit von t oder mehr Teilnehmern kann der geheime Schlüssel s berechnet werden, indem man das Lagrange-Interpolationspolynom (siehe Abschnitt 3.7.2) vom Grad $t-1$ an der Stelle 0 berechnet.

Sei I eine t-elementige Indexmenge der zusammenarbeitenden Personen A_j. Man erhält s durch:

$$f(0) = \sum_{j \in I} \ell_j(0) \cdot s_j. \tag{3.10}$$

Dabei sind $\ell_j(0) = \prod_{i \in I \setminus \{j\}} i(i-j)^{-1}$ die Lagrange-Faktoren an der Stelle 0. Nach Abschnitt 3.7.2 ist das Polynom und somit der Schlüssel eindeutig.

Der geheime Schlüssel kann also aus t Teilgeheimnissen berechnet werden. Es ist jedoch sinnvoller, nicht den geheimen Schlüssel selbst zu berechnen, sondern eine Nachricht ohne explizite Rekonstruktion des geheimen Schlüssels zu dechiffrieren, um den geheimen Schlüssel nicht nur einmal verwenden zu können.

Entschlüsselung einer ElGamal-verschlüsselten Nachricht ohne explizite Berechnung des geheimen Schlüssels

Gegeben sei eine ElGamal-verschlüsselte Nachricht $(x, y) = (g^\alpha, h^\alpha m)$ mit $h = g^s$. Der geheime Schlüssel s ist auf n Teilnehmer A_j, $(j = 1, \ldots, n)$, aufgeteilt.

Die Nachricht soll entschlüsselt werden, ohne dass der geheime Schlüssel explizit rekonstruiert wird, damit nach der Entschlüsselung der Nachricht weiterhin kein Teilnehmer den geheimen Schlüssel kennt.
Das Verfahren wird im Folgenden beschrieben.

1. Jede Person A_j veröffentlicht $z_j = x^{s_j}$ und zeigt mit einem Zero-Knowledge-Beweis (siehe Beispiel 3.2.2), dass $\log_g(h_j) = \log_x(z_j)$ gilt.

2. Es sei I eine Indexmenge der t Personen, die den Zero-Knowledge-Beweis bestehen, dann gilt:

$$\begin{aligned} m = g^{-\alpha s} g^{\alpha s} m &\underset{\text{nach (3.10)}}{=} (g^\alpha)^{-\sum_{j \in I} \ell_j(0) \cdot s_j} h^\alpha m = \left(x^{-\sum_{j \in I} \ell_j(0) \cdot s_j} \right) \cdot y \\ &= \left(\prod_{j \in I} x^{-s_j \cdot \ell_j(0)} \right) \cdot y = \left(\prod_{j \in I} z_j^{-\ell_j(0)} \right) \cdot y. \end{aligned}$$

Die Entschlüsselung ist global verifizierbar, da sich jeder anhand des Zero-Knowledge-Beweises überzeugen kann, dass richtig entschlüsselt wird.

Die Entschlüsselung ist robust. Das bedeutet, dass sie erfolgreich durchgeführt werden kann, solange mindestens t Personen den Beweis bestehen. Betrugsversuche werden erkannt und die entsprechende Person kann ausgeschlossen werden.

3.7.6 Ein (t, n)-Schwellensystem mit modifiziertem ElGamal

Für den Nachweis der Erpressungsresistenz des in Kapitel 6 vorgestellten neuen Wahlverfahrens wird die Verwendung einer Verschlüsselung wie der in Abschnitt 2.4 eingeführten modifizierten ElGamal-Verschlüsselung benötigt. Beim modifizierten ElGamal-Verfahren besteht der geheime Schlüssel allerdings aus zwei geheimen Werten s_1 und s_2. Daher wird hier in diesem Abschnitt ein neues $(t, n)-$Schwellensystem mit modifiziertem ElGamal-Verfahren vorgestellt. Wie in Abschnitt 3.7.4 wird dabei der geheime Schlüssel so aufgeteilt, dass einzelne an der Schlüsselerzeugung beteiligte Personen nichts über den Schlüssel aussagen können.

Um auch die Verschlüsselung so zu gestalten, dass nur eine Gruppe von t oder mehr Personen A_j, für $j \in \{1, \ldots, n\}$, eine verschlüsselte Nachricht entschlüsseln kann, werden die Schlüsselwerte s_1, s_2, h so konstruiert, dass jede Person A_j einen Anteil $s_{1,j}$ von s_1 und $s_{2,j}$ von s_2 in einem (t, n)-Geheimnisteilungssystem erhält und öffentlich auf diese Anteile über $h_j = g_1^{s_{1,j}} g_2^{s_{2,j}}$ festgelegt wird.
Die Vorgehensweise wird im Folgenden beschrieben:

1. Jede Person A_j, $(j = 1, \ldots, n)$, veröffentlicht $\beta_{1,j} := g_1^{\alpha_{1,j}}$ und $\beta_{2,j} := g_2^{\alpha_{2,j}}$ mit gemäß einer Gleichverteilung aus $\mathbb{Z}_q$ zufällig ausgewählten Werten $\alpha_{1,j}, \alpha_{2,j}$.

2. Jeder Teilnehmer A_j beweist, dass er tatsächlich $\alpha_{1,j}$ und $\alpha_{2,j}$ kennt (vgl. Abschnitt 3.2.2).

3. Jeder Teilnehmer A_j wählt zwei Polynome $f_{1,j}, f_{2,j}$ mit Koeffizienten $a_{1,i,j} \in_R \mathbb{Z}_q$ bzw. $a_{2,i,j} \in_R \mathbb{Z}_q$ $(i = 1, \ldots, t-1)$, mit $\mathrm{Grad}(f_{1,j}) = \mathrm{Grad}(f_{2,j}) = t - 1$ und $f_{1,j}(0) = \alpha_{1,j}$, also $a_{1,0,j} := \alpha_{1,j}$ und $f_{2,j}(0) = \alpha_{2,j}$, also $a_{2,0,j} := \alpha_{2,j}$.

4. Nun veröffentlichen alle A_j die Werte[10] $g_1^{a_{1,i,j}}, g_2^{a_{2,i,j}}$ für $i = 1, \ldots, t-1$.

5. Nachdem alle Teilnehmer die Werte veröffentlicht haben, sendet A_j, $(j = 1, \ldots, n)$, die Werte $s_{1,j,k} := f_{1,j}(k)$ und $s_{2,j,k} := f_{2,j}(k)$ mit einer Signatur über einen sicheren Kanal zur Person A_k (für $k = 1, \ldots, n$, $k \neq j$).

6. Jeder Teilnehmer A_k überprüft die Signaturen und Korrektheit von $s_{1,j,k}, s_{2,j,k}$ indem er feststellt, ob $g_1^{s_{1,j,k}} = \prod_{i=0}^{t-1} g_1^{a_{1,i,j}k^i}$ und $g_2^{s_{2,j,k}} = \prod_{i=0}^{t-1} g_1^{a_{2,i,j}k^i}$ gelten, denn für korrekte $s_{1,j,k}, s_{2,j,k}$ gilt
$g_1^{s_{1,j,k}} = g_1^{f_{1,j}(k)} = g_1^{\sum_{i=0}^{t-1} a_{1,i,j}k^i} = \prod_{i=0}^{t-1} g_1^{a_{1,i,j}k^i}$ bzw. $g_2^{s_{2,j,k}} = g_2^{f_{2,j}(k)} = g_2^{\sum_{i=0}^{t-1} a_{2,i,j}k^i} = \prod_{i=0}^{t-1} g_2^{a_{2,i,j}k^i}$.

[10] Die Werte $g_1^{a_{1,0,j}}$ und $g_2^{a_{2,0,j}}$ wurden bereits im ersten Schritt veröffentlicht.

7. Jede Person A_k berechnet ihre Teilgeheimnisse $s_{1,k}, s_{2,k}$ jeweils als Summe aller $s_{1,j,k}$ bzw. $s_{2,j,k}$ und veröffentlicht $h_k := g_1^{s_{1,k}} g_2^{s_{2,k}}$ als zugehörigen öffentlichen Teilgeheimnisschlüssel.

Bei einer Zusammenarbeit von t oder mehr Teilnehmern kann der geheime Schlüssel s_1, s_2 berechnet werden, indem man das Lagrange-Interpolationspolynom (siehe Abschnitt 3.7.2) vom Grad $t-1$ an der Stelle 0 berechnet.

Sei I eine t-elementige Indexmenge der zusammenarbeitenden Personen A_j. Man erhält s_1 durch:

$$f_1(0) = \sum_{j \in I} \ell_j(0) \cdot s_{1,j} \tag{3.11}$$

und s_2 mittels:

$$f_1(0) = \sum_{j \in I} \ell_j(0) \cdot s_{2,j}. \tag{3.12}$$

Dabei sind $\ell_j(0) = \prod_{i \in I \setminus \{j\}} i\,(i-j)^{-1}$ die Lagrange-Faktoren an der Stelle 0. Nach Abschnitt 3.7.2 ist das jeweilige Polynom und somit der Schlüssel eindeutig.

Wie im Fall der ElGamal-Verschlüsselung ist es sinnvoller, nicht den geheimen Schlüssel selbst zu berechnen, sondern eine Nachricht ohne explizite Rekonstruktion des geheimen Schlüssels zu dechiffrieren, um den geheimen Schlüssel mehrfach verwenden zu können.

Entschlüsselung einer modifiziert-ElGamal-verschlüsselten Nachricht ohne explizite Berechnung des geheimen Schlüssels

Gegeben sei eine modifiziert-ElGamal-verschlüsselte Nachricht $(g_1^\alpha, g_2^\alpha, h^\alpha m)$ mit $h = g_1^{s_1} g_2^{s_2}$. Der geheime Schlüssel s_1, s_2 ist auf n Teilnehmer A_j, $(j = 1, \ldots, n)$, aufgeteilt.

Die Nachricht soll entschlüsselt werden, ohne dass der geheime Schlüssel explizit rekonstruiert wird, damit nach der Entschlüsselung der Nachricht weiterhin kein Teilnehmer den geheimen Schlüssel kennt.
Das Verfahren wird im Folgenden beschrieben.

1. Jede Person A_j veröffentlicht $z_{1,j} = (g_1^\alpha)^{s_{1,j}}$, $z_{2,j} = (g_2^\alpha)^{s_{2,j}}$ und beweist mittels Zero-Knowledge-Beweisen (siehe Abschnitt 3.2.2), dass die Gleichungen

   $$\log_{g_1}(h_{1,j}) = \log_{(g_1^\alpha)}(z_{1,j}) \text{ und } \log_{g_2}(h_{2,j}) = \log_{(g_2^\alpha)}(z_{2,j})$$

   gelten.

2. Es sei I eine Indexmenge der t Personen, die den Zero-Knowledge-Beweis bestehen, dann gilt:

$$\begin{aligned} m &= g_1^{-\alpha s_1} g_2^{-\alpha s_2} g_1^{\alpha s_1} g_2^{-\alpha s_2} m \\ &\underset{\text{nach (3.11) und (3.12)}}{=} (g_1^\alpha)^{-\sum_{j \in I} \ell_j(0) \cdot s_{1,j}} (g_2^\alpha)^{-\sum_{j \in I} \ell_j(0) \cdot s_{2,j}} h^\alpha m \\ &= \left(\prod_{j \in I} (g_1^\alpha)^{-s_{1,j} \cdot \ell_j(0)} \right) \cdot \left(\prod_{j \in I} (g_1^\alpha)^{-s_{1,j} \cdot \ell_j(0)} \right) \cdot h^\alpha m \\ &= \left(\prod_{j \in I} z_{1,j}^{-\ell_j(0)} \right) \cdot \left(\prod_{j \in I} z_{2,j}^{-\ell_j(0)} \right) \cdot h^\alpha m \end{aligned}$$

Die Entschlüsselung ist global verifizierbar und robust, da sich jeder anhand des Zero-Knowledge-Beweises überzeugen kann, dass richtig entschlüsselt wird.

3.8 MIX-Netze

Ein MIX-Netz erhält Geheimtexte von den Benutzern und gibt eine permutierte Liste von Klartexten aus, ohne zu offenbaren, wer welchen Klartext gesendet hat, d. h. welcher Klartext zu welchem Geheimtext gehört. David Chaum führte 1981 das erste Modell eines solchen MIX-Netzes ein [Cha81]. MIX-Netze werden unter anderem zur anonymen Kommunikation, bei elektronischen Zahlungs- und Wahlsystemen angewendet.

Es gibt auch Fälle, in denen zwar eine Permutation, aber keine Entschlüsselung gewünscht wird. Allgemein funktioniert ein MIX-Netz wie folgt.

3.8.1 Aufbau eines MIX-Netzes

Es sei E eine probabilistische Verschlüsselungsfunktion. Zunächst verschlüsselt der Sender V_i die Nachricht $d_{i,j}$ mit dem öffentlichen Schlüssel h_{A_j} des Empfängers A_j:

$$\tilde{C}_{i,j} := E_{h_{A_j}}(d_{i,j}).$$

Anschließend verschlüsselt er den Geheimtext $\tilde{C}_{i,j}$ zusammen mit der Empfängeradresse Adr_{A_j} unter dem öffentlichen Schlüssel h_{MIX} des MIX-Servers:

$$C_{i,j} := E_{h_{MIX}}(E_{h_{A_j}}(d_{i,j}), Adr_{A_j}).$$

Dann sendet er $C_{i,j}$ zum MIX.

Dieser entschlüsselt mit Hilfe seines privaten Schlüssels s_{MIX} die Nachricht $C_{i,j}$ und erhält den Chiffretext $\tilde{C}_{i,j}$ zu $d_{i,j}$ und die Empfängeradresse Adr_{A_j}. Der MIX sendet $\tilde{C}_{i,j}$ an A_j.

Der Empfänger A_j dechiffriert den erhaltenen Geheimtext mit Hilfe seines privaten Schlüssels s_{A_j} und erhält so $d_{i,j}$. Er weiß aber nicht, dass $d_{i,j}$ vom Sender V_i stammt. Damit eine gleichzeitige Überwachung von Nachrichteneingang und -ausgang keine Rückschlüsse auf den Sender zulässt, sollte der MIX die Nachrichten zeitlich versetzt und in permutierter Reihenfolge ausgeben.

Es ist vorteilhaft, mehrere MIX-Server in Reihe zu schalten, um eine größere Sicherheit zu erreichen. Die Anonymität in einer solchen *MIX-Netz-Kaskade* ist gewährleistet, solange sich mindestens ein MIX-Server korrekt verhält.

3.8.2 Anforderungen an MIX-Netze

Ein MIX permutiert und entschlüsselt eine Liste von Geheimtexten $(C_1, \ldots, C_N)$ und gibt die permutierte entschlüsselte Liste $(\tilde{C}_1, \ldots, \tilde{C}_N)$ aus.

Definition 3.11 (Durchführbarkeit)
Ein MIX-Netz erfüllt die Durchführbarkeit, wenn bei einem ehrlichen Verhalten aller MIX-Server die Ausgabe eine korrekte Liste entschlüsselter und zufällig permutierter Nachrichten ist.

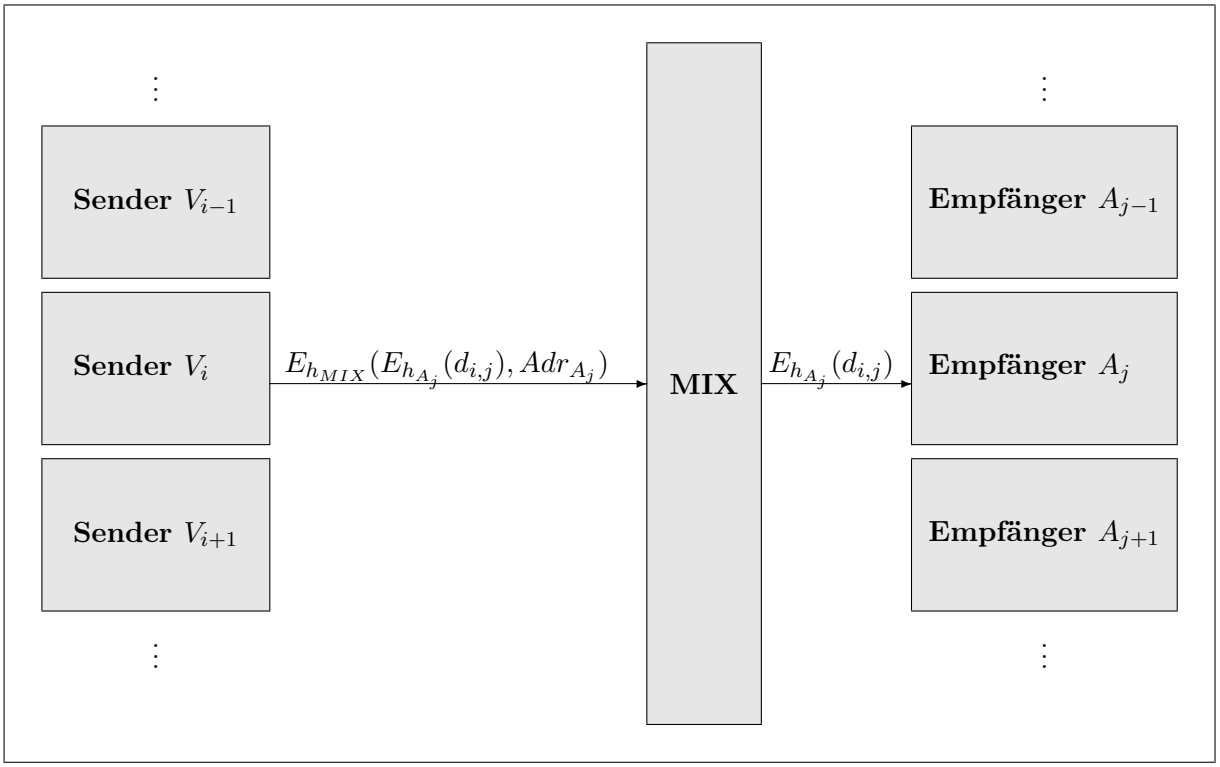

Abbildung 3.9: Ein MIX-Netz mit einem MIX-Server.

Definition 3.12 (Anonymität)
Wenn die Anzahl korrupter und kooperierender MIX-Server einen bestimmten Schwellenwert nicht überschreitet, so darf für alle Teilnehmer die Erfolgswahrscheinlichkeit, die Beziehung zwischen den Ein- und Ausgaben des MIX-Netzes herauszufinden, nicht größer sein als die Wahrscheinlichkeit, die richtige Permutation der Werte zufällig zu raten. Ein solches MIX-Netz erfüllt die *Geheimhaltung* (*Anonymität*).

Definition 3.13 (Verifizierbarkeit)
Die Ausgabe eines MIX-Netzes ist inkorrekt, wenn es keine Bijektion π zwischen den Ein- und Ausgabewerten gibt oder wenn es einen Eingabewert a gibt, zu dem $\pi(a)$ nicht die korrekte Entschlüsselung von a ist. Ein MIX-Netz nennt man *verifizierbar*, wenn die Wahrscheinlichkeit vernachlässigbar ist, dass eine inkorrekte Ausgabe akzeptiert wird.

Eine allgemeine Methode, um Verifizierbarkeit zu erreichen, besteht darin, jeden MIX-Server in einem nicht-interaktiven Zero-Knowledge-Beweis zeigen zu lassen, dass er sich korrekt verhalten hat. Sako und Kilian [SK95] stellten ein solches effizientes Beweissystem auf. Ein weiteres verifizierbares MIX-Netz ist beispielsweise das von Wikström [Wik05].

Definition 3.14 (Robustheit von MIX-Netzen)
Wenn eine Gruppe von MIX-Servern von den Protokollvorschriften abweicht, die anderen ehrlichen MIX-Server dies erkennen und die Identität der unehrlichen MIX-Server effizient herausfin-

den können, so nennt man das MIX-Netz *robust*.

Die MIX-Netze von Chaum [Cha81] oder Park, Itoh und Kurosawa [PIK93] sind allerdings nicht robust. Bricht nur ein MIX-Server ab, so stoppt das ganze System. Ogata, Kurosawa, Sako und Takatani [OKST97] stellten 1997 das erste MIX-Netz vor, das gleichzeitig robust, anonym und verifizierbar ist.

Definition 3.15 (Resilient)
Ein MIX-Netz, das anonym, verifizierbar und robust ist, nennt man *resilient* (*unverwüstlich*).

3.8.3 Verschlüsselungsverfahren in MIX-Netzen

Das MIX-Netz von Chaum [Cha81] gewährleistet Anonymität nur unter der Bedingung, dass alle Sender ehrlich sind. Pfitzmann und Pfitzmann stellten 1989 einen Angriff gegen die RSA-Implementierung des MIX-Netzes von Chaum vor (siehe [PP89]).

Das eigentliche Konzept eines MIX-Netzes musste daraufhin verfeinert werden, um diesem Angriff zu begegnen. MIX-Netze von Chaum, die auf dem RSA basieren, weisen ein weiteres Problem auf. Die Größe eines jeden Geheimtextes ist proportional zur Anzahl der MIX-Server.

Park, Itoh und Kurosawa [PIK93] lösten dieses Problem, indem sie ElGamal zur Verschlüsselung benutzten, so dass die Größe der Geheimtexte unabhängig von der Anzahl der MIX-Server wurde. Da das ElGamal-Verfahren ohnehin probabilistisch ist, muss keine Zufallszahl zusätzlich an die Nachricht angehängt werden.

Die überwiegende Zahl der seither vorgestellten MIX-Netze basiert auf Verschlüsselung mit ElGamal (z. B. [Abe98], [Jak98] oder [DK00]).

3.9 Test auf Gleichheit zugrundeliegender Klartexte

In diesem Abschnitt wird ein Spezialfall des folgenden Szenarios behandelt. Eine festgelegte Anzahl von Beteiligten möchte einen Funktionswert einer Funktion f berechnen. Die Eingaben in f sollen dabei geheim bleiben. Lediglich der Funktionswert soll bei dieser Multiparty-Computation von n Instanzen berechnet werden.

Spezielle Problemstellungen, die unter dieses Hauptproblem fallen sind z. B. das Millionärsproblem[11] oder private Auktionen[12].

Im Folgenden wird ein Test beschrieben und untersucht, mit dem man entscheiden kann, ob die den probabilistisch verschlüsselten Geheimtexten zugrundeliegenden Klartexte gleich sind. Dieser Test auf Gleichheit der Klartexte findet Anwendung in erpressungsresistenten Wahlsystemen, wie dem in Kapitel 6. Er wurde erstmals von Jakobsson und Juels in [JJ00] vorgestellt und wird zunächst hier so abgeändert, dass er mit der modifizierten ElGamal-Verschlüsselung arbeitet und anschließend analysiert.

[11] Man möchte herausfinden, welcher Millionär mehr Geld hat, ohne die Beträge zu veröffentlichen.

[12] Hier möchte man wissen, wer die Auktion gewinnt, ohne jedoch die Beträge der anderen Gebote erkennen zu können.

Gegeben sei eine Gruppe G mit Primzahlordnung q. Die Werte g_1 und g_2 seien Generatoren von G. Die in diesem Test betrachteten Geheimtexte seien unter dem in Abschnitt 2.4 beschriebenen modifizierten ElGamal-Kryptosystem mit verteilten geheimen Schlüsseln s_1, s_2 zu einem öffentlichen Schlüssel $h = g_1^{s_1} g_2^{s_2}$ verschlüsselt.

Der Test soll bei zwei modifiziert ElGamal verschlüsselten Klartexten m_1 und m_2 anhand der Geheimtexte $(x_{1,1}, x_{2,1}, y_1) = (g_1^{a_1}, g_2^{a_1}, h^{a_1} m_1)$ und $(x_{1,2}, x_{2,2}, y_2) = (g_1^{a_2}, g_2^{a_2}, h^{a_2} m_2)$ entscheiden, ob $m_1 = m_2$ gilt, ohne aber m_1 und m_2 explizit zu bestimmen.
Innerhalb des Tests betrachtet man den Geheimtext $(x_1, x_2, y) = (x_{1,1} x_{1,2}^{-1}, x_{2,1} x_{2,2}^{-1}, y_1 y_2^{-1})$.
Wenn den Geheimtexten $(x_{1,1}, x_{2,1}, y_1)$ und $(x_{1,2}, x_{2,2}, y_2)$ der gleiche Klartext zugrunde liegt, so ist (x_1, x_2, y) eine Verschlüsselung des Klartextes 1, andernfalls stellt (x_1, x_2, y) eine Verschlüsselung von $m_1 m_2^{-1}$ dar. Die beteiligten Instanzen P_i (für $1 \leq i \leq n$) blenden daher zunächst (x_1, x_2, y), indem sie die Einträge einzeln mit $z_i \in \mathbb{Z}_q$ potenzieren und dann das Ergebnis gemeinsam entschlüsseln. Wenn der zugrundeliegende Klartext eine 1 ist, wird er durch diese Blendung intakt gelassen. Andernfalls wird er durch diese Berechnungen zufällig geblendet und der Zusammenhang zwischen den beiden Klartexten m_1 und m_2 verschleiert.

3.9.1 Pedersen-Commitment

Damit die Blendung der Werte x_1, x_2 und y auch beweisbar korrekt durchgeführt wird, muss sich jeder der Beteiligten zunächst auf einen Wert z_i festlegen. Dies geschieht über ein sogenanntes *Pedersen-Commitment* (vgl. [Ped91]).

Satz 3.10
Gegeben seien eine Gruppe G mit Primzahlordnung $|G| = q$, ein Generator g von G und eine Zufallszahl $r_i \in_R \mathbb{Z}_q$. Es sei ein weiterer Generator $h = g^s$ von G gegeben. Der Wert $s \in \mathbb{Z}_q$, $s \neq q$, ist dabei den Beteiligten nicht bekannt.

- Das Pedersen-Commitment $c_i = g^{z_i} h^{r_i}$ für den Wert $z_i \in \mathbb{Z}_q$ bindet den Mitspieler A_i' rechnerisch, d. h. A_i' kann unter der Diskreter-Logarithmus-Annahme keine Werte $r_i', z_i' \in \mathbb{Z}_q$ mit $z_i' \neq z_i$ und $g^{z_i'} h^{r_i'} = c_i$ finden.
- Das Pedersen-Commitment offenbart keine Informationen im informationstheoretischen Sinn, d. h. die Werte c_i verbergen z_i perfekt.

Beweis:

- Da die Werte x_1, x_2, y öffentlich feststehen, besteht höchstens eine Angriffsmöglichkeit in der Wahl der Werte z_i bzw. r_i.

  Angenommen, ein polynomiell beschränkter Mitspieler A_i' könnte Werte $z_i', r_i' \in \mathbb{Z}_q$ finden, so dass zwar $g^{z_i} h^{r_i} = g^{z_i'} h^{r_i'}$, aber $z_i \neq z_i'$ und somit auch $r_i \neq r_i'$ gilt. Dann hätte A_i' eine weitere von $g^{z_i} h^{r_i}$ verschiedene Darstellung von c_i gefunden und könnte den diskreten Logarithmus $\log_g h = \frac{z_i - z_i'}{r_i' - r_i}$ berechnen. Das steht im Widerspruch zur Diskreter-Logarithmus-Annahme (vgl. Abschnitt 2.2.4).
  Das bedeutet, dass A_i' über c_i auf den Wert z_i rechnerisch festgelegt wird.

- Da h ein Generator von G ist, wird für feste Werte g, h und z_i durch $r_i \mapsto g^{z_i} h^{r_i}$ eine bijektive Abbildung von $\mathbb{Z}_q$ auf G definiert. Somit sind die Commitments c_i für jedes z_i gleichverteilt über G. Die Werte c_i verbergen also z_i perfekt.

□

3.9.2 Test auf Gleichheit

Im Einzelnen läuft dann das Protokoll des Tests auf Gleichheit der zugrundeliegenden Klartexte wie folgt ab:

1. Jeder Mitspieler P_i (für $1 \leq i \leq n$) wählt Werte $z_i, r_i \in \mathbb{Z}_q$ und veröffentlicht ein Pedersen-Commitment $c_i = g^{z_i} h^{r_i}$ für z_i.

2. Nun berechnet jeder Mitspieler P_i den Geheimtext $(\tilde{x}_{1,i}, \tilde{x}_{2,i}, \tilde{y}_i) = (x_1^{z_i}, x_2^{z_i}, y^{z_i})$ und überträgt diesen.

3. Zudem muss jeder Mitspieler in einem Zero-Knowledge-Beweis (vgl. Abschnitt 3.2.3) zeigen, dass er $(z_i, r_i) \in \mathbb{Z}_q^2$ mit $c_i = g^{z_i} h^{r_i}$, $\tilde{x}_{1,i} = x_1^{z_i}$, $\tilde{x}_{2,i} = x_2^{z_i}$ und $\tilde{y}_i = y^{z_i}$ kennt.

4. Die Beteiligten berechnen nun $(x'_1, x'_2, y') = \left(\prod_{i=1}^{n} \tilde{x}_{1,i}, \prod_{i=1}^{n} \tilde{x}_{2,i}, \prod_{i=1}^{n} \tilde{y}_i \right)$.

5. Die Mitspieler entschlüsseln den Geheimtext (x'_1, x'_2, y') verteilt (vgl. Abschnitt 3.7.6), so dass weiterhin keiner der Beteiligten den gemeinsamen geheimen Schlüssel kennt.

6. Ist der Geheimtext 1, so schließt man daraus mit der hohen Wahrscheinlichkeit $1 - \frac{1}{q}$ (siehe Beweis zu Satz 3.11 (a)), dass $(x_{1,1}, x_{2,1}, y_1)$ und $(x_{1,2}, x_{2,2}, y_2)$ der gleiche Klartext zugrunde liegt.
   Wenn der Geheimtext ungleich 1 ist, folgt daraus, dass $m_1 \neq m_2$ sein muss.

Satz 3.11

(a) Das vorgestellte Protokoll ist durchführbar.

(b) Unter der Diffie-Hellman-Annahme ist das Protokoll korrekt.

(c) Das Protokoll besitzt unter der Diffie-Hellman-Entscheidungsannahme die Zero-Knowledge-Eigenschaft.

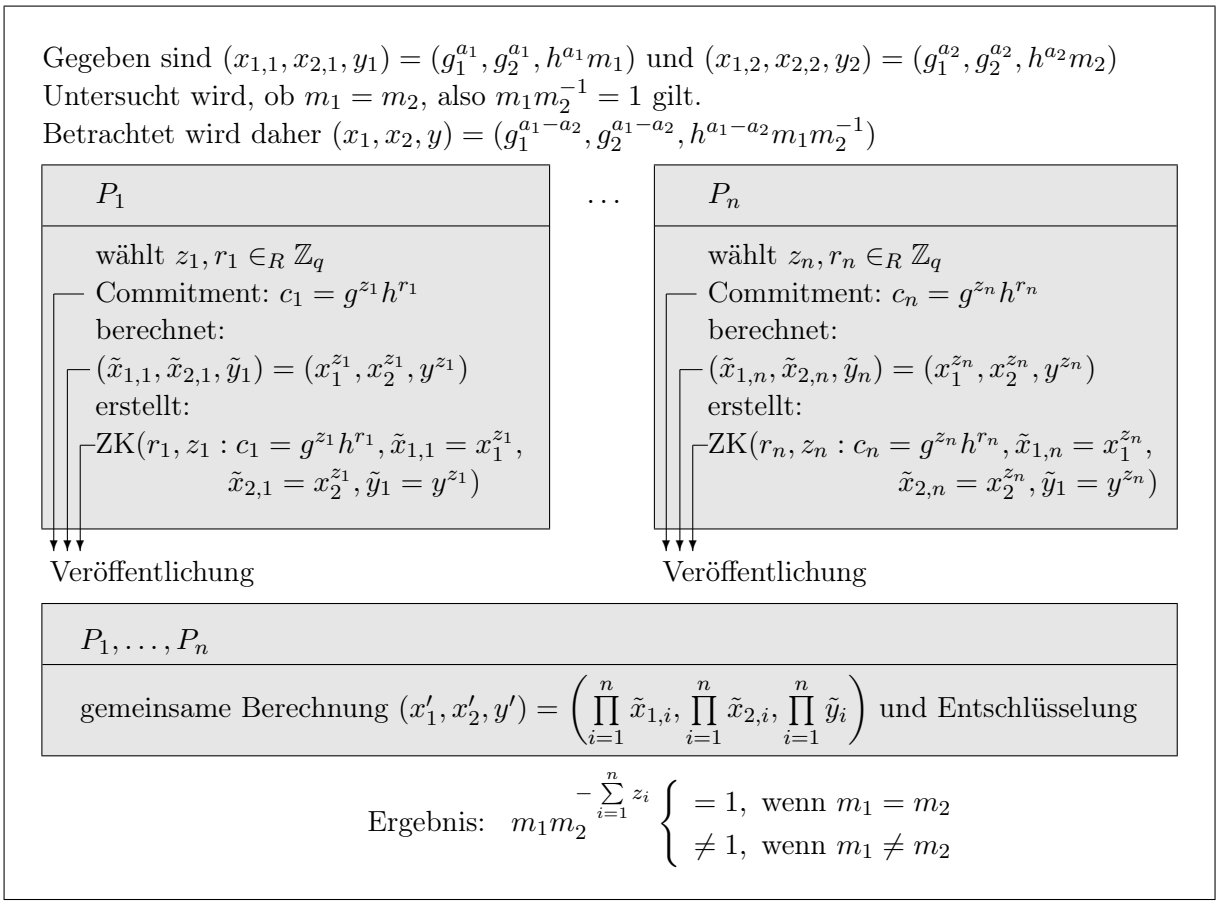

Abbildung 3.10: Test auf Gleichheit zugrundeliegender Klartexte.

Beweis:

(a) Zur Vereinfachung sei $z = \sum_{i=1}^n z_i$.
Wenn sich die Beteiligten korrekt verhalten, ist

$$\begin{aligned}
(x_1', x_2', y') &= \left(\prod_{i=1}^n \tilde{x}_{1,i}, \prod_{i=1}^n \tilde{x}_{2,i}, \prod_{i=1}^n \tilde{y}_i\right) \\
&= \left(\prod_{i=1}^n x_1^{z_i}, \prod_{i=1}^n x_2^{z_i}, \prod_{i=1}^n y^{z_i}\right) \\
&= (x_1^z, x_2^z, y^z) \\
&= \left((x_{1,1} x_{1,2}^{-1})^z, (x_{2,1} x_{2,2}^{-1})^z, (y_1 y_2^{-1})^z\right) \\
&= \left(\left(g_1^{a_1-a_2}\right)^z, \left(g_2^{a_1-a_2}\right)^z, \left(h^{a_1-a_2} \cdot m_1 m_2^{-1}\right)^z\right) \\
&= \left(g_1^{(a_1-a_2)\cdot z}, g_2^{(a_1-a_2)\cdot z}, h^{(a_1-a_2)\cdot z} \cdot (m_1 m_2^{-1})^z\right).
\end{aligned}$$

Dies ist eine modifizierte ElGamal-Verschlüsselung der Nachricht $(m_1 m_2^{-1})^z$ unter dem öffentlichen Schlüssel h mit der Zufallszahl $(a_1 - a_2) \cdot z$. Der Geheimtext lässt sich verteilt entschlüsseln, wie es in Abschnitt 3.7.6 gezeigt ist. Man erhält den Klartext $(m_1 m_2^{-1})^z$. Der Exponent z kann mit Wahrscheinlichkeit $\frac{1}{q}$ ein Vielfaches der Gruppenordnung q sein. Dann hat der Klartext unabhängig von m_1 und m_2 den Wert 1.

Mit Wahrscheinlichkeit $1 - \frac{1}{q}$ ist der Exponent aber kein Vielfaches der Gruppenordnung. Dann kann der Klartext genau dann den Wert 1 annehmen, wenn $m_1 = m_2$ gilt.

(b) Die Entschlüsselung von (x', y') kann nur von einer Gruppe, bestehend aus mindestens t der Beteiligten, in diesem (t, n)-Schwellenschema durchgeführt werden.
Diese Entschlüsselung ist durch Zero-Knowledge-Beweise der korrekten Entschlüsselung global verifizierbar und robust, solange höchstens $n - t$ Beteiligte korrupt sind (vgl. Abschnitt 3.7.4).
Die Berechnung der Werte $(x'_1 = \prod_{i=1}^{n} \tilde{x}_{1,i}, x'_2 = \prod_{i=1}^{n} \tilde{x}_{2,i}, y' = \prod_{i=1}^{n} \tilde{y}_i)$ kann von jedem durchgeführt und überprüft werden.
Die veröffentlichten Werte $\tilde{x}_{1,i}$, $\tilde{x}_{2,i}$ und $\tilde{y}_i$ sind nach Abschnitt 3.2 korrekt berechnet.
Ein Angreifer hat also lediglich die Möglichkeit die Werte z_i so zu wählen, dass die entschlüsselte Nachricht $(m_1 m_2^{-1})^z$ den Wert 1 annimmt, obwohl $m_1 \neq m_2$ ist oder umgekehrt.
Ein Angreifer kann somit zwei mögliche Angriffsziele haben.

- Er kann versuchen die Werte z_i zu Beginn so zu wählen, dass zwei Geheimtexte, die die gleichen zugrundeliegenden Klartexte beinhalten, nicht als solche erkannt werden, d. h. dass die Entschlüsselung von (x'_1, x'_2, y') einen Wert ungleich 1 ergibt, obwohl $m_1 = m_2$ gilt.
  Ist $m_1 = m_2$, so folgt $m_1 m_2^{-1} = 1$ und $(m_1 m_2^{-1})^z = 1$ für alle möglichen Werte z_i.
  Ein Angreifer hat also nicht die Möglichkeit, diesen Angriff durchzuführen.
- Ein Angreifer könnte umgekehrt versuchen, die Werte z_i zu Beginn so zu wählen, dass die Entschlüsselung von (x'_1, x'_2, y') den Wert 1 ergibt, obwohl $m_1 \neq m_2$ gilt.
  Da aus $m_1 \neq m_2$ auch $m_1 m_2^{-1} \neq 1$ folgt, muss der Exponent für diesen Angriff ein Vielfaches der Gruppenordnung sein. Der Angreifer könnte also versuchen, die z_i so zu wählen, dass z ein Vielfaches der Gruppenordnung ist.
  Solange mindestens ein ehrlicher Mitspieler sein gewähltes z_i nicht preisgibt, kann der Angreifer die anderen Werte nicht entsprechend anpassen, da der Wert z_i durch c_i statistisch sicher verborgen wird (vgl. Satz 3.10).

(c) Es ist zu zeigen, dass unter der Diffie-Hellman-Entscheidungsannahme niemand mehr über die Klartexte erfährt, als ob $m_1 = m_2$ gilt oder nicht. Wir betrachten daher einen Simulator, der lediglich weiß, ob $m_1 = m_2$ gilt.

Der Simulator wählt ein $1 \neq u \in_R \mathbb{Z}_q$, falls $m_1 \neq m_2$ gilt. Andernfalls setzt er $u = 1$.
Er generiert Zufallszahlen $z_i, r_i \in_R \mathbb{Z}_q$ und veröffentlicht $c_i = g^{z_i} h^{r_i}$, $\tilde{x}_{1,i} = g_1^{z_i}$, $\tilde{x}_{2,i} = g_2^{z_i}$ und $\tilde{y}_i = h^{z_i} u$ (für $1 \leq i \leq n$).
Dann „beweist" er für jedes $1 \leq i \leq n$, dass er z_i, r_i kennt, so dass $c_i = g^{z_i} h^{r_i}$, $\tilde{x}_{1,i} = x_1^{z_i}$, $\tilde{x}_{2,i} = x_2^{z_i}$ und $\tilde{y}_i = y^{z_i}$ gelten. Da das nicht zutrifft, muss er diesen Beweis simulieren (vgl. Satz 3.2):

- Der Simulator wählt einen Wert $c' \in_R \{0, 1\}$ und Werte $r, r' \in_R \mathbb{Z}_q$. Er berechnet zunächst die Werte $b_{1,1} = x_1^r \tilde{x}_1^{-c'}$, $b_{2,1} = x_2^r \tilde{x}_2^{-c'}$, $b_2 = y^r \tilde{y}^{-c'}$, $b_3 = g^r h^{r'} d^{-c'}$ und sendet diese an den Verifier.

- Der Verifier wählt $c \in_R \{0,1\}$ und sendet das Bit an den Simulator.
- Der Simulator vergleicht c' mit c.
  - Ist $c' = c$, so sendet er r, r' an den Verifier.
    Der Verifier stellt fest, dass die Verifikationsgleichungen

$$\begin{array}{lcccl} x_1^r & = & x_1^r g_1^{z_i \cdot (-c)} (g_1^{z_i})^c & = & b_{1,1} \tilde{x}_1^c \\ x_2^r & = & x_2^r g_2^{z_i \cdot (-c)} (g_2^{z_i})^c & = & b_{2,1} \tilde{x}_2^c \\ y^r & = & y^r (h^{z_i} u)^{-c} (h^{z_i} u)^c & = & b_2 \tilde{y}^c \\ g^r h^{r'} & = & g^r h^{r'} (g^{z_i} h^{r_i})^{-c} (g^{z_i} h^{r_i})^c & = & b_3 \cdot d^c \end{array}$$

    erfüllt sind, und die Simulation war erfolgreich.
  - Ist jedoch $c' \neq c$, so wird die Runde gelöscht.

Nun berechnet der Simulator

$$\left(\prod_{i=1}^{n} \tilde{x}_{1,i}, \prod_{i=1}^{n} \tilde{x}_{2,i}, \prod_{i=1}^{n} \tilde{y}_i \right) = \left(\prod_{i=1}^{n} g_1^{z_i}, \prod_{i=1}^{n} g_2^{z_i}, \prod_{i=1}^{n} (h^{z_i} r) \right) = (g_1^z, g_2^z, h^z r^n) \,.$$

Die Entschlüsselung, die der Simulator nicht durchführen kann, liefert r^n. Falls $m_1 = m_2$ ist, ist der Klartext der Wert 1. Wenn $m_1 \neq m_2$ gilt (und wenn n kein Vielfaches der Gruppenordnung ist) ist der Klartext ein zufälliger Wert $r^n \neq 1$.
Da der Simulator dieses Ergebnis aber schon kennt, kann er es ohne Entschlüsselung ausgeben.

Die Werte z_i, r_i sind in der Simulation und auch in der normalen Protokolldurchführung unabhängig und gleichverteilt aus $\mathbb{Z}_q$.
Somit sind z_i, r_i, c_i in Simulation und normaler Protokollausführung statistisch ununterscheidbar[13].
Die Tripel $(g_1^{z_i}, g_2^{z_i}, h^{z_i} u)$ aus der Simulation und $\left(g_1^{(a_1 - a_2) z_i}, g_2^{(a_1 - a_2) z_i}, (h^{a_1 - a_2} m_1 m_2^{-1})^{z_i} \right)$ sind unter der Diffie-Hellman-Entscheidungsannahme rechnerisch ununterscheidbar.
Die weiteren Berechnungen verlaufen in Simulation und normaler Protokollausführung identisch, und die Verteilung der Ausgabewerte des Simulators stimmt mit der Verteilung der im normalen Protokoll durch Entschlüsselung berechneten Werte überein.
Das bedeutet, dass die Protokollmitschriften von Simulation und normaler Ausführung unter der Diffie-Hellman-Entscheidungsannahme rechnerisch ununterscheidbar sind. Die Durchführung des Protokolls offenbart also lediglich, ob zwei Geheimtexte die gleichen zugrundeliegenden Klartexte enthalten.

□

[13] Man nennt zwei Ensembles $X = \{X_n\}_{n \in \mathbb{N}}$ und $Y = \{Y_n\}_{n \in \mathbb{N}}$ von Zufallsvariablen *statistisch ununterscheidbar*, wenn die statistische Differenz $\Delta = \frac{1}{2} \cdot \sum_{\alpha} |P(X_n = \alpha) - P(Y_n = \alpha)|$ vernachlässigbar ist.

3.10 Verteilte Entschlüsselung und deterministische Blendung

Das in Abschnitt 3.9 beschriebene Protokoll zur Prüfung auf Gleichheit von Klartexten muss für die Zwecke der Wahlsysteme, wie sie in Abschnitt 6 oder bei Juels et al. in [JCJ05] erläutert werden, für jedes Paar zweier abgegebener Stimmen durchgeführt werden. Effizienter ist der von Smith in [Smi05] vorgestellte Ansatz, die Geheimtexte so zu entschlüsseln und gleichzeitig zu blenden, dass aus der probabilistischen Verschlüsselung eine deterministische Blendung wird. Wendet man nun noch eine Hashfunktion auf diese deterministischen Werte an, kann man in einer Hashtabelle einfach paarweise die Gleichheit der Hashwerte überprüfen. Das folgende Protokoll stellt eine Vereinfachung und Anpassung auf das modifizierte ElGamal-Verschlüsselungsschema der in [Smi05] vorgestellten Methode der verteilten Entschlüsselung und gleichzeitigen deterministischen Blendung dar. Es sei G eine Gruppe mit Primzahlordnung q und g_1, g_2 und g Generatoren von G. Die geheimen Schlüssel s_1, s_2 sowie ein weiterer geheimer Wert $\hat{s}$ sind in einem $(t, n_{A'})-$Schwellenschema (vgl. Abschnitte 3.7.6 und 3.7.5) auf die Autoritäten $A'_1, \ldots, A'_{n_{A'}}$ aufgeteilt. Jede Autorität A'_j verfügt somit über einen Anteil $s_{1,j}$ an s_1, einen Anteil $s_{2,j}$ an s_2 und einen Anteil $\hat{s}_j$ an $\hat{s}$ und ist auf diese Anteile über $h_{1,j} = g_1^{s_{1,j}}$, $h_{2,j} = g_2^{s_{2,j}}$ bzw. $\hat{h}_j = g^{\hat{s}_j}$ öffentlich festgelegt. Der zu s_1, s_2 gehörende öffentliche Schlüssel sei $h = g_1^{s_1} g_2^{s_2}$.

Es seien $(x_{1,1}, x_{2,1}, y_1) = (g_1^{r_1}, g_2^{r_1}, h^{r_1}\sigma_1), \ldots, (x_{1,n}, x_{2,n}, y_n) = (g_1^{r_n}, g_2^{r_n}, h^{r_n}\sigma_n)$ probabilistisch unter der modifizierten ElGamal-Verschlüsselung chiffrierte Nachrichten σ_i. Anstelle der schrittweisen Entschlüsselung durch t der Autoritäten werden die folgenden Schritte für jede Nachricht $(x_{1,i}, x_{2,i}, y_i)$ $(1 \leq i \leq n)$ durchgeführt.

1. Jede Autorität A'_j veröffentlicht $\hat{x}_{1,i,j} := x_{1,i}^{\hat{s}_j}$, $\hat{x}_{2,i,j} := x_{2,i}^{\hat{s}_j}$ und $\hat{y}_{i,j} := y_i^{\hat{s}_j}$ und zeigt in einem Zero-Knowledge-Beweis, dass $\log_{x_{1,i}}(\hat{x}_{1,i,j}) = \log_{x_{2,i}}(\hat{x}_{2,i,j}) = \log_{y_i}(\hat{y}_{i,j}) = \log_g(\hat{h}_j)$ gilt.

2. Es sei J eine Indexmenge der t Autoritäten, die den Zero-Knowledge-Beweis bestehen. Jeder kann nun $\hat{x}_{1,i} := \prod_{j \in J} \hat{x}_{1,i,j}^{\ell_j(0)}$, $\hat{x}_{2,i} := \prod_{j \in J} \hat{x}_{2,i,j}^{\ell_j(0)}$ und $\hat{y}_i := \prod_{j \in J} \hat{y}_{i,j}^{\ell_j(0)}$ berechnen. Dabei sind $\ell_j(0) = \prod_{k \in J \setminus \{j\}} k(k-j)^{-1}$ die Lagrange-Faktoren an der Stelle 0 (vgl. Abschnitt 3.7.4).

3. Jede Autorität berechnet nun $\hat{x}'_{1,i,j} := \hat{x}_{1,i}^{s_{1,j}}$ und $\hat{x}'_{2,i,j} := \hat{x}_{2,i}^{s_{2,j}}$ und zeigt in zwei Zero-Knowledge-Beweisen, dass $\log_{\hat{x}_{1,i}}(\hat{x}'_{1,i,j}) = \log_{g_1}(h_{1,j})$ und $\log_{\hat{x}_{2,i}}(\hat{x}'_{2,i,j}) = \log_{g_2}(h_{2,j})$ gelten.

4. Es sei J' eine Indexmenge der t Autoritäten, die den Zero-Knowledge-Beweis bestehen. Jeder kann nun den deterministisch geblendeten Wert berechnen:

$$\sigma_i^{\hat{s}} = \left(\prod_{j \in J'} \hat{x'}_{1,i,j}^{-\ell_j(0)} \right) \cdot \left(\prod_{j \in J'} \hat{x'}_{2,i,j}^{-\ell_j(0)} \right) \cdot \hat{y}_i,$$

denn es gilt:

$$
\begin{aligned}
\left(\prod_{j\in J'} \hat{x'}_{i,j}^{-\ell_j(0)}\right)\cdot \hat{y}_i &= \left(\prod_{j\in J'} \hat{x}_{1,i}^{-s_{1,j}\ell_j(0)}\right)\cdot\left(\prod_{j\in J'} \hat{x}_{2,i}^{-s_{2,j}\ell_j(0)}\right)\cdot\left(\prod_{j\in J} \hat{y}_{i,j}^{\ell_j(0)}\right)\\
&= \left(\hat{x}_{1,i}^{-\sum\limits_{j\in J'} s_{1,j}\ell_j(0)}\right)\cdot\left(\hat{x}_{2,i}^{-\sum\limits_{j\in J'} s_{2,j}\ell_j(0)}\right)\cdot\left(\prod_{j\in J} y_i^{\hat{s}_j\ell_j(0)}\right)\\
&= \left(\hat{x}_{1,i}^{-s_1}\right)\cdot\left(\hat{x}_{2,i}^{-s_2}\right)\cdot\left(y_i^{\sum\limits_{j\in J}\hat{s}_j\ell_j(0)}\right)\\
&= \left(\prod_{j\in J}\hat{x}_{1,i,j}^{\ell_j(0)}\right)^{-s_1}\cdot\left(\prod_{j\in J}\hat{x}_{2,i,j}^{\ell_j(0)}\right)^{-s_2}\cdot y_i^{\hat{s}}\\
&= \left(\prod_{j\in J} x_{1,i}^{\hat{s}_j\ell_j(0)}\right)^{-s_1}\cdot\left(\prod_{j\in J} x_{2,i}^{\hat{s}_j\ell_j(0)}\right)^{-s_2}\cdot(h^{r_i}\sigma_i)^{\hat{s}}\\
&= \left(x_{1,i}^{\sum\limits_{j\in J}\hat{s}_j\ell_j(0)}\right)^{-s_1}\cdot\left(x_{2,i}^{\sum\limits_{j\in J}\hat{s}_j\ell_j(0)}\right)^{-s_2}\cdot(h^{r_i})^{\hat{s}}\,\sigma_i^{\hat{s}}\\
&= \left(x_{1,i}^{\hat{s}}\right)^{-s_1}\cdot\left(x_{2,i}^{\hat{s}}\right)^{-s_2}\cdot(g_1^{s_1}g_2^{s_2})^{r_i\hat{s}}\,\sigma_i^{\hat{s}}\\
&= (g_1^{r_i})^{-\hat{s}s_1}\cdot(g_2^{r_i})^{-\hat{s}s_2}\cdot(g_1^{s_1}g_2^{s_2})^{r_i\hat{s}}\,\sigma_i^{\hat{s}}\\
&= g_1^{-r_i\hat{s}s_1}\cdot g_2^{-r_i\hat{s}s_2}\cdot g_1^{r_i\hat{s}s_1}\cdot g_2^{r_i\hat{s}s_2}\sigma_i^{\hat{s}}\\
&= g_1^{-r_i\hat{s}s_1+r_i\hat{s}s_1}\cdot g_2^{-r_i\hat{s}s_2+r_i\hat{s}s_2}\cdot\sigma_i^{\hat{s}}\\
&= \sigma_i^{\hat{s}}.
\end{aligned}
$$

Die berechneten, deterministisch geblendeten Nachrichten können nun beispielsweise anhand einer Hashtabelle auf Duplikate geprüft werden.

The well being of democracies regardless
of their type and status is dependent
on one small technical detail:
The right to vote.
Everything else is secondary.

Jose Ortega y Gasset

Kapitel 4

Anforderungen an elektronische Wahlsysteme und Sicherheitsziele

Elektronische Wahlen gewinnen immer mehr an Bedeutung. In der Schweiz wurde ein elektronisches Wahlsystem sogar bereits in einem nationalen Referendum eingesetzt (siehe [BB06]).
Bei den Parlamentswahlen in Estland im Herbst 2005 konnten die Wähler landesweit ihre Stimmen verbindlich über ein elektronisches Wahlsystem abgeben (vgl. [MM06]). Allerdings waren bis 2005 noch nicht alle in diesem Kapitel beschriebenen Sicherheitsziele demokratischer Wahlen bei elektronischen Wahlsystemen verwirklicht. Die Eigenschaften elektronischer Medien bieten gegenüber Urnenwahlen beispielsweise eine schnellere Stimmauszählung, ermöglichen es aber auch, ohne großen Aufwand die Aktionen vieler Wähler zu überwachen. Daher müssen an elektronische Wahlen weitere Anforderungen gestellt werden. In diesem Kapitel wird beschrieben, welche Anforderungen an elektronische demokratischen Wahlen zu stellen sind. Die wichtigsten Sicherheitsziele der Unüberprüfbarkeit bzw. Erpressungsresistenz, Verifizierbarkeit und Korrektheit werden im nächsten Abschnitt formal definiert.

4.1 Anforderungen

In den Wahlrechtsgrundsätzen des Artikels 38 des Grundgesetzes ist festgelegt, wie eine demokratische Wahl in Deutschland durchgeführt werden muss.

Das Wahlrecht steht mit Vollendung des 18. Lebensjahres jeder Staatsbürgerin und jedem Staatsbürger zu, der weder entmündigt ist noch seine bürgerlichen Ehrenrechte durch ein Gerichtsurteil verloren hat. Es ist also notwendig, dass der Wahlleiter vor der Wahl überprüft, wer wahlberechtigt ist.

Zur Wahl werden die Wahlberechtigten in Wählerlisten ihres Wahlbezirkes eingetragen oder sie erhalten auf Antrag einen Wahlschein, der Briefwahl ermöglicht. Die Briefwahl wird mit der ansteigenden Mobilität des Bürgers immer stärker in Anspruch genommen.

Bei elektronischen Wahlen muss zur Authentifikation der abgegebenen Stimme diese durch den Wähler signiert oder auf anderem Weg, z. B. durch Hinzufügen eines Credentials, einem Wahlberechtigten zugeordnet werden können. In jedem Fall muss vor der eigentlichen Wahl in einer Registrierungsphase der Wähler authentisch mit der Registrierungsbehörde kommunizieren.

Dies kann z. B. geschehen, indem der Wähler vor der Wahl das Wahlamt seiner Gemeinde aufsucht und dort sein digitaler Signaturschlüssel zertifiziert wird, er ein Credential erhält oder ein personalisierter Observer vom Wahlamt zur Verfügung gestellt wird.

In der Wahlphase geben die Wähler ihre Stimme ab. Diese Stimmabgabe unterscheidet sich bei den einzelnen Verfahren bereits im Ansatz. Allen Verfahren ist lediglich gemeinsam, dass die Stimmabgabe so gestaltet sein muss, dass selbst bei einem gewissen Anteil an korrupten Wählern und korrupten Personen im Wahlausschuss[1] die im Folgenden beschriebenen Wahlrechtsgrundsätze eingehalten werden.
Die minimale Anforderung, die in diesem Zusammenhang an ein Wahlsystem zu stellen ist, ist ein anonymer Kanal vom Wähler zur Wahlbehörde. Wenn ein Angreifer überwachen kann, wer eine Stimme abgibt, kann er Wähler erpressen, der Wahl fernzubleiben. Diese Angriffsmöglichkeit weisen alle bis 2005 bekannten Wahlsysteme auf, die versuchen, ohne einen solchen anonymen Kanal auszukommen, unter anderem das von Hirt und Sako 2000 vorgestellte System [HS00].

Einen solchen anonymen Kanal kann man in der Praxis durch den Einsatz eines asynchronen MIX-Netzes oder durch öffentlich zugängliche Wahlterminals (Wahlkiosk) erreichen.

4.1.1 Ehrlichkeit, Robustheit

Ein unehrlicher Wähler darf nicht die Möglichkeit haben, eine ungültige Stimme so abzugeben dass das Wahlergebnis ungültig wird und die Wahl wiederholt werden muss. Auch darf er die Wahl nicht stören unterbrechen können.

Durch die Benutzung elektronischer Medien zur Stimmabgabe bieten sich jedoch erheblich mehr Möglichkeiten, mit geringem Aufwand den Wahlverlauf in hohem Ausmaß zu stören oder gar die Durchführung der Wahl teilweise oder vollständig zu verhindern.

Es ist daher wichtig, dass die in den elektronischen Wahlverfahren verwendeten kryptografischen Protokolle Störversuche erkennen und es ermöglichen, die beteiligten Instanzen effizient zu identifizieren und ihren Einfluss auf die Wahl auszuschließen. Erfüllt ein Protokoll diese Forderung, so nennt man es *robust*.

4.1.2 Fälschungssicherheit

Eine von einem Wähler abgegebene Stimme muss authentisch sein. Es dürfen nur die wirklich von einem Wähler stammenden Stimmen gewertet werden. Daher muss sichergestellt sein, dass keiner die Stimme eines anderen Wählers fälschen kann. Nach der Abgabe einer Stimme darf diese von keinem mehr geändert werden. Auch der Wähler selbst darf, um Beeinflussung zu verhindern, im Nachhinein seine eigene Stimme nicht mehr revidieren oder abändern. Dies entspricht bei Urnenwahlen dem Einwurf des Stimmzettels in eine Wahlurne.

4.1.3 Unabhängigkeit

Kein Wähler darf z. B. durch Werbung innerhalb des Wahllokals oder durch die Bekanntgabe von Zwischenergebnissen beeinflusst werden.

[1] Ein korrupter Wähler bzw. eine korrupte Person im Wahlausschuss arbeitet mit einem Angreifer zusammen.

Bei elektronischen Wahlen müssen besondere Vorkehrungen getroffen werden, dass niemand die Stimme eines anderen Wählers kopieren oder eine Stimme abgeben kann, die in einem Zusammenhang mit der Stimmabgabe eines anderen Wählers steht (siehe „non-malleability" - Abschnitt 2.3).

4.1.4 Wahlaufwand

Es ist ein erheblicher Aufwand nötig, eine Wahl nach den durch die Wahlrechtsgrundsätze gegebenen Maßstäben auszurichten. Ein elektronisches Wahlprotokoll sollte daher so effizient wie möglich sein.

Damit jeder Wahlberechtigte von seinem Wahlrecht Gebrauch machen kann, ist es erforderlich, dass der Aufwand zur Stimmabgabe auf Seiten des Wählers in einem von ihm zu bewältigenden Rahmen liegt. Es ist sinnvoll, den Aufwand auf der Wählerseite so weit wie möglich zu reduzieren, damit das Ergebnis einer solchen Wahl auch wirklich eine Mehrheitsentscheidung darstellt und nicht durch Bequemlichkeit einer Wählergruppe oder durch ungültige Stimmen, die aufgrund von unüberschaubaren Anforderungen an den Wähler entstehen können, verfälscht wird.

4.1.5 Geheime Wahl und Einmaligkeit

Jede abgegebene Stimme hat das gleiche Gewicht für die Auszählung. Das Stimmengewicht der Wahlberechtigten (Zählwertgleichheit der Stimme) darf nicht abhängig gemacht werden von Besitz, Einkommen, Steuerleistung, Bildung, Religion, Rasse, Geschlecht oder politischer Einstellung. Daraus resultiert, dass jeder Wähler seine Stimme nur einmal abgeben darf. Dies nennt man *Einmaligkeit*. Die Identität der Wähler muss also vor jeder Stimmabgabe zusammen mit der Wahlberechtigung überprüft werden. Eine erfolgte Stimmabgabe wird im Wahlverzeichnis vermerkt, um mehrfache Stimmabgaben zu verhindern.

4.1.6 Geheime Wahl und Anonymität

Man nennt eine Wahl *geheim*, wenn niemand herausfinden kann, was ein anderer gewählt hat. Allenfalls darf der Wähler selbst bekannt geben, wem er seine Stimme gegeben hat. Jeder Wähler muss seine Stimme persönlich abgeben, daher muss die Wahlberechtigung überprüft werden. Es darf jedoch nicht möglich sein, eine abgegebene Stimme einem bestimmten Wähler zuzuordnen. Die Anonymität einer geheimen Wahl sichert die freie Wahlentscheidung der Wahlberechtigten.

Wahlgeheimnis und Wahlfreiheit wurden in Deutschland erst 1903 mit der Einführung von Wahlumschlägen und Wahlkabinen garantiert und 1913 durch eine verbindliche Regelung hinsichtlich der Beschaffenheit von Wahlurnen ergänzt.

Bei elektronischen Wahlen kommt der Anonymität eine besondere Bedeutung zu, da es bei elektronischen Medien wesentlich einfacher ist, eine große Anzahl von Wählern bezüglich ihres Wahlverhaltens zu überwachen, wenn das Wahlsystem nicht gegen solche passiven Angriffe gesichert ist.

4.1.7 Unüberprüfbarkeit, Erpressungsresistenz

Es darf auf Wählerinnen und Wähler von keiner Seite irgendein Druck ausgeübt werden, zugunsten des einen oder anderen Kandidaten abzustimmen oder sich der Wahl zu enthalten. Diese Unabhängigkeit der Stimmabgabe darf nicht durch Beeinflussung des Wählers gefährdet werden. So ist beispielsweise durch das Bundeswahlgesetz während der Wahlzeit in und an dem Gebäude, in dem sich der Wahlraum befindet, sowie unmittelbar vor dem Eingang des Gebäudes jede Beeinflussung der Wähler durch Wort, Ton, Schrift oder Bild sowie jede Unterschriftensammlung verboten. Darüber hinaus ist die Veröffentlichung von Ergebnissen von Wählerbefragungen nach der Stimmabgabe über den Inhalt der Wahlentscheidung vor Ablauf der Wahlzeit unzulässig.

Der Grundsatz einer freien Wahl impliziert auch, dass eine Beeinflussung durch Bestechung oder Erpressung nicht ermöglicht werden darf. Das wird nicht allein durch den Grundsatz einer geheimen Wahl erfüllt. Bei der Benutzung von elektronischen Medien ist es oft einfach, gesendete oder empfangene Daten aufzuzeichnen und sich so einen Beleg über die Stimme zu generieren. Einen Beleg erstellen zu können, ist keinesfalls ein erwünschtes, zusätzliches Merkmal. Durch einen solchen Beleg kann ein Wähler gegenüber Dritten nachweisen, wie er gewählt hat. Er wird erpressbar oder bestechlich. Daher muss diese Möglichkeit ausgeschlossen werden, um Stimmenkauf oder Erpressung zu verhindern. Der Wähler darf selbst bekannt geben, wem er seine Stimme gegeben hat. Allerdings darf er keine Gelegenheit haben, gegenüber Dritten zu belegen, wie er gewählt hat. Diese Eigenschaft wird in bisherigen Arbeiten *Unüberprüfbarkeit* (receipt-freeness) genannt. Wird sie erfüllt, kann der Wähler seine tatsächliche Wahl gegenüber einem Erpresser oder Stimmenkäufer falsch angeben. Dieser kann die Aussage nicht auf ihren Wahrheitsgehalt hin überprüfen.

Es hat sich kürzlich (siehe [JCJ05]) gezeigt, dass sogar die Unüberprüfbarkeit noch nicht ausreicht, ein Wahlsystem gegen die folgenden realistischen Angriffe abzusichern:

- *Gleichverteilungsangriff* (*Randomisationsangriff*)
  In dem Wahlsystem, das Hirt und Sako im Jahr 2000 vorstellten, sind alle Wahloptionen in zufälliger Reihenfolge verschlüsselt aufgelistet. Nur der Wähler kennt die Position der einzelnen Wahloptionen und gibt seine Stimme ab, indem er den Index der verschlüsselten Wahloption angibt.
  Das Wahlsystem bietet Unüberprüfbarkeit in dem Sinne, dass der Wähler nicht beweisen kann, welche Wahloption sich hinter seiner gewählten Position verbirgt. Jedoch kann ein Erpresser den Wähler zwingen eine von ihm vorgegebene Position der Liste zu wählen und so bei Erpressung vieler Wähler eine Gleichverteilung der Stimmen auf die möglichen Wahloptionen herbeiführen. In einem Gebiet, das als Hochburg einer bestimmten Partei gilt, kann dieser Angriff durchaus sinnvoll sein, da er eine gewollte Veränderung zugunsten der Partei herbeiführt, die in diesem Gebiet eher unterprozentual repräsentiert war.

- *Erzwungene Wahlenthaltung*
  Ein Angreifer kann in einem Wahlsystem, bei der eine Nachricht eindeutig dem Wähler als Absender zugeordnet werden kann, verlangen, dass der Wähler seine Stimme nicht abgibt. Dann kann ein solcher Angriff das Wahlergebnis in eine vom Erpresser gewünschte Richtung

beeinflussen, wenn der Erpresser die Präferenzen des Wähler kennt oder richtig einschätzen kann.

- *Angriff durch Impersonation bzw. Simulation*
  Ein Erpresser kann bereits vor der eigentlichen Wahlphase (im Anschluss an die Registrierung) vom Wähler verlangen, dass ihm dieser seinen geheimen Schlüssel, seinen Observer, sein Credential oder anderes Schlüsselmaterial zur Verfügung stellt.
  Dann kann der Erpresser an Stelle des Wählers seine Stimme abgeben.

Aus diesem Grund haben Juels, Catalano und Jakobsson in [JCJ05] den neuen Begriff der *Erpressungsresistenz* (*coercion-resistance*) eingeführt.

4.1.8 Korrektheit

Nur gültige Stimmen dürfen gezählt werden und so ihren Anteil zum in der Auszählung ermittelten Endergebnis beitragen. Andererseits müssen alle gültigen Stimmen gezählt werden. Es darf nicht vorkommen, dass gültige Stimmen für ungültig oder ungültige Stimmen für gültig erklärt werden, und dass so das Auszählungsergebnis verfälscht wird.

In demokratischen Wahlen wird dies meist durch den Einsatz mehrerer unabhängiger Personen sichergestellt. Die Bedeutung dieser Entscheidungen wurde beispielsweise im November 2000 bei der Präsidentschaftswahl in den USA deutlich, als die Entscheidung nur von einigen tausend Stimmen abhing, deren Gültigkeit durch ungenaue Lochung der Stimmzettel zweifelhaft war.

4.1.9 Verifizierbarkeit

Das Wahlrecht muss von der Bevölkerung als gerecht und sachgemäß empfunden werden. Nur dann hat eine Wahlentscheidung ihre legitimierende Wirkung, und der gewählte Abgeordnete kann im Sinn der vom Grundgesetz geforderten repräsentativen Demokratie die „Herrschaft des Volkes" ausüben. Daher muss gewährleistet sein, dass das Ergebnis der Wahl korrekt ist, wovon sich jeder Teilnehmer selbst überzeugen können sollte. In diesem Fall spricht man von *globaler Verifizierbarkeit*.

Bei Wahlen mit einer großen Anzahl von Wählern ist es oft schwierig für den einzelnen, die Korrektheit der Auszählung zu überprüfen. Kann ein Wähler lediglich überprüfen, ob seine eigene Stimme korrekt gezählt wurde, so spricht man von *lokaler Verifizierbarkeit*.

4.2 Formale Definition der wichtigsten Sicherheitsziele

In diesem Abschnitt werden die insbesondere für die Wahlsysteme in den Kapiteln 5 und 6 wichtigsten Sicherheitsziele formaler definiert, um die Analyse dieser Systeme in den Abschnitten 5.2.4, 5.3.4 und 6.3 zu präzisieren.
An einer elektronischen Wahl sind mehrere Instanzen aktiv oder auch passiv beteiligt:

1. Registrierungsautoritäten: Eine Menge $\mathcal{R} = \{R_1, \ldots, R_{n_R}\}$ von Autoritäten, die für die Registrierung der Wähler verantwortlich sind, stellen gemeinsam die erforderlichen Schlüs-

sel, Credentials bzw. Observer sowie die Liste der Wahlberechtigten und Kandidaten zur Verfügung.

2. Autoritäten: Die Autoritäten, die für die Durchführung der Wahlphase und für die Auszählung der Stimmen zuständig sind, werden hier in der Menge $\mathcal{A} = \{A_1, \ldots, A_{n_A}\}$ bzw. für das Wahlsystem in Kapitel 6 in den Mengen $\mathcal{A} = \{A_1, \ldots, A_{n_A}\}$, $\mathcal{A}' = \{A'_1, \ldots, A'_{n'_A}\}$ und $\tilde{\mathcal{A}} = \{\tilde{A}_1, \ldots, \tilde{A}_{n_{\tilde{A}}}\}$ zusammengefasst.

3. Wähler, Wahlberechtigte: Die Wahlberechtigten $\mathcal{V} = \{V_1, \ldots, V_{n_V}\}$ sind die Personen, die von den Registrierungsautoritäten als wahlberechtigt ermittelt worden sind. Davon zu unterscheiden ist die Menge der Wähler, also der Wahlberechtigten, die eine Stimme abgeben. Wir unterscheiden weiterhin zwischen ehrlichen Wählern $\mathcal{W} = \{W_1, \ldots, W_{n_W}\}$, die ihre Wahlentscheidung unbeeinflusst von Dritten treffen und sich an den vorgesehenen Ablauf der Wahl halten, und korrupten Wählern $\mathcal{U} = \{U_1, \ldots, U_{n_U}\}$, die von einem Angreifer beispielsweise durch Bestechung beeinflusst werden, ihre Stimme entsprechend den Wünschen des Angreifers abzugeben. Natürlich bilden $\mathcal{U}$ und $\mathcal{W}$ eine Partition der Wähler.

4. Angreifer: Der Angreifer X versucht durch Bestechung von Wählern aus $\mathcal{U}$, durch Erpressung eines Wählers aus $\mathcal{W}$ bzw. durch Einflussnahme auf korrupte Autoritäten aus $\mathcal{R}$ bzw. $\mathcal{A}$, $\mathcal{A}'$ oder $\tilde{\mathcal{A}}$ das Wahlergebnis zu beeinflusssen.

5. Schwarzes Brett: In den betrachteten Wahlsystemen wird die Wahlentscheidung des Wählers an ein Schwarzes Brett B geschickt. Ein Schwarzes Brett ist ein Speicher, auf den alle Instanzen Schreibrechte besitzen, um Daten hinzuzufügen. Keine Instanz hat jedoch die Möglichkeit, Daten von diesem Speicher zu löschen, zu ändern oder zu überschreiben. Je nach Wahlsystem haben die Wähler die Möglichkeit, die Daten, die auf dem Schwarzen Brett hinterlegt werden während oder nach der Wahlphase zu lesen.

6. Kandidaten, Wahloptionen: Die zur Wahl stehenden Kandidaten bzw. Wahloptionen werden auf einer Kandidatenliste $\mathbf{L} = (m_1, \ldots, m_{n_L})$ aufgeführt.
   Entsprechend der Anordnung dieser Liste ist das Auszählungsergebnis der $j-$ten Auszählung eine Liste $\mathbf{T_j} = (t_{1,j}, \ldots, t_{n_L,j}, t_{n_L+1,j}, t_{n_L+2,j})$ natürlicher Zahlen. Dabei sei $t_{i,j}$ $(1 \leq i \leq n_L)$ die Anzahl der für die Wahloption m_i abgegebenen Stimmen bei der $j-$ten Auszählung. Der Wert $t_{n_L+1,j}$ steht für die Anzahl der Enthaltungen, also für die Anzahl der Wahlberechtigten, für die keine Stimme registriert wurde. Mit $t_{n_L+2,j}$ wird die Anzahl der abgegebenen, aber als ungültig erkannten Stimmen gezählt.

Ein Wahlsystem WS besteht aus vier Protokollen:

$$\mathsf{WS} = \{\mathsf{Registrierung}, \mathsf{Abstimmung}, \mathsf{Auszaehlung}, \mathsf{Verifikation}\}.$$

Ein erpressungsresistentes Wahlsystem benötigt noch ein weiteres Protokoll:

$$\mathsf{WS} = \{\mathsf{Registrierung}, \mathsf{Abstimmung}, \mathsf{Auszaehlung}, \mathsf{Verifikation}, \mathsf{Schluesselfaelschung}\}.$$

- Registierung: Die Funktion $\mathsf{Registrierung}(SK_{\mathcal{R}}, V_i, k_1) \rightarrow (sk_{V_i}, pk_{V_i})$ gibt unter Eingabe des geheimen Schlüssels $SK_{\mathcal{R}}$ der Registrierungsautoritäten, der Wähler-ID V_i und eines Sicherheitsparameters k_1 ein Schlüsselpaar (sk_{V_i}, pk_{V_i}) aus. Dieses Schlüsselpaar wird gemeinsam von den Registrierungsautoritäten, ggf. in Interaktion mit dem Wähler V_i, berechnet.

- Abstimmung: Die Funktion $\mathsf{Abstimmung}(sk_{V_i}, PK_{\mathcal{A}}, \mathbf{L}, m_i, k_2) \rightarrow \mathit{Stimme}$ generiert die Stimme des Wählers V_i. Als Eingabe für die Funktion wird der geheime Schlüssel sk_{V_i} des Wählers V_i, der öffentliche Schlüssel $PK_{\mathcal{A}}$ der Autoritäten $\mathcal{A}$, die Wahlliste $\mathbf{L}$, die auf dieser Wahlliste ausgesuchte Wahloption m_i und ein Sicherheitsparameter k_2 erwartet.
  Die Gestalt der von Abstimmung ausgegebenen Stimme variiert von Wahlsystem zu Wahlsystem. Allgemein kann man sich unter der Stimme eine Wahloption vorstellen, die unter dem öffentlichen Schlüssel des Wahlausschusses verschlüsselt und mit einer Wahlberechtigung (Signatur, Credential etc.) versehen wurde.

- Auszaehlung: Die Funktion $\mathsf{Auszaehlung}(SK_{\mathcal{A}}, SK_{\mathcal{A}'}, SK_{\tilde{\mathcal{A}}}, B, \mathbf{L}, \{pk_{V_i}\}_{i=1}^{n_V}, k_3) \rightarrow (\mathbf{T}, P_T)$ gibt unter Eingabe der geheimen Schlüssel $SK_{\mathcal{A}}, SK_{\mathcal{A}'}, SK_{\tilde{\mathcal{A}}}$, der Autoritäten, des Inhalts des Schwarzen Brettes B, der Wahloptionsliste $\mathbf{L}$, aller öffentlichen Schlüssel der Wahlberechtigten und eines Sicherheitsparameters k_3 das Auszählungsergebnis $\mathbf{T}$ zusammen mit einem nicht interaktiven Beweis P_T der Korrektheit der Auszählung aus.
  In der Auszählung werden die zu den abgegebenen Stimmen gehörenden Credentials mit der Liste der Wahlberechtigten abgeglichen. In diesem Zusammenhang wird die Anzahl der Wähler ermittelt, die keine Stimme abgegeben haben, sowie die Anzahl der als ungültig erkannten Stimmen festgehalten.

- Verifikation: Die Funktion $\mathsf{Verifikation}(PK_{\mathcal{A}}, PK_{\mathcal{A}'}, PK_{\tilde{\mathcal{A}}}, B, \mathbf{L}, \mathbf{T}, P_T) \rightarrow \{0, 1\}$ gibt unter Eingabe der öffentlichen Schlüssel $PK_{\mathcal{A}}, PK_{\mathcal{A}'}, PK_{\tilde{\mathcal{A}}}$ der Autoritäten, des Inhalts des Schwarzen Brettes B, der Wahloptionsliste $\mathbf{L}$, des Auszählungsergebnisses $\mathbf{T}$ und des nichtinteraktiven Korrektheitsbeweises P_T eine 0 aus, wenn die Korrektheit der Auszählung nicht verifiziert werden kann, und eine 1, wenn das Auszählungsergebnis gültig ist.

Damit ein Wahlsystem erpressungsresistent ist, muss es dem Wähler möglich sein, dem Erpresser zu suggerieren, sich an die Anweisungen zu halten, aber dennoch frei in seiner Wahlentscheidung zu sein. Dazu gehört auch, dass er einer möglichen Forderung des Erpressers nach Bekanntgabe des geheimen Wahlschlüssels scheinbar nachkommen kann, ohne den wirklichen Schlüssel preiszugeben. Daher benötigt man noch eine weitere Funktion im Wahlverfahren:

- Schluesselfaelschung: Die Funktion $\mathsf{Schluesselfaelschung}(PK_{\mathcal{A}}, sk_{V_i}, pk_{V_i}) \rightarrow (\widetilde{sk}_{V_i}, pk_{V_i})$ gibt unter Eingabe des öffentlichen Schlüssels $PK_{\mathcal{A}}$ der Autoritäten und des geheimen und öffentlichen Schlüssels des Wählers V_i einen gefälschten geheimen Schlüssel $\widetilde{sk}_{V_i}$ aus. Dieser Schlüssel muss von einem Erpresser ununterscheidbar zum echten Schlüssel sk_{V_i} sein.

Von den Wahlberechtigten abgegebene, gültige Stimmen müssen genau einmal gezählt werden. Da das Schwarze Brett B ein Speicher ist, auf dem jeder Beteiligte im Wahlsystem Schreibrechte besitzt, um Daten hinzuzufügen, aber keine bereits vorhandenen Daten löschen oder ändern

kann, können Manipulationen nur in der Phase der Stimmabgabe zwischen Wähler und Schwarzem Brett oder in der Phase der Stimmauszählung auftreten. Letztere Betrugsversuche werden durch die Eigenschaft der Verifizierbarkeit unterbunden. Durch die Eigenschaften des Schwarzen Brettes ist es aus kryptografischer Sicht unwichtig, ob die Gültigkeit der Stimme im Rahmen der Korrektheit während der eigentlichen Wahlphase oder im Zusammenhang mit der Verifizierbarkeit zur Auszählungsphase geprüft wird. Da es Wahlsysteme gibt, die zunächst die Nachrichten auf dem Schwarzen Brett verarbeiten, entschlüsseln und dann erst prüfen, ob es sich um gültige Stimmen handelt, wird die Gültigkeitsprüfung zur Auszählungsphase gerechnet. Um die Sicherheit eines elektronischen Wahlprotokolls formal zu definieren, werden im Folgenden die drei Experimente $Korr.$, $Ver.$ und $E.-res.$ festgelegt, die jeweils beschreiben, welche Auswirkungen sich aus Interaktionen zwischen dem Angreifer X und dem Wahlsystem WS ergeben. Das Experiment $E \in \{Korr., Ver., E.-res.\}$ wird dabei jeweils so festgelegt, dass ein erfolgreicher Angriff durch X zu einer Ausgabe des Wertes 1 führt, während bei einem vollkommen sicheren Wahlsystem in allen Experimenten 0 ausgegeben wird. Somit definiert man für ein Experiment $\mathbf{Exp}^{E}_{\mathsf{WS},X}(\cdot)$ den Erfolg eines Angreifers als die Wahrscheinlichkeit, dass das Experiment den Wert 1 liefert: $\mathbf{Succ}^{E}_{\mathsf{WS},X}(\cdot) = Pr\left(\mathbf{Exp}^{E}_{\mathsf{WS},X}(\cdot) = 1\right)$.
Innerhalb der Experimente sei $\leftarrow$ eine Zuweisung und $\Leftarrow$ ein Senden von Daten.

4.2.1 Korrektheit

Betrachten wir zunächst die Wahlphase und die dazu gehörende Eigenschaft der Korrektheit. Im folgenden Experiment werden dem Angreifer Fähigkeiten zugestanden, über die er im Normalfall nicht verfügt. Der Angreifer X kann die Kandidatenliste $\mathbf{L}$ festlegen. Er korrumpiert eine Menge U von Wählern nach der Registrierungsphase. Außerdem kann X im Rahmen des Experiments neben der Kontrolle der korrupten Wähler U auch bestimmen, welche Stimmen die nicht kontrollierten, ehrlichen Wähler W abgeben. Wenn es dem Angreifer selbst unter diesen Bedingungen nicht gelingt, die Nachrichten auf dem Schwarzen Brett und somit das zugehörige Auszählungsergebnis zu verfälschen, ist das Wahlsystem erst recht in einem realistischen Szenario korrekt. Der Angreifer hat lediglich die folgenden drei möglichen Angriffsziele. Er könnte eine Stimme eines ehrlichen Wählers abfangen und somit verhindern, dass diese auf das Schwarze Brett gelangt. Andererseits könnte der Angreifer versuchen, selbst Stimmen im Namen der ehrlichen Wähler ohne deren Mithilfe[2] abzugeben und diesen so zuvorzukommen. Neben dem Löschen oder Hinzufügen von Nachrichten besteht ein mögliches Angriffsziel im Verändern von Nachrichten. In jedem Fall ist es das Ziel des Angreifers, dass die Manipulation unentdeckt bleibt, die Auszählung anschließend also erfolgreich durchgeführt wird.

Definition 4.1 (Korrektheit)
Ein Wahlsystem WS ist korrekt, wenn für alle natürlichen Zahlen n_L, n_V und alle polynomiell beschränkten Angreifer X die Erfolgswahrscheinlichkeit $\mathbf{Succ}^{Korr.}_{\mathsf{WS},X}(k_1, k_2, k_3, n_L, n_V)$ unter den

[2] Der Angriff einer Stimmabgabe im Namen eines Wählers mit dessen Mithilfe wird im Zusammenhang mit der Erpressungsresistenz untersucht.

Sicherheitsparametern k_1, k_2, k_3 vernachlässigbar ist.
Dabei sei das Experiment $\mathbf{Exp}^{Korr.}_{\mathsf{WS},X}(k_1, k_2, k_3, n_L, n_V)$ wie in Abbildung 4.1 auf Seite 78 definiert.

4.2.2 Verifizierbarkeit

Einem Angreifer X könnte es gelingen, eine Anzahl von Autoritäten aus $\mathcal{A}, \mathcal{A}'$ bzw. $\tilde{\mathcal{A}}$ zu korrumpieren. In einem global verifizierbaren Wahlsystem muss jeder Wähler überprüfen können, ob alle Stimmen korrekt in die Auszählung eingeflossen sind.
In der folgenden Sicherheitsdefinition wird vorausgesetzt, dass der Angreifer X *alle* Wähler und eine Minderheit der Autoritäten aus $\mathcal{A}, \mathcal{A}'$ bzw. $\tilde{\mathcal{A}}$ kontrolliert. Eine derart hohe Sicherheitsanforderung ist im Hinblick auf die Bedeutung der Wahlen und der damit verbundenen Legitimation essentiell.
Das Ziel des Angreifers ist es, dass der Beweis der korrekten Auszählung akzeptiert wird und die Verifikation der Auszählung durchgeführt werden kann, die Auszählung aber in Wirklichkeit nicht korrekt ist. Das bedeutet, dass die gezählte Stimme nicht der Wahloption des Wählers entspricht oder eine Stimme mehrfach gezählt wird.

Definition 4.2 (Verifizierbarkeit)
Ein Wahlsystem WS ist verifizierbar, wenn für alle natürlichen Zahlen n_L, n_V und alle polynomiell beschränkten Angreifer X die Erfolgswahrscheinlichkeit $\mathbf{Succ}^{Ver.}_{\mathsf{WS},X}(k_1, k_3, n_L, n_V)$ unter den Sicherheitsparametern k_1, k_3 vernachlässigbar ist.
Dabei sei das Experiment $\mathbf{Exp}^{Ver.}_{\mathsf{WS},X}(k_1, k_3, n_L, n_V)$ wie folgt definiert:

$\mathbf{Exp}^{Ver.}_{\mathsf{WS},X}(k_1, k_3, n_L, n_V)$

$\{(sk_{V_i}, pk_{V_i}) \leftarrow \mathsf{Registrierung}(SK_{\mathcal{A}}, V_i, k_1)\}_{i=1}^{n_V}$	Die Wähler werden registriert.
$(B, \mathbf{T}, P) \leftarrow (SK_{\mathcal{A}}, \{(sk_{V_i}, pk_{V_i})\}_{i=1}^{n_V}$, 'fälsche Stimmen')	X führt die komplette Stimmabgabe und Auszählung durch.
$(\mathbf{T}', P') \leftarrow \mathsf{Auszaehlung}(SK_{\mathcal{A}}, B, \mathbf{L}, \{pk_{V_i}\}_{i=1}^{n_V}, k_3)$	Es wird ausgezählt.
if $(\mathbf{T} \neq \mathbf{T}')$	Weicht die korrekte Auszählung von der des Angreifers ab?
and $(\mathsf{Verifikation}(PK_{\mathcal{A}}, B, \mathbf{L}, \mathbf{T}, P) = 1)$ then	Verifikation erfolgreich?
output 1	Angriff erfolgreich.
else	
output 0	Angriff nicht erfolgreich.

4.2.3 Erpressungsresistenz

Um den Erfolg eines Angreifers X bewerten zu können, vergleichen wir ihn mit einem schwächeren polynomiell beschränkten Angreifer X' in einem perfekten Wahlsystem pWS. Dieses Wahlsystem pWS und dessen Funktionen müssen so nicht existieren. Sie dienen lediglich zum Vergleich eines Wahlsystems mit diesem perfekten System. Wenn ein Angreifer X in einem zu untersuchenden Wahlsystem nicht mehr Erfolg hat, als der Angreifer X' in dem perfekten System, so ist das

betrachtete Wahlsystem sicher. Die Fähigkeiten des Angreifers X' charakterisieren das Maß an Erpressungsresistenz, das im Wahlsystem vorliegen soll.

Der Angreifer X' ist passiv und erhält als Information die in der Auszählung für jeden zugänglichen Werte, das endgültige Auszählungsergebnis und die Anzahl der wegen ungültiger Credentials als ungültig gewerteten, aussortierten Stimmen.
Er kann nichts mit den geheimen Schlüsseln sk_{V_i} ($V_i \in U$) der korrupten Wähler anfangen, also keine aktiven Angriffe durchführen. Außerdem kann X' auch mit den geheimen Schlüsseln der ehrlichen oder erpressten Wähler nichts über deren Wahlverhalten aussagen, da er in diesem perfekten Wahlsystem das Schwarze Brett *nie*, auch nicht nach Beendigung der Wahlphase, zu Gesicht bekommt. X' kann selbst keine Stimmen abgeben, sondern lediglich die Stimmen der korrupten Wähler vorgeben.

Um diesen Anforderungen gerecht zu werden, muss eine neue Funktion zur Auszählung definiert werden.

- perfekte − Auszaehlung: Die Funktion perfekte − Auszaehlung gibt das Auszählungsergebnis $\mathbf{T}$ aus.
  Die Stimmen ehrlicher Wähler, die in pWS vor den Stimmen des Angreifers X' abgegeben werden[3], werden dabei wie in der Funktion Auszaehlung ausgewertet.
  Für die weiteren abgegebenen Stimmen bestimmt die Funktion perfekte − Auszaehlung jeweils das zugrunde liegende Credential sk_{V_i}.
  Wenn sk_{V_i} nicht zu einem korrupten Wähler gehört ($V_i \notin U$), werden die durch X' abgegebenen Stimmen nicht gezählt, d. h. X' kann dadurch nur die Stimmen für die korrupten Wähler vorgeben.
  Die vom Angreifer X' abgegebene Stimme, die auf dem Credential $\widetilde{sk} = sk_{V_j}$ basiert, wird im Fall[4] $b = 0$ nicht gezählt; im Fall $b = 1$ jedoch gewertet.
  Außerdem wird entsprechend der Policy jeweils nur eine Stimme gewertet, wenn Stimmen doppelt abgegeben werden, also auf dem gleichen Credential sk_{V_i} basieren.

Wie genau diese Fähigkeiten der Funktion erreicht werden, muss nicht spezifiziert werden. Es genügt, dieses Wahlsystem und insbesondere die Funktion perfekte − Auszaehlung als Vergleichsmaßstab zu haben, da diese Funktion im Beweis durch einen Simulator durchgeführt wird.

Definition 4.3 (Erpressungsresistenz)
Ein Wahlsystem WS ist erpressungsresistent, wenn für alle natürlichen Zahlen n_L, n_V, für alle Verteilungen D_{n_L,n_V} der Stimmen auf die Wahloptionen und alle polynomiell beschränkten Angreifer X der Wert

$$\mathbf{Erpressung}^{E.-res.}_{\mathsf{WS},X} = |\mathbf{Succ}^{E.-res.}_{\mathsf{WS},X}(k_1, k_2, k_3, n_L, n_V) - \mathbf{Succ}^{p.-E.-res.}_{\mathsf{WS},X'}(k_1, k_2, k_3, n_L, n_V)|$$

[3] Wenn die ehrlichen Wähler ihre Stimmen vor dem Angreifer X' abgeben müssen, kann der Angreifer seine Stimmen adaptiv erstellen und ist somit potentiell stärker.

[4] In den Experimenten wird ein Bit b gewählt, das anzeigt, ob der Wähler V_j der Erpressung des Angreifers nachkommt (Fall $b = 1$) oder widersteht (Fall $b = 0$).

unter den Sicherheitsparametern k_1, k_2, k_3 vernachlässigbar ist.
Dabei seien die Experimente $\mathbf{Exp}^{E.-res}_{\mathsf{WS},X}(k_1, k_2, k_3, n_L, n_V)$ und $\mathbf{Exp}^{p.-E.-res}_{\mathsf{WS},X}(k_1, k_2, k_3, n_L, n_V)$ wie in den Abbildungen 4.2 und 4.3 auf den Seiten 79 und 80 definiert.

Anmerkungen:

Die obige formale Definition der Erpressungsresistenz setzt voraus, dass die Verteilung D_{n_L,n_V} der Stimmen auf die Wahloptionen der Gestalt ist, dass der Angreifer anhand dieser Verteilung über das Verhalten der ehrlichen Wähler keine Aussage treffen kann. In wenigen Spezialfällen kann es dem Angreifer in einem System, das der obigen Definition der Erpressungsresistenz genügt, dennoch möglich sein, über das Wahlverhalten der ehrlichen Wähler Aussagen zu treffen. Beispielsweise kann ein Angreifer, der von ehrlichen Wählern verlangt, für Partei A zu stimmen, herausfinden, dass der Wähler seiner Erpressung oder Bestechung nicht nachgegeben hat, wenn im Auszählungsergebnis keine Stimme für Partei A enthalten ist. In der Realität werden solche Spezialfälle kaum auftreten, so dass ein Angreifer nur sehr begrenzte Aussagen über das Wahlverhalten der ehrlichen Wähler treffen kann. Je mehr Entropie[5] die Verteilung D_{n_L,n_V} enthält, desto geringer sind die Chancen des Angreifers, Aussagen über die Stimmen ehrlicher Wähler zu treffen.

[5] Die *Entropie* H ist nach Shannon ein Maß für den *mittleren Informationsgehalt* einer Nachricht, unabhängig von deren Codierung. Es sei p_i die Wahrscheinlichkeit mit der das i-te Zeichen auftritt. Dann ist die Entropie wie folgt definiert: $H := \sum_i p_i \cdot \log_2\left(\frac{1}{p_i}\right)$.

$\mathbf{Exp}^{Korr.}_{\mathsf{WS},X}(k_1, k_2, k_3, n_L, n_V)$

$\{(sk_{V_i}, pk_{V_i}) \leftarrow \mathsf{Registrierung}(SK_{\mathcal{R}}, V_i, k_1)\}_{i=1}^{n_V}$	Die Wähler werden registriert.
$U \leftarrow X(\{pk_{V_i}\}_{i=1}^{n_V},$ 'erstelle Menge kontrollierter Wähler')	X korrumpiert Wähler U.
$\{v_i\}_{V_i \in W} \leftarrow X$('lege Stimmen v_i für ehrliche Wähler $V_i \in W$ fest')	X legt Stimmen der ehrlichen Wähler fest.
$B \Leftarrow \{$Stimme $\leftarrow X($ 'versuche, Stimme für $V_i \in W$ abzugeben')$\}$	X versucht, Stimmen ehrlicher Wähler abzugeben.
$(\mathbf{T_1}, P_{T,1}) \leftarrow \mathsf{Auszaehlung}(SK_{\mathcal{A}}, B, \mathbf{L}, \{pk_{V_i}\}_{i=1}^{n_V}, k_3)$	Auszählung der bisherigen Stimmen.
$B \Leftarrow \{v_i \leftarrow \mathsf{Abstimmung}(V_i, sk_{V_i}, PK_{\mathcal{K}}, PK_{\mathcal{A}}, \mathbf{L}, m_i, k_2)\}_{V_i \in W}$	Die ehrlichen Wähler schicken Stimmen ab.
$(\mathbf{T_2}, P_{T,2}) \leftarrow \mathsf{Auszaehlung}(SK_{\mathcal{A}}, B, \mathbf{L}, \{pk_{V_i}\}_{i=1}^{n_V}, k_3)$	Erneute Auszählung der bisherigen Stimmen.
$B \Leftarrow \{$Stimme $\leftarrow X(B,$ 'versuche, Stimme für $V_i \in W$ abzugeben')$\}$	X versucht, Stimmen ehrlicher Wähler abzugeben.
$B \Leftarrow \{$Stimme $\leftarrow X(B,$ 'versuche, Stimme für $V_i \notin V$ abzugeben')$\}$	X versucht, Stimmen hinzuzufügen.
$(\mathbf{T_3}, P_{T,3}) \leftarrow \mathsf{Auszaehlung}(SK_{\mathcal{A}}, B, \mathbf{L}, \{pk_{V_i}\}_{i=1}^{n_V}, k_3)$	Erneute Auszählung der bisherigen Stimmen.
$B \Leftarrow \{$Stimme $\leftarrow X(B,$ 'gibt Stimmen für $V_i \in U$ ab')$\}$	X gibt die Stimmen korrumpierter Wähler ab.
$(\mathbf{T_4}, P_{T,4}) \leftarrow \mathsf{Auszaehlung}(SK_{\mathcal{A}}, B, \mathbf{L}, \{pk_{V_i}\}_{i=1}^{n_V}, k_3)$	Erneute Auszählung aller Stimmen.
if $\Big((\mathsf{Verifikation}(PK_{\mathcal{A}}, B, \mathbf{L}, \mathbf{T_1}, P_{T,1}) = 1)$ and $\Big(\sum_{i=1}^{n_L} t_{i,1} > 0\Big)\Big)$	Hat X Stimmen für ehrliche Wähler abgegeben?
or $\Big((\mathsf{Verifikation}(PK_{\mathcal{A}}, B, \mathbf{L}, \mathbf{T_3}, P_{T,3}) = 1)$ and	
$(\mathsf{Verifikation}(PK_{\mathcal{A}}, B, \mathbf{L}, \mathbf{T_2}, P_{T,2}) = 1)$ and $\Big(\sum_{i=1}^{n_L} t_{i,3} - \sum_{i=1}^{n_L} t_{i,2} > 0\Big)\Big)$	Hat X Stimmen für ehrliche oder für nicht-wahlberechtigte Wähler abgegeben?
or $(\mathsf{Verifikation}(PK_{\mathcal{A}}, B, \mathbf{L}, \mathbf{T_4}, P_{T,4}) = 1)$ and $\{v_i\}_{V_i \in W} \not\subseteq T_4$	Hat X eine Stimme unbemerkt verändert?[a]
then output 1	X hat erfolgreich manipuliert.
else output 0	X hat keinen Erfolg gehabt.

Abbildung 4.1: $\mathbf{Exp}^{Korr.}_{\mathsf{WS},X}(k_1, k_2, k_3, n_L, n_V)$.

[a] Die Stimme kann gültig sein, dann hat der Angreifer eine Stimme abgeändert. Die Stimme kann auch ungültig sein, dann hat X de facto eine Stimme gelöscht. In jedem Fall geht der Wähler davon aus, dass seine Stimme erfolgreich abgegeben wurde.

Abbildung 4.2: $\mathbf{Exp}^{E.-res}_{\mathsf{WS},X}(k_1, k_2, k_3, n_L, n_V)$.

$\mathbf{Exp}^{E.-res}_{\mathsf{WS},X}(k_1, k_2, k_3, n_L, n_V)$

$U \leftarrow X(\{pk_{V_i}\}_{i=1}^{n_V},$ 'erstelle Menge kontrollierter Wähler') — X korrumpiert Wähler U.

$\{(sk_{V_i}, pk_{V_i}) \leftarrow \mathsf{Registrierung}(SK_{\mathcal{R}}, V_i, k_1)\}_{i=1}^{n_V}$ — Die Wähler werden registriert.

$(V_j, m_j) \leftarrow X(\{sk_{V_i}\}_{V_i \in U}, \mathbf{L},$ 'zu erpressenden Wähler und dessen Stimme festlegen') — X bestimmt ehrlichen Wähler, erpresst ihn und gibt dessen Stimme vor.

if $|U| \neq n_U$ — Wenn X nicht alle korrupten Wähler besticht

  or $V_j \notin V \setminus U$ — oder der Erpresste nicht wahlberechtigt oder korrupt ist

  or $m_j \notin \langle L \rangle \cup \{\phi\}$ then — oder die erpresste Stimme keine gültige Wahloption ist,

  output 0 — dann bricht das Experiment mit Ausgabe 0 ab.

else

  $b \in_R \{0, 1\}$ — Münzwurf (ein Bit b wird gewählt)

  if $b = 0$ then — Fall, in dem der Wähler der Erpressung widersteht.

    $\widetilde{sk} \leftarrow \mathsf{Schluesselfaelschung}(PK_{\mathcal{A}}, sk_{V_j}, pk_{V_j})$ — Der Wähler lässt sich nicht erpressen, fälscht das Credential

    $B \Leftarrow \mathsf{Abstimmung}(sk_{V_j}, PK_{\mathcal{A}}, \mathbf{L}, m_i, k_2)$ — und gibt seine Stimme mit seinem echten Credential ab.

  else — Fall $b = 1$, in dem sich der Wähler erpressen lässt.

    $\widetilde{sk} \leftarrow sk_{V_j}$ — Der Wähler gibt seinen echten geheimen Schlüssel preis.

    $B \Leftarrow \mathsf{Abstimmung}((sk_{V_i})_{V_i \neq V_j, V_i \notin U}, PK_{\mathcal{A}}, \mathbf{L}, D_{n_W, n_L}, k_2)$ — Stimmabgabe der ehrlichen Wähler.

    $B \Leftarrow X(\widetilde{sk},$B, 'Stimmabgabe') — Angreifer X gibt Stimmen ab.

  $(\mathbf{T}, P_T) \leftarrow \mathsf{Auszaehlung}(SK_{\mathcal{A}}, B, \mathbf{L}, \{pk_{V_i}\}_{i=1}^{n_V}, k_3)$ — Die Auszählung wird durchgeführt.

  $b' \leftarrow X(\mathbf{T}, P_T,$ 'ermittle, bzw. rate' $b)$ — X äußert Vermutung über Erpressbarkeit.

  if $b' = b$ then output 1 — X erkennt, dass seine Vermutung richtig war.

  else output 0 — X liegt mit seiner Vermutung falsch.

$\mathbf{Exp}^{p.-E.-res.}_{\mathsf{WS},X'}(k_1,k_2,k_3,n_L,n_V)$

$U \leftarrow X'(\{pk_{V_i}\}_{i=1}^{n_V},$ 'erstelle Menge kontrollierter Wähler')	X' korrumpiert Wähler U.
$\{(sk_{V_i},pk_{V_i}) \leftarrow \mathsf{Registrierung}(SK_{\mathcal{R}},V_i,k_1)\}_{i=1}^{n_V}$	Die Wähler werden registriert.
$(V_j,m_j) \leftarrow X'(\{sk_{V_i}\}_{V_i\in U},\mathbf{L},$ 'zu erpressenden Wähler und dessen Stimme festlegen')	X' bestimmt das Ziel der Erpressung.
if $\lvert U\rvert \neq n_U$	Wenn X' nicht alle korrupten Wähler besticht
or $V_j \notin V \setminus U$	oder der Erpresste nicht wahlberechtigt oder korrupt ist
or $m_j \notin \langle L\rangle \cup \{\phi\}$ then	oder die erpresste Stimme keine gültige Wahloption ist,
output 0	dann bricht das Experiment mit Ausgabe 0 ab.
else	
$b \in_R \{0,1\}$	Münzwurf (ein Bit b wird gewählt)
if $b = 0$ then	Fall, in dem der Wähler der Erpressung widersteht.
$B \Leftarrow \mathsf{Abstimmung}(sk_{V_j},PK_{\mathcal{A}},\mathbf{L},m_i,k_2)$	Der Wähler lässt sich nicht erpressen u. gibt seine Stimme ab.
$\widetilde{sk} \leftarrow sk_{V_j}$	Der Wähler gibt seinen echten geheimen Schlüssel preis.
$B \Leftarrow \mathsf{Abstimmung}((sk_{V_i})_{V_i\neq V_j,V_i\notin U},PK_{\mathcal{A}},\mathbf{L},D_{n_W,n_L},k_2)$	Stimmabgabe der ehrlichen Wähler.
$B \Leftarrow X'(\widetilde{sk},\{sk_{V_i}\}_{V_i\in U},$ 'Stimmabgabe')	Angreifer X' gibt Stimmen ab.
$(\mathbf{T},P_T) \leftarrow \mathsf{perfekte-Auszaehlung}(SK_{\mathcal{A}},B,\mathbf{L},\{pk_{V_i}\}_{i=1}^{n_V},k_3)$	Die perfekte Auszählung wird durchgeführt.
$b' \leftarrow X(\mathbf{T},P_T,$ 'ermittle, bzw. rate'b)	X' äußert Vermutung über Erpressbarkeit.
if $b' = b$ then output 1	X' erkennt, ob Erpressung erfolgreich war.
else output 0	X' liegt mit seiner Vermutung falsch.

Abbildung 4.3: $\mathbf{Exp}^{p.-E.-res.}_{\mathsf{WS},X'}(k_1,k_2,k_3,n_L,n_V)$.

It's not the voting that's democracy, it's the counting.

Tom Stoppard

Kapitel 5

Effiziente observerbasierte Wahlsysteme mit Unüberprüfbarkeit

Das System von Hirt und Sako [HS00] hatte bis zum Jahr 2005 die Unüberprüfbarkeit am besten realisiert. Allerdings wird dort vorausgesetzt, dass ein physikalisch sicherer Kanal von jeder Autorität im Wahlausschuss zu jedem Wähler besteht. Die Anforderung ist für mittlere und große Wahlen unrealistisch, da dies nicht durch Verschlüsselung oder andere kryptografische Maßnahmen zu erreichen ist.[1]

Ein weiteres Wahlsystem, das laut Meinung der Autoren Unüberprüfbarkeit bietet, ist das System von Magkos et al. [MBC01]. Dieses Wahlsystem ist effizienter als das von Hirt und Sako [HS00], da es zur Stimmabgabe eine manipulationssichere Hardware, einen Observer, verwendet. In [Sch05a] habe ich allerdings gezeigt, dass die Unüberprüfbarkeit nicht gegeben ist, da der innerhalb des Protokolls verwendete Zero-Knowledge-Beweis vom Observer gegenüber dem Wähler übertragbar ist, wenn der Wähler *eine* Challenge wählt, die der Angreifer vorher festgelegt hat. In Abschnitt 5.1.3 wird beschrieben, wie dieser Angriff funktioniert und wie man ihm begegnen kann. Das Wahlverfahren von Magkos et al. weist neben der fehlenden Unüberprüfbarkeit noch eine weitere Schwäche auf. Da das verwendete Verschlüsselungsverfahren malleable (vgl. Abschnitt 2.3) ist, könnte es einem Angreifer gelingen, eine Kopie einer abgegebenen verschlüsselten Wahloption als seine eigene Stimme abzugeben oder diesen Geheimtext so zu modifizieren, dass die zugrundeliegenden Klartexte in einer dem Angreifer bekannten Relation stehen. Somit ist die Unabhängigkeit der Stimmabgabe nicht gewährleistet. Das in Abschnitt 5.1 von mir verbesserte Verfahren verwendet eine non-malleable Verschlüsselung und schließt somit diese Sicherheitslücke.

In Abschnitt 5.2 wird das von mir in [Sch05a] vorgestellte Wahlsystem mit Observer analysiert. Dieses Wahlsystem bietet Unüberprüfbarkeit und ist effizienter als die bisher bekannten Wahlsysteme mit Unüberprüfbarkeit.

In Abschnitt 5.3 wird ein weiteres Wahlsystem konstruiert und analysiert, das auf dem von mir in [Sch06a] vorgestellten Wahlsystem aufbaut. Dieses Wahlsystem verzichtet auf den Beweis von Seiten des Wählers bzw. Observers, dass die abgegebene Stimme eine gültige Wahlopti-

[1] Wenn der Wähler erpresst oder bestochen wird, so kann er dem Angreifer beweisen, welche Nachrichten er vom Wahlausschuss erhalten hat, indem er einfach die Verschlüsselung dieser Nachrichten aufdeckt.

on beinhaltet. Ungültige Stimmen werden später bei der Auszählung ignoriert. Dazu muss im Rahmen der Auszählung ein verifizierbares MIX-Netz eingesetzt werden. Der Wahlaufwand auf Seiten des Wählers wird dadurch deutlich reduziert.
Schließlich werden die vorgestellten Wahlsysteme im vierten Abschnitt vergleichend analysiert.

5.1 Unüberprüfbarkeit im System [MBC01]

Das Wahlsystem von Magkos et al. basiert auf dem Wahlschema von Cramer, Gennaro und Schoenmakers [CGS97]. Die Verschlüsselung der Stimme wird allerdings gemeinsam von Wähler und Observer übernommen.

Entgegen der Annahme der Autoren ist das System nicht unüberprüfbar, da der Angreifer eine vom Wähler zu erzeugende Challenge vorgeben kann. Der Beweis des Observers, korrekt verschlüsselt zu haben, wird dann übertragbar (vgl. Abschnitt 5.1.3). Darüber hinaus ist die Unabhängigkeit der Stimmabgabe im ursprünglichen System nicht gewährleistet.

Das System von Magkos et al. basiert auf der ElGamal-Verschlüsselung (vgl. Abschnitt 2.2.4). Diese Verschlüsselung ist malleable. Ein Angreifer ist also unter Umständen in der Lage, einen Geheimtext in Abhängigkeit eines anderen Geheimtextes zu erzeugen, so dass die zugrundeliegenden Klartexte in einer dem Angreifer bekannten Relation stehen. Um die Unabhängigkeit der Stimmabgabe zu gewährleisten, setzen wir hier stattdessen eine non-malleable ElGamal-Verschlüsselung ein (siehe Abschnitt 2.3). Dabei ist zu beachten, dass nicht der Wähler alleine, sondern Wähler und Observer gemeinsam die Geheimtexte erzeugen, und die dafür verwendeten Zufallszahlen teilen sich auf Observer und Wähler auf. Daher muss eine Modifikation der non-malleable ElGamal-Verschlüsselung (vgl. Abschnitt 2.3) für die zwei Parteien verwendet werden.

Es sei G eine endliche Untergruppe mit Primzahlordnung $|G| = q$ von $\mathbb{Z}_p^*$, wobei p eine Primzahl ist und $q|p-1$ gilt. g und γ seien erzeugende Elemente von G. $\mathcal{H}$ sei eine kryptografische Hashfunktion. Im ursprünglichen System von Magkos et al. sind nur zwei mögliche Stimmen, z. B. (nein, ja) vorgesehen: $\mathbf{L} = (m_1, m_2) = (-1, 1)$.

5.1.1 Registrierungsphase

Der Wähler V authentifiziert sich im Registrierungsbüro gegenüber den Registrierungsautoritäten und erhält seinen Observer, der für mehrere Wahlen verwendet werden kann. Auf dem Observer ist das Zertifikat des öffentlichen Signaturschlüssels des Wählers sowie der öffentliche Schlüssel h zum verteilten geheimen Schlüssel der Wahlautoritäten gespeichert (siehe Abschnitt 3.7.5). Außerdem wird auf dem Observer noch ein Identitätstoken id_V des Wählers V gespeichert - eine Ergänzung, um eine non-malleable Verschlüsselung zu erreichen.

5.1.2 Stimmerzeugung - Verschlüsselung, non-malleability

Der Wähler V erzeugt die Zufallszahlen $a_1, a_2, a_1', a_2' \in \mathbb{Z}_q$ und verschlüsselt die beiden Wahloptionen m_1 und m_2 mit Hilfe der ElGamal-Verschlüsselung:

$$(x_1, y_1) = (g^{a_1}, h^{a_1}\gamma^{m_1}) = \left(g^{a_1}, h^{a_1}\gamma^{-1}\right) \text{ und } (x_2, y_2) = (g^{a_2}, h^{a_2}\gamma^{m_2}) = (g^{a_2}, h^{a_2}\gamma)\,.$$

Bevor er $\left(g^{a'_1}, x_1, y_1\right)$ und $\left(g^{a'_2}, x_2, y_2\right)$ an den Observer sendet, ordnet er diese lexikographisch, um die Reihenfolge der enthaltenen Wahloptionen zu verbergen. Es sei π die dazu verwendete Transposition.

Der Observer erzeugt ebenfalls Zufallszahlen $b_1, b_2, b'_1, b'_2 \in \mathbb{Z}_q$ und verschlüsselt die beiden Geheimtexte erneut:

$$\left(x'_1, y'_1\right) = \left(g^{b_1} g^{a_{\pi(1)}}, h^{b_1} h^{a_{\pi(1)}} \gamma^{m_{\pi(1)}}\right) \text{ und } \left(x'_2, y'_2\right) = \left(g^{b_2} g^{a_{\pi(2)}}, h^{b_2} h^{a_{\pi(2)}} \gamma^{m_{\pi(2)}}\right).$$

Er berechnet außerdem die für die non-malleability notwendigen Werte

$$g^{a'_{\pi(1)}+b'_1},\ g^{a'_{\pi(2)}+b'_2},\ b_1 \cdot \mathcal{H}\left(g, x'_1, y'_1, g^{a'_{\pi(1)}+b'_1}, id_V\right) + b'_1 \text{ und } b_2 \cdot \mathcal{H}\left(g, x'_2, y'_2, g^{a'_{\pi(2)}+b'_2}, id_V\right) + b'_2.$$

Dann sendet er $\left(x'_1,\ y'_1,\ g^{a'_{\pi(1)}+b'_1},\ b_1 \cdot \mathcal{H}\left(g, x'_1, y'_1, g^{a'_{\pi(1)}+b'_1}, id_V\right) + b'_1,\ id_V\right)$ und

$\left(x'_2,\ y'_2,\ g^{a'_{\pi(2)}+b'_2},\ b_2 \cdot \mathcal{H}\left(g, x'_2, y'_2, g^{a'_{\pi(2)}+b'_2}, id_V\right) + b'_2,\ id_V\right)$ an den Wähler.

Der Wähler kann daraus dann $g^{b_1} = x'_1 \cdot g^{-a_{\pi(1)}}$, $h^{b_1} = y'_1 \cdot h^{-a_{\pi(1)}}$ und $g^{b_2} = x'_2 \cdot g^{-a_{\pi(2)}}$, $h^{b_2} = y'_2 \cdot h^{-a_{\pi(2)}}$ berechnen, sofern der Observer korrekt gearbeitet hat. Des Weiteren kann der Wähler nun für die beiden Wahloptionen jeweils den nicht-interaktiven Zero-Knowledge-Beweis erstellen, der für die non-malleability bzw. Unabhängigkeit notwendig ist:

$$\begin{array}{l} \left(a_{\pi(1)} + b_1\right) \mathcal{H}\left(g, x'_1, y'_1, g^{a'_{\pi(1)}+b'_1}, id_V\right) + \left(a'_{\pi(1)} + b'_1\right) \quad \text{bzw.} \\ \left(a_{\pi(2)} + b_2\right) \mathcal{H}\left(g, x'_2, y'_2, g^{a'_{\pi(2)}+b'_2}, id_V\right) + \left(a'_{\pi(2)} + b'_2\right). \end{array}$$

5.1.3 Stimmerzeugung - keine Unüberprüfbarkeit in [MBC01] und Verbesserung

Der Observer beweist nun dem Wähler jeweils in *einem* interaktiven Zero-Knowledge-Beweis der Gleichheit diskreter Logarithmen (siehe Abschnitt 3.2.2), dass er korrekt gearbeitet hat. Er zeigt dafür, dass er Werte $b_1, b_2 \in \mathbb{Z}_q$ kennt, so dass gilt:

$$\log_g\left(g^{b_1}\right) = \log_h\left(h^{b_1}\right) \text{ und } \log_g\left(g^{b_2}\right) = \log_h\left(h^{b_2}\right).$$

Magkos, Burmester und Chrissikopoulos gehen davon aus, dass die zwei interaktiven Zero-Knowledge-Beweise des Observers gegenüber dem Wähler nicht übertragbar sind, ein Erpresser oder Bestecher also keinerlei Wahrheitsgehalt aus der Mitschrift des Wählers in dem Beweis ableiten kann.

Gibt der Angreifer dem Wähler allerdings die im Zero-Knowledge-Beweis verwendete Challenge $c \in_R \mathbb{Z}_q$ vor, so ist der Beweis nicht mehr simulierbar und somit übertragbar. Der Angreifer kann diesen übertragbaren Beweis als Beleg für die Anordnung der Wahloptionen in den Geheimtexten nehmen. Das Wahlsystem ist somit nicht unüberprüfbar.

Diese Schwäche habe ich in [Sch05a] behoben. Der Zero-Knowledge-Beweis darf nicht in drei Runden durchgeführt werden, sondern muss bitweise in $\log_2 c$ Durchführungen ablaufen (vgl. Abschnitt 3.2.2).

Der Vorteil ist, dass der Beweis simulierbar ist[2]. Die Mitschrift des Beweises kann nicht mehr als Beleg für die Anordnung der Wahloptionen innerhalb der Geheimtexte gelten. Das Wahlsystem ist unüberprüfbar.

Der Nachteil dieses Verfahrens ist der große Aufwand. Statt drei Runden sind für jede Gleichung nun $\log_2 c$ mal so viele Runden durchzuführen.

5.1.4 Stimmerzeugung - Stimmabgabe

Da der Wähler beide Wahloptionen verschlüsselt und vom Observer wiederverschlüsseln lässt, ist zu vermuten, dass Magkos et al. dem Observer die tatsächliche Auswahl der Wahloption vorenthalten wollen, um das Maß an Vertrauen zu minimieren, das der Wähler dem Observer entgegenbringen muss.

Zusätzlich zum Zero-Knowledge-Beweis der unabhängigen Stimmgenerierung muss der Wähler gegenüber dem Wahlausschuss beweisen, dass der von ihm und dem Observer erstellte Geheimtext eine Verschlüsselung einer gültigen Wahloption darstellt. Das in [MBC01] vorgeschlagene Beweisprotokoll ist nicht effizient simulierbar und weist daher ebenfalls nicht die Zero-Knowledge-Eigenschaft auf. Das stellt keine Schwächung der Anonymität oder Unüberprüfbarkeit dar. Der beschriebene Beweis ist vielmehr ein interaktiver Witness-Indistinguishable-Beweis (siehe Abschnitt 3.6).

In Abschnitt 5.3 wird ein Wahlprotokoll vorgestellt, das eine Erweiterung dieses Beweises von zwei auf n_L Geheimtexte verwendet. Dieser Beweis muss für beide Wahloptionen generiert werden.

Wähler und Observer erzeugen gemeinsam für jeden Geheimtext einen 1-von-2 Witness-Indistinguishable-Wiederverschlüsselungsbeweis, dass der Geheimtext eine korrekte Wahloption enthält.

Bei Magkos et al. wird die Wahlberechtigung nicht geprüft. Da der Observer unbedingt an der Generierung der Stimme beteiligt werden muss, muss er die Wahlberechtigung verwalten. Der Observer signiert daher die Geheimtexte inklusive der Beweise mit dem geheimen Signaturschlüssel des Wählers und sendet die Signatur an den Wähler. Der Wähler kann nun die Signatur überprüfen, die Stimme auswählen und seine signierte Stimme an das Schwarze Brett schicken.

5.1.5 Auszählung

Die Auszählung ist für die Unüberprüfbarkeit im System [MBC01] nicht entscheidend. Sie stellt einen Spezialfall für zwei Wahloptionen der Auszählung des in Abschnitt 5.3 vorgestellten Wahlverfahrens für n_L Wahloptionen dar. Daher habe ich mich hier auf die genaue Beschreibung und Verbesserung der Stimmerzeugungsphase beschränkt, in der die oben genannten Schwächen bestanden.

[2] Gibt der Angreifer ein unpassendes Bit als Challenge vor, wird die Runde einfach gelöscht. Es ist zu erwarten, dass daher „nur“ $2 \cdot \log_2 c$ Durchführungen nötig sind.

5.2 Ein allgemeines observerbasiertes Wahlsystem mit Unüberprüfbarkeit

Der entscheidende Nachteil des in Abschnitt 5.1 verbesserten Systems ist die Effizienz. Zwar verzichtet es im Gegensatz zu [HS00] auf die unrealistische Forderung der physikalisch sicheren Kommunikationskanäle von den Autoritäten zu den Wählern, doch ist der Aufwand bei der Stimmerzeugung durch die $\log_2 c$ Runden für jede Gleichung (d. h. pro Wahloption) während der interaktiven Zero-Knowledge-Beweise erheblich.

Daher wird im Folgenden ein allgemeines Wahlsystem mit Observer beschrieben, das die Vorteile der Systeme von [HS00] und [MBC01] aufweist, jedoch effizienter ist und auf unrealistische Anforderungen verzichtet. Darüber hinaus wird das System von zwei auf n_L mögliche Wahloptionen erweitert. Dieses Wahlsystem entspricht im Wesentlichen dem von mir in [Sch05a] vorgestellten System.

5.2.1 Anforderungen

Das im Folgenden beschriebene Wahlsystem kann mit verschiedenen kryptografischen Primitiven konstruiert werden. Es sind folgende Grundanforderungen an das Protokoll zu stellen:

- *Existenz einer PKI:* Dem Protokoll muss eine Public-Key-Infrastruktur (PKI) zugrunde liegen.
- *Homomorphie:* Die Public-Key-Verschlüsselungsfunktion soll homomorph sein.
- *Non-malleability:* Das verwendete Verschlüsselungsverfahren muss non-malleable sein (vgl. Definition 2.2), damit es einem Angreifer nicht möglich ist, abgegebene Stimmen zu kopieren oder sinnvoll[3] modifiziert als seine eigene abzugeben. Andernfalls wäre die Unabhängigkeit der Stimmabgabe nicht gewährleistet.
- *Probabilistische Wiederverschlüsselung:* Um die Unüberprüfbarkeit zu ermöglichen, benötigt man einen probabilistischen Wiederverschlüsselungsalgorithmus R, so dass die Wiederverschlüsselung über der Menge der möglichen Chiffretexte gleichverteilt ist.
- *Existenz eines 1-von-n_L Witness-Indistinguishable-Wiederverschlüsselungsbeweises:* Basierend auf R existiert ein 1-von-n_L WI-Wiederverschlüsselungsbeweis für zwei Prover (vgl. Abschnitt 3.6.1), dass ein Chiffretext c_k aus einer Liste von Chiffretexten $c_1, \ldots, c_{n_L}$ tatsächlich eine Wiederverschlüsselung des Chriffretextes c ist, ohne k zu offenbaren.
- *Existenz eines Designated-Verifier-Wiederverschlüsselungsbeweises:* Es wird ein effizienter Designated-Verifier-Beweis benötigt (vgl. Abschnitt 3.5), der dem Wähler als Designated-Verifier beweist, dass ein Geheimtext eine gültige Wiederverschlüsselung eines anderen Geheimtextes darstellt.

[3] Mit einer sinnvollen Modifikation ist eine Änderung der Geheimtexte gemeint, bei der der Angreifer die Relation zwischen den zugrunde liegenden Klartexten kennt.

- *Kenntnis des geheimen Schlüssels:* Jeder Wähler wählt einen öffentlichen Schlüssel, wobei sichergestellt werden muss, dass er seinen dazugehörigen privaten Schlüssel kennt. Dies ist essentiell für die Unüberprüfbarkeit des Wahlsystems. Kennt ein Wähler seinen geheimen Schlüssel nicht, wird der Designated-Verifier-Beweis übertragbar.

- *Observer:* Jeder Wähler besitzt eine manipulationssichere Hardware, einen Observer, der in der Lage ist, R, den Designated-Verifier-Beweis und mit dem Wähler zusammen die 1-von-n_L WI-Wiederverschlüsselungsbeweise sowie die Zero-Knowledge-Beweise der Kenntnis der zur Verschlüsselung verwendeten Zufallszahlen durchzuführen. Auf dem Observer muss dazu der Public-Key des Wahlausschusses, der Signaturschlüssel des Wählers und der öffentliche Schlüssel des Wählers gespeichert sein. Der Observer muss außerdem mit Hilfe des Signaturschlüssels qualifizierte digitale Signaturen erstellen können.

- *Existenz eines Schwarzen Brettes:* Die Stimmen werden an ein Schwarzes Brett geschickt, auf dem jeder Leserechte und das Recht besitzt, Daten anzuhängen, aber keiner über das Recht verfügt, Daten zu löschen oder zu ändern.

- *Sicherheit der Verschlüsselung:* Der Wahlausschuss besteht aus $n \geq t$ Personen. Für jede Gruppe von weniger als t Personen des Wahlausschusses muss es unmöglich sein, einen Chiffretext zu entschlüsseln.

- *Verifizierbare Entschlüsselung:* Für eine gegebene Summe von Chiffretexten kann der Wahlausschuss die zugrundeliegende Summe der Klartextstimmen effizient berechnen und beweisen, dass diese Entschlüsselung korrekt ist. Die Entschlüsselung und der Beweis dürfen jedoch keine Informationen aufdecken, die es ermöglichen, einen Chiffretext durch weniger als t Personen des Wahlausschusses zu entschlüsseln. Die Entschlüsselung muss aber noch funktionieren, wenn bis zu $n_A - t$ Personen des Wahlausschusses die Kooperation verweigern oder sich protokollkonform verhalten.

- *Stimmabgabe in einer virtuellen Wahlkabine:* Die Kommunikation zwischen Observer und Wähler darf nicht durch Angreifer überwacht werden können. Wähler und Observer befinden sich sozusagen in einer virtuellen Wahlkabine.

5.2.2 Allgemeiner Ablauf des Wahlsystem

Registrierungsphase

Der Wähler begibt sich in das Registrierungsbüro, authentifiziert sich gegenüber den Registrierungsautoritäten und erhält seinen Observer.
Auf dem Observer sind der öffentliche Schlüssel der Auszählungsautoritäten und der geheime Signaturschlüssel des Wählers gespeichert. Der zugehörige öffentliche Signaturschlüssel wird auf der Liste wahlberechtigter Signaturschlüssel von den Registrierungsautoritäten pseudonym (z. B. über ein MIX-Netz) veröffentlicht. Diese Liste wird, wie auch die Liste der möglichen Wahloptionen von den Registrierungsautoritäten signiert und veröffentlicht.
Der Wähler generiert sich einen privaten Schlüssel und den dazugehörigen öffentlichen Schlüssel.

Der öffentliche Schlüssel wird auf dem Observer gespeichert.
Es ist für die Nichtübertragbarkeit des Designated-Verifier-Beweises wichtig, dass der Wähler seinen privaten Schlüssel kennt, der Observer darf diesen hingegen nicht kennen.

Wahlphase, Auszählung

Der Wähler verschlüsselt (non-malleable) jede der möglichen Wahloptionen unter dem öffentlichen Schlüssel der Auszählungsautoritäten und sendet diese Geheimtexte an den Observer, der sie wiederverschlüsselt. Neben den wiederverschlüsselten Geheimtexten sendet der Observer nichtinteraktive Designated-Verifier-Beweise der korrekten Wiederverschlüsselung an den Wähler.

Der Wähler verifiziert die Beweise und erstellt zusammen mit dem Observer 1-von-n_L WI-Wiederverschlüsselungsbeweise (vgl. Abschnitt 3.6.1) für jede Wahloption. Der Observer signiert jede wiederverschlüsselte Nachricht zusammen mit dem WI-Wiederverschlüsselungsbeweis und sendet die Signaturen an den Wähler. Dieser kann die Signaturen verifizieren, wählt seine Stimme aus und gibt diese ab, d. h. er sendet sie an das Schwarze Brett.

Auf dem Schwarzen Brett kann jeder die Signatur und die Beweise der non-malleability bzw. der korrekten Stimmerzeugung überprüfen. Die gültigen, auszuzählenden Geheimtexte werden homomorph verrechnet und anschließend das Auszählungsergebnis verifizierbar entschlüsselt.

Im Folgenden wird ein konkretes Beispiel eines solchen observerbasierten Wahlsystems mit Unüberprüfbarkeit beschrieben und untersucht.

5.2.3 Beispiel eines observerbasierten Wahlsystems mit Unüberprüfbarkeit

Wahlvorbereitung

Die Autoritäten $A_1, \ldots, A_n$ bestimmen zusammen eine multiplikative Gruppe G mit Primzahlordnung $|G| =: q$ und erzeugenden Elementen g und γ von G. Dann erzeugen sie gemeinsam ein ElGamal-Schlüsselpaar (s, h) mit $h = g^s$ (siehe Abschnitt 3.7.5), so dass jede Person des Wahlausschusses A_j einen Anteil s_j von s in einem $(t, n)-$Schwellensystem erhält und öffentlich auf diesen Anteil über $h_j = g^{s_j}$ festgelegt wird.

Durch das Schwellensystem der Verschlüsselung ist gewährleistet, dass nur t oder mehr Autoritäten zusammen eine mit dem öffentlichen Schlüssel h verschlüsselte Stimme entschlüsseln können. Dieser öffentliche Schlüssel h wird als öffentlicher Schlüssel des Wahlausschusses bekanntgegeben und auf den Observern gespeichert.

Registrierungsphase

Jeder Wahlberechtigte V_i, $(i = 1, \ldots, n_V)$, wird vom Wahlausschuss benachrichtigt und begibt sich zum Registrierungsbüro, wo er nach einer geeigneten Authentifikation einen Observer ausgehändigt bekommt.

Der Wähler wählt ein Geheimnis $z_V \in_R \mathbb{Z}_q$ und berechnet $h_V = g^{z_V}$ als öffentlichen Anteil des Wertes z_V. Dieser Wert h_V wird ebenfalls auf dem Observer gespeichert. Wichtig ist, dass der Observer z_V selbst nicht kennt.

Auf dem Observer ist neben dem öffentlichen Schlüssel des Wahlausschusses und des Wertes h_V noch der geheime Signaturschlüssel des Wählers gespeichert. Der dazu gehörende öffentliche Schlüssel wird auf einer Liste der wahlberechtigten Signaturen vom Registrierungsbüro zertifiziert.

Wahlphase

Um das in Abschnitt 5.1 verbesserte System auf n_L Stimmen zu erweitern, werden in diesem Wahlsystem die Wahloptionen in einem Zahlensystem zur Basis n_V beschrieben. Da n_V die Anzahl der Wähler ist, kann es nur dann zu einer Stellenüberschreitung kommen, wenn alle Wähler die gleiche Wahloption wählen. Sei also $\mathbf{L} = (m_1, \ldots, m_{n_L}) = \left(1, n_V, n_V^2, \ldots, n_V^{n_L-1}\right)$ die Menge der Wahlmöglichkeiten, dann kann man leicht die Anzahl der Stimmen für jede Wahlmöglichkeit berechnen, wenn die Summe der Stimmen berechnet wurde (vgl. Abschnitt 5.2.3). Die Wahloptionen werden so verschlüsselt, dass sie sich bei der Multiplikation der Geheimtexte im Exponenten addieren. Damit Wähler und Observer gemeinsam nachweisen können, dass sie eine gültige Wahloption verschlüsselt haben, wird für jede der Wahloptionen $1, \ldots, n_V^{n_L-1}$ vorausgesetzt, dass eine deterministische Standardverschlüsselung existiert. Diese Verschlüsselung der Wahloptionen kann von jedem nachvollzogen werden, indem man beispielsweise alle Wahloptionen so verschlüsselt und die Geheimtexte vergleicht. Eine solche Standardverschlüsselung einer Wahloption m_ℓ aus $\mathbf{L}$ habe die Form $(1, \gamma^{m_\ell})$. Die Liste der standardverschlüsselten Wahloptionen hat dann die Gestalt $((\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_{n_L}, \tilde{y}_{n_L})) = \left(\left(1, \gamma^1\right), \ldots, \left(1, \gamma^{n_V^{n_L}}\right)\right)$.

Jeder Wähler wählt Zufallszahlen $a_1, \ldots, a_{n_L} \in_R \mathbb{Z}_q$ und verschlüsselt

$$(x_1, y_1) = \left(g^{a_1}, h^{a_1}\gamma^{m_1}\right), \ \ldots, \ (x_{n_L}, y_{n_L}) = \left(g^{a_{n_L}}, h^{a_{n_L}}\gamma^{m_{n_L}}\right).$$

Diese Geheimtexte sendet der Wähler an den Observer.

Der Observer wählt zufällig eine Permutation π aus Σ_{n_L} und permutiert die Geheimtexte. Anschließend wählt der Observer Zufallszahlen $b_1, \ldots, b_{n_L}$ und verschlüsselt die permutierten Geheimtexte:

$$\left(x'_1, y'_1\right) = \left(g^{b_1} g^{a_{\pi(1)}}, h^{b_1} h^{a_{\pi(1)}} \gamma^{m_{\pi(1)}}\right), \ \ldots, \ \left(x'_{n_L}, y'_{n_L}\right) = \left(g^{b_{n_L}} g^{a_{\pi(n_L)}}, h^{b_{n_L}} h^{a_{\pi(n_L)}} \gamma^{m_{\pi(n_L)}}\right).$$

Er berechnet die Werte $g^{a'_{\pi(1)}+b'_1}, \ldots, g^{a'_{\pi(n_L)}+b'_{n_L}}$ und daraus die für die non-malleability notwendigen Werte

$$b_1 \cdot \mathcal{H}\left(g, x'_1, y'_1, g^{a'_{\pi(1)}+b'_1}, id_V\right) + b'_1, \ \ldots, \ b_{n_L} \cdot \mathcal{H}\left(g, x'_{n_L}, y'_{n_L}, g^{a'_{\pi(n_L)}+b'_{n_L}}, id_V\right) + b'_{n_L}.$$

Dann sendet er

$$\left(x'_1, y'_1, g^{a'_{\pi(1)}+b'_1}, b_1 \cdot \mathcal{H}\left(g, x'_1, y'_1, g^{a'_{\pi(1)}+b'_1}, id_V\right) + b'_1, id_V\right), \ \ldots,$$
$$\left(x'_{n_L}, y'_{n_L}, g^{a'_{\pi(n_L)}+b'_{n_L}}, b_{n_L} \cdot \mathcal{H}\left(g, x'_{n_L}, y'_{n_L}, g^{a_\pi(n_L)'+b'_{n_L}}, id_V\right) + b'_{n_L}, id_V\right),$$

sowie die verwendete Permutation π an den Wähler. Zwar kennen nun Wähler und Observer die Permutation, sie ist aber dennoch nicht überflüssig. Da der Observer die Permutation erzeugt und anwendet, kann der Wähler einem Angreifer gegenüber nicht beweisen, welche Permutation

der Observer verwendet hat. Der Wähler kann jede beliebige Permutation als die vom Observer erhaltene ausgeben. Er muss sich allerdings sicher sein, dass die Permutation, die er vom Observer erhalten hat, die richtige ist. Daher beweist der Observer mittels eines nicht-interaktiven Designated-Verifier-Beweises für jede Wahloption, dass $\left(x'_{\pi(i)}, y'_{\pi(i)}\right)$ eine Wiederverschlüsselung von (x_i, y_i) ist (siehe hierzu Abschnitt 3.5). Dieser Beweis ist nicht übertragbar, da er nur für den Designated-Verifier, also den Wähler eine Aussagekraft besitzt. Der Wähler kann den Beweis, da er das Geheimnis z_V kennt, problemlos so abändern, dass gezeigt wird, dass $\left(x'_j, y'_j\right)$ eine Wiederverschlüsselung von (x_i, y_i) ist (wobei $j \neq \pi(i)$ möglich ist). Der Angreifer kann zwar vom Wähler Aussagen fordern, doch alle Werte, die er vom Wähler erhält, kann er nicht als Beweis ansehen. Das Wahlsystem ist somit unüberprüfbar.

Voraussetzung für dieses Wahlverfahren ist wie bei anderen Verfahren auch, dass der Angreifer dem Wähler bei der Erstellung und Abgabe der Stimme nicht direkt über die Schulter schaut. Der Wähler genießt also zuhause die Privatsphäre einer virtuellen Wahlkabine. Andernfalls wäre es nicht möglich, ein geheimes Wahlsystem mit Unüberprüfbarkeit zu erstellen.

Wenn der Observer korrekt gearbeitet hat, kann der Wähler nun $\left(g^{b_1}, h^{b_1}\right), \ldots, \left(g^{b_{n_L}}, h^{b_{n_L}}\right)$ berechnen. Des Weiteren kann der Wähler für die Wahloptionen jeweils den nicht-interaktiven Zero-Knowledge-Beweis erstellen, der für die non-malleability bzw. Unabhängigkeit notwendig ist:

$$\left(a_{\pi(1)} + b_1\right) \mathcal{H}\left(g, x'_1, y'_1, g^{a'_{\pi(1)}+b'_1}, id_V\right) + \left(a'_{\pi(1)} + b'_1\right), \ldots,$$
$$\left(a_{\pi(n_L)} + b_{n_L}\right) \mathcal{H}\left(g, x'_{n_L}, y'_{n_L}, g^{a'_{\pi(n_L)}+b'_{n_L}}, id_V\right) + \left(a'_{\pi(n_L)} + b'_{n_L}\right).$$

Der Wähler muss zusammen mit dem Observer nachweisen, dass seine Geheimtexte tatsächlich verschlüsselte Wahloptionen sind. Dazu erstellen Observer und Wähler zusammen für jede standardverschlüsselte Wahloption einen Beweis, dass diese in verschlüsselter Form in der Liste der verschlüsselten Stimmen enthalten ist, ohne jedoch den Geheimtext konkret anzugeben. Dies kann jeweils effizient in nicht interaktiven Witness-Indistinguishable-Beweisen für zwei Prover der 1-aus-n_L Wiederverschlüsselung der standardverschlüsselten Wahloptionenen geschehen, wie es in Abschnitt 3.6.1 gezeigt wurde. Der Wähler lässt den Observer die verschlüsselten Wahloptionen, die Zero-Knowledge-Beweise der non-malleability, die WI-Beweise der korrekten Wiederverschlüsselung und schließlich eine der verschlüsselten Wahloptionen als seine Stimme signieren[4], prüft die Signatur und schickt dies alles an das Schwarze Brett.

Das Schwarze Brett ist öffentlich lesbar, d. h. die Beweise und die Signatur sind global verifizierbar. Jeder kann also die Wahlberechtigung, die Zero-Knowledge- und die WI-Beweise überprüfen.

Auszählungsphase

Nach Beendigung der Wahlphase werden die gültigen non-malleable Geheimtextstimmen, die von wahlberechtigten Wählern stammen[5] und von diesen als ihre Stimme ausgewählt wurden,

[4] Beispielsweise mit Hilfe des Schnorr-Signaturschemas [Sch91].

[5] Natürlich wird von jedem Wahlberechtigten nur eine Stimme entsprechend einer Policy, z. B. die zuerst abgegebene, gewertet.

komponentenweise multipliziert. Das Ergebnis ist das Wertepaar

$$Z := \left(\prod_{i=1}^{n_V} g^{\xi_i}, \prod_{i=1}^{n_V} h^{\xi_i} \gamma^{v_i} \right) = \left(g^{\sum_{i=1}^{n_V} \xi_i}, h^{\sum_{i=1}^{n_V} \xi_i} \gamma^{\sum_{i=1}^{n_V} v_i} \right),$$

wobei für $1 \leq i \leq n_V$ der Wert v_i die vom Wähler V_i gewählte Wahloption aus $\mathbf{L}$ darstellt. Das Ergebnis z wird anschließend von jeder Autorität des Wahlausschusses verifizierbar ohne explizite Berechnung des kompletten geheimen Schlüssels entschlüsselt (vgl. Abschnitt 3.7.5).

Man erhält den Wert $\gamma^{\sum_{i=1}^{n_V} v_i}$. Jedes v_i ist eine abgegebene Stimme, die jeweils einer der n_L Wahloptionen entspricht. Das bedeutet, dass man aus $T := \sum_{i=1}^{n_V} v_i = \sum_{r=1}^{n_L-1} t_r \cdot n_V^r$ durch die Darstellung der Wahloptionen die Anzahlen t_r der auf die einzelnen Optionen entfallenen Stimmen ablesen kann. Dazu muss der Wahlausschuss T aus γ^T berechnen.

Für jedes t_r, $(0 \leq r \leq n_L)$, gibt es n_V mögliche Werte. Darüber hinaus gilt die Beziehung $\sum_{r=0}^{n_L-1} t_r = n_V$, so dass es also insgesamt $n_V^{n_L-1}$ mögliche Werte für T gibt.

Die naive Methode der Ordnung $O\left(n_V^{n_L-1}\right)$, bei der alle möglichen Kombinationen durchprobiert werden, kann man aber noch entscheidend verbessern, indem man den Baby-Step-Giant-Step-Algorithmus[6] anwendet. Die Komplexität dieser Berechnung des diskreten Logarithmus liegt dann bei $O\left(\sqrt{n_V}^{n_L-1}\right)$.

5.2.4 Analyse des Wahlsystems

Das Protokoll ist zunächst sehr allgemein gehalten. Daher beziehen sich die Aussagen zur Analyse des Protokolls auf das in Abschnitt 5.2.3 angegebene konkrete Beispiel.

Wahlberechtigung

Die Überprüfung der Wahlberechtigung wird durch die digitale Signatur sichergestellt. In der Registrierungsphase werden die öffentlichen Signaturschlüssel der Wahlberechtigten auf einer Liste zertifiziert. Mit Hilfe der Zertifikate kann jeder feststellen, welche Nachrichten von Wahlberechtigten signiert wurden. Nachrichten, die unberechtigt abgegeben wurden, werden nicht berücksichtigt.

Einmaligkeit

Wenn nur die zuerst veröffentlichten Stimmen mit korrektem Gültigkeitsbeweis und korrekter Signatur gezählt werden, eventuell später veröffentlichte Stimmen der gleichen Wähler für ungültig erklärt und verworfen werden, ist sichergestellt, dass jeder Wähler nur einmal seine Stimme abgeben kann. Die Einmaligkeit basiert auf der Fälschungssicherheit des Signaturschemas.

Um die Reihenfolge der vom Wähler veröffentlichten Stimmen festzulegen und so die zuerst veröffentlichte Stimme auszuzeichnen, können die auf dem Schwarzen Brett eintreffenden Stimmen beispielsweise durch einen Zeitstempel-Dienst gestempelt werden.

[6] Der *Baby-Step-Giant-Step-Algorithmus* (siehe [BSW06]) ist ein Verfahren zur Berechnung des Diskreten Logarithmus. Die Komplexität des Algorithmus ist $O\left(\sqrt{q}\right)$ in einer zyklischen Gruppe G mit $|G| = q$.

Fälschungssicherheit

Die Stimme eines Wähler ist fälschungssicher, da jeder Wähler die von ihm veröffentlichten Nachrichten digital signiert. Darüber hinaus kann kein Wähler seine Stimme nach deren Abgabe noch abändern, da nur die ersten veröffentlichten Stimmen gewertet werden. Die Fälschungssicherheit der Stimme leitet sich also direkt von der Fälschungssicherheit der digitalen Signatur ab.

Verifizierbarkeit

Während der Auszählungsphase müssen die Autoritäten die Summe der abgegebenen Stimmen als Entschlüsselung der verschlüsselten Geheimtexte berechnen. Da sie interaktiv beweisen, dass sie korrekt entschlüsselt haben, kann dies von allen Beteiligten nachvollzogen werden. Das bedeutet, dass dieses allgemeine Wahlverfahren global verifizierbar ist.

Korrektheit

Durch die 1-von-n_L-Wiederverschlüsselungsbeweise, die öffentlich geführt werden, können alle Teilnehmer oder Wahlbeobachter sicher sein, dass auch nur gültige verschlüsselte Wahloptionen gewählt wurden.

Die Korrektheit der Auszählung ist garantiert, wenn alle Wähler die Stimme ihrer Wahl abgeben können, also jeweils die Permutationen des Observers nachvollziehen können. Dies wird durch die Designated-Verifier-Beweise der Stimmenpermutation gewährleistet.

Ehrlichkeit, Robustheit

Ein unehrlicher Wähler kann keine ungültige Stimme in die Auszählung einfließen lassen, da er zu jeder erstellten Stimme einen Beweis der Gültigkeit erbringen muss.

Die globale Verifizierbarkeit der Aktionen der Autoritäten ermöglicht es, betrügerische Autoritäten zu identifizieren und auszuschließen. Solange höchstens $n_A - t$ Personen des Wahlausschusses unehrlich sind, kann die Wahl mit Ausschluss dieser Autoritäten durchgeführt werden. Daher ist das Wahlverfahren robust.

Wahlaufwand

Die Komplexität der Kommunikation hängt stark von den verwendeten Beweisen, also vom Designated-Verifier-Beweis der Permutation und vom 1-von-n_L-Wiederverschlüsselungsbeweis ab. Diese Beweise sind effizient durchführbar und der Kommunikationsaufwand ist linear in der Anzahl der Wahloptionen. Im Gegensatz zum Wahlverfahren von Hirt und Sako [HS00] ist der Aufwand jedoch unabhängig von der Anzahl der Autoritäten!

Da insbesondere der 1-von-n_L-Wiederverschlüsselungsbeweis für alle Wahloptionen durchgeführt, signiert und an das Schwarze Brett gesendet werden muss, wird in Abschnitt 5.3 ein allgemeines Wahlverfahren mit Observer vorgestellt und analysiert, bei dem der Aufwand auf Seiten des Wählers nochmals entscheidend reduziert wird.

Wahlgeheimnis, Anonymität

Die Anonymität eines jeden Wählers ist garantiert, wenn die verschlüsselte Stimme nicht durch eine außenstehende Person oder eine Gruppe von weniger als t Personen des Wahlausschusses entschlüsselt werden kann. Das trifft für dieses Wahlsystem zu.

Jeder kann feststellen, wer eine Stimme abgegeben hat, aber nur eine Menge von mindestens t Autoritäten kann das Vertauschen der Wahloptionen nachvollziehen und eine einzelne Stimme entschlüsseln, um festzustellen, wie ein einzelner Wähler gewählt hat. Solange nicht mehr als $t - 1$ Personen des Wahlausschusses kooperieren, ist die Anonymität durch die Sicherheit des Verschlüsselungsverfahrens gewährleistet.

Unabhängigkeit

Es ist nicht möglich, eine Stimme eines anderen Wählers zu kopieren, da man für jede verschlüsselte Wahloption einen Zero-Knowledge-Beweis liefern muss, die zur Verschlüsselung verwendete Zufallszahl zu kennen. Aufgrund der non-malleability der Verschlüsselung ist es nicht möglich, Stimmen abzugeben, die in einer dem Angreifer bekannten Korrelation zu Stimmen anderer Wähler stehen.

Unüberprüfbarkeit

Das Wahlsystem ist unüberprüfbar, d. h. es ist nicht möglich, dass der Wähler einen Beleg erstellen kann, wie er gewählt hat. Die Unüberprüfbarkeit in diesem System basiert auf dem Einsatz des Observers, der einen Teil der zur Verschlüsselung verwendeten Zufallszahl generiert. Diese Zufallszahl ist dem Wähler unbekannt und kann daher auch nicht als Beleg für die abgegebene Stimme gelten. Der Designated-Verifier-Beweis, mit dem der Observer dem Wähler beweist, welche Permutation er angewendet hat, ist nicht übertragbar. Der Wähler hat also keine Möglichkeit, andere beweisbar davon zu überzeugen, welche Wahloption in welchem Geheimtext verborgen ist.

Zusammenfassende Analyse

Das Wahlsystem erfüllt alle angegebenen Anforderungen einschließlich der Unüberprüfbarkeit. Es verzichtet auf unrealistische Annahmen physikalisch sicherer Kanäle von jedem Wähler zu jeder Person im Wahlausschuss. Es ist effizienter als das in Abschnitt 5.1 verbesserte System, welches für jede Wahloption einen *bitweisen* Beweis der Gleichheit diskreter Logarithmen verwendet. Hier muss nur ein Designated-Verifier-Beweis der korrekten Permutation pro Wahloption geführt werden.

Das System besitzt aber nicht nur Vorteile. Der Aufwand auf Seiten des Wählers sollte bei einer Wahl möglichst gering gehalten werden. Meist werden Chipkarten als Observer eingesetzt, die naturgemäß Rechnungen nicht besonders performant ausführen. Deshalb sollten nur notwendige Berechnungen auf Seiten des Wählers bzw. Observers durchgeführt werden. Im nächsten Abschnitt wird daher eine Verbesserung des hier vorgestellten Wahlsystems analysiert, bei der

wegen einer modifizierten Auszählung die Witness-Indistinguishable-Beweise der korrekten Verschlüsselung gültiger Wahloptionen entfallen.

Erpressungsresistenz

Ein Angreifer kann vom Wähler fordern, eine bestimmte Stimme aus der Liste der von Wähler und Observer erstellten Geheimtexte abzugeben. Zwar kennt der Angreifer den zugrundeliegenden Klartext nicht, dieser Angriff läuft aber auf eine Gleichverteilung der abgegebenen Stimmen hinaus, was insbesondere bei einer Parteienhochburg oder bei sonst schwachen Parteien Auswirkungen hat. Das Wahlsystem ist also nicht sicher gegen einen Randomisationsangriff.

Jeder kann anhand der Signatur überprüfen, ob eine bestimmte Person gewählt hat. Der Angreifer kann also den Wähler zwingen, keine Stimme abzugeben und dies kontrollieren. Ein Enthaltungsangriff (abstention attack) ist also möglich.

Ein Wähler könnte den Observer und seinen geheimen Schlüssel an den Angreifer übergeben, so dass dieser an Stelle des Wählers abstimmen kann. Diesem Impersonationsangriff kann leicht begegnet werden, indem zur Benutzung des Observers biometrische Merkmale des Wählers geprüft werden. Es ist auch möglich, dass der Observer einen zweiten Wert h'_V und den zugehörigen geheimen Wert z'_V gespeichert hat. Der Wähler würde dann an den Angreifer den Wert z'_V, den Observer und eine PIN weitergeben, die den Observer zum falschen Arbeiten und falschen Designated-Verifier-Beweis bezüglich z'_V bzw. h'_V veranlasst.

Zusammenfassend ist das Wahlsystem nicht erpressungsresistent, da ein Randomisations- und ein Enthaltungangriff möglich sind.

5.3 Ein effizientes allgemeines elektronisches Wahlsystem mit Observer

Das Wahlsystem aus Abschnitt 5.2 hat den Nachteil, dass jeder Wähler für alle Wahloptionen eine Verschlüsselung und einen WI-Wiederverschlüsselungsbeweis durchführen muss. Darüber hinaus liegt der Aufwand für die Auszählung bei $O\left(\sqrt{n_V}^{n_L-1}\right)$. In diesem Abschnitt wird ein observerbasiertes Wahlsystem mit Unüberprüfbarkeit vorgestellt, das die Kommunikation zwischen Wahlausschuss und Wählern deutlich reduziert. Damit nur gültige Stimmen in die Auszählung eingehen, können die Geheimtexte nicht ohne vorherige Entschlüsselung verrechnet werden. Sie müssen ein robustes, verifizierbares MIX-Netz (siehe z. B. [DK00]) durchlaufen, um die Anonymität der Stimmen sicherzustellen. Diese noch probabilistisch verschlüsselten Stimmen werden nun verteilt verifizierbar entschlüsselt und dabei deterministisch, also mit einem festen Wert, geblendet (vgl. Abschnitt 3.10). Das gleiche Verfahren wird mit den standardverschlüsselten Wahloptionen durchgeführt. Die abgegebenen deterministisch geblendeten Stimmen können nun überprüft werden, ob sie gültigen Wahloptionen entsprechen. Ungültige Stimmen werden ingoriert, die gültigen können einfach ausgezählt werden.

5.3.1 Anforderungen

Dem Wahlsystem können verschiedene kryptografische Primitive zugrunde liegen. Dabei müssen sie allerdings folgende Grundanforderungen erfüllen:

- *Existenz einer PKI:* Dem Protokoll muss eine Public-Key-Infrastruktur (PKI) zugrunde liegen.
- *Non-malleability:* Das verwendete Verschlüsselungsverfahren muss non-malleable sein (vgl. Abschnitt 2.3), damit es einem Angreifer nicht möglich ist, abgegebene Stimmen zu kopieren oder sinnvoll[7] modifiziert als seine eigene abzugeben. Andernfalls wäre die Unabhängigkeit der Stimmabgabe nicht gewährleistet.
- *Probabilistische Wiederverschlüsselung:* Um die Unüberprüfbarkeit zu ermöglichen, benötigt man einen probabilistischen Wiederverschlüsselungsalgorithmus R, so dass die Wiederverschlüsselung über der Menge der möglichen Chiffretexte gleichverteilt ist.
- *Existenz eines Designated-Verifier-Wiederverschlüsselungsbeweises:* Es wird ein effizienter Designated-Verifier-Beweis benötigt (vgl. Abschnitt 3.5), der zeigt, dass ein Geheimtext eine Wiederverschlüsselung eines anderen darstellt.
- *Kenntnis des geheimen Schlüssels:* Jeder Wähler wählt einen öffentlichen Schlüssel. Dabei muss sichergestellt werden, dass er seinen dazugehörigen privaten Schlüssel kennt. Dies ist essentiell für die Designated-Verifier-Eigenschaft des Beweises und somit für die Unüberprüfbarkeit des Wahlsystems.
- *Observer:* Jeder Wähler besitzt eine manipulationssichere Hardware, einen Observer, der in der Lage ist, R, den Designated-Verifier-Beweis und mit dem Wähler zusammen die Zero-Knowledge-Beweise der non-malleability durchzuführen. Auf dem Observer muss dazu der Public-Key des Wahlausschusses, der Signaturschlüssel des Wählers und der öffentliche Schlüssel des Wählers gespeichert sein. Der Observer muss außerdem mit Hilfe des Signaturschlüssels qualifizierte digitale Signaturen erstellen können. Nach der Berechnung der Signatur ist das Protokoll für den Observer abgeschlossen und er löscht sämtliche berechneten Werte.
- *Existenz eines Schwarzen Brettes:* Die Stimmen werden an ein Schwarzes Brett geschickt, auf dem jeder Leserechte und das Recht besitzt, Daten anzuhängen, aber keiner über das Recht verfügt, Daten zu löschen oder zu ändern.
- *Sicherheit der Verschlüsselung:* Der Wahlausschuss besteht aus $n \geq t$ Personen. Für jede Gruppe von weniger als t Personen des Wahlausschusses muss es unmöglich sein, einen Chiffretext zu entschlüsseln.
- *Verifizierbares, robustes MIX-Netz:* In der Auszählungsphase wird ein MIX-Netz benötigt, das die Geheimtexte der Wahlberechtigten verifizierbar permutiert und wiederverschlüsselt,

[7] So dass die Klartexte in einer dem Angreifer bekannten Relation stehen.

so dass der Zusammenhang zwischen der Wahlberechtigung bzw. dem Wähler und der Stimme nicht mehr nachvollziehbar ist.

- *Verteilte Entschlüsselung mit deterministischer Blendung:* Eine Gruppe von t Personen des Wahlausschusses muss die probabilistisch verschlüsselten Stimmen verifizierbar entschlüsseln und dabei deterministisch blenden können.

- *Stimmabgabe in einer virtuellen Wahlkabine:* Die Kommunikation zwischen Observer und Wähler darf nicht durch Angreifer überwacht werden können. Wähler und Observer befinden sich sozusagen in einer virtuellen Wahlkabine.

5.3.2 Allgemeiner Ablauf des Wahlsystems

Registrierungsphase

Im Registrierungsbüro erhält der Wähler seinen Observer, nachdem er sich gegenüber den Registrierungsautoritäten authentifiziert hat.
Der Wähler generiert sich einen geheimen Wert und berechnet den dazugehörigen öffentlichen Schlüssel, der mit dem öffentlichen Schlüssel der Auszählungsautoritäten und dem geheimen Signaturschlüssel des Wählers auf dem Observer gespeichert wird. Der öffentliche Signaturschlüssel wird von den Registrierungsautoritäten veröffentlicht. Die Liste der wahlberechtigten Signaturschlüssel wird, wie auch die Liste der möglichen Wahloptionen, von den Registrierungsautoritäten digital signiert und veröffentlicht.

Wahlphase, Auszählung

Der Wähler verschlüsselt (non-malleable) die gewünschte Wahloption unter dem öffentlichen Schlüssel der Auszählungsautoritäten. Er sendet sie an den Observer, der sie wiederverschlüsselt und einen nicht-interaktiven Designated-Verifier-Beweis der korrekten Wiederverschlüsselung erstellt. Der Observer sendet die wiederverschlüsselte Wahloption und den Beweis an den Wähler, welcher den Beweis verifiziert. Der Wähler lässt den Observer die Stimme signieren, prüft die Signatur und sendet die signierte und verschlüsselte Stimme an das Schwarze Brett.

Jeder kann die Signatur und den Beweis der non-malleability auf dem Schwarzen Brett überprüfen. Die Geheimtexte, die akzeptiert wurden und mit wahlberechtigten gültigen Signaturen versehen waren, werden durch ein verifizierbares MIX-Netz geführt. Dabei werden die Geheimtexte wiederverschlüsselt und zufällig permutiert, um den Zusammenhang mit den Signaturen zu verschleiern. Die Ausgabe des MIX-Netzes ist eine zufällige Anordnung probabilistisch verschlüsselter Stimmen. Diese werden nun entschlüsselt und dabei deterministisch geblendet. Jede Stimme, die einer den ebenfalls deterministisch geblendeten gültigen Wahloptionen entspricht, fließt in die Auszählung ein.

Im Folgenden wird ein konkretes Beispiel eines solchen effizienten observerbasierten Wahlsystems mit Unüberprüfbarkeit beschrieben und untersucht.

5.3.3 Ein Beispiel eines solchen Wahlsystems

Wahlvorbereitung

Die MIX-Server übernehmen hier die Rolle der Autoritäten $A_1, \ldots, A_n$ des Wahlausschusses. Sie bestimmen zusammen eine multiplikative Gruppe G mit Primzahlordnung $|G| =: q$ und ein erzeugendes Element g von G. Dann erzeugen sie gemeinsam ein ElGamal-Schlüsselpaar (s, h) mit $h = g^s$ und ein Wertepaar $\hat{s}, \hat{h} = g^{\hat{s}}$ (siehe Abschnitt 3.7.5), so dass jede Person des Wahlausschusses A_j einen Anteil s_j von s und einen Anteil $\hat{s}_j$ von $\hat{s}$ in einem (t, n) −Schwellensystem erhält und öffentlich auf diesen Anteil über $h_j = g^{s_j}$ bzw. $\hat{h}_j = g^{\hat{s}_j}$ festgelegt wird.

Durch das Schwellensystem der Verschlüsselung ist gewährleistet, dass nur t oder mehr Autoritäten zusammen eine mit dem öffentlichen Schlüssel h verschlüsselte Stimme entschlüsseln können. Der öffentliche Schlüssel h wird als öffentlicher Schlüssel des Wahlausschusses bekanntgegeben und auf den Observern gespeichert.

Registrierungsphase

Jeder Wahlberechtigte V_i, $(i = 1, \ldots, n_V)$, wird vom Wahlausschuss benachrichtigt und begibt sich zum Registrierungsbüro, wo er nach einer geeigneten Authentifikation einen Observer ausgehändigt bekommt.

Der Wähler V wählt ein Geheimnis $z_V \in_R \mathbb{Z}_q$ und berechnet $h_V = g^{z_V}$ als öffentlichen Anteil des Wertes z_V. Dieser Wert h_V wird ebenfalls auf dem Observer gespeichert. Wichtig ist, dass der Observer z_V selbst nicht kennt.

Auf dem Observer ist neben dem öffentlichen Schlüssel des Wahlausschusses und des Wertes h_V noch der geheime Signaturschlüssel des Observers und die Identität id_V des Wählers V gespeichert. Der dazu gehörende öffentliche Schlüssel ist auf eine Liste der wahlberechtigten Signaturen gespeichert. Der Wähler darf den geheimen Signaturschlüssel nicht kennen, da er sonst die Rolle des Observers selbst übernehmen könnte und das Wahlverfahren nicht mehr unüberprüfbar wäre.

Wahlphase

Es sei $\mathbf{L} = (m_1, \ldots, m_{n_L})$ die Menge der Wahlmöglichkeiten. Da in diesem System Wähler und Observer nicht nachweisen müssen, dass sie gültige Wahloptionen verschlüsselt haben, benötigt der Wähler in diesem Wahlsystem auch keine Standardverschlüsselung der Stimmen.

Jeder Wähler wählt Zufallszahlen $a, a' \in_R \mathbb{Z}_q$, seine Stimme m aus $\mathbf{L}$ und verschlüsselt $(x, y) = (g^a, h^a m)$. Diesen Geheimtext sendet der Wähler zusammen mit $g^{a'}$ an den Observer.

Der Observer wählt Zufallszahlen $b, b' \in \mathbb{Z}_q$ und verschlüsselt dann den Geheimtext erneut: $(x', y') = \left(g^b g^a, h^b h^a m\right)$. Er berechnet den Wert $g^{a'+b'}$ und daraus den für die non-malleability notwendigen Wert $b \cdot \mathcal{H}\left(g, x', y', g^{a'+b'}, id_V\right) + b'$.
Dann sendet er $\left(x', y', g^{a'+b'}, b \cdot \mathcal{H}\left(g, x', y', g^{a'+b'}, id_V\right) + b', id_V\right)$ an den Wähler.

Wenn der Observer korrekt gearbeitet hat, kann der Wähler daraus $\left(g^b, h^b\right)$ berechnen. Den nicht-interaktiven Zero-Knowledge-Beweis für die non-malleability bzw. Unabhängigkeit kann

der Wähler nun, wie bereits in Gleichung 5.2.3 gezeigt, erstellen.

Der Wähler muss sich sicher sein, dass der Observer korrekt verschlüsselt hat. Dies beweist der Observer mittels eines Designated-Verifier-Beweises wie es in Abschnitt 3.5 gezeigt wurde. Der Wähler lässt den Observer die verschlüsselte Stimme und den Zero-Knowledge-Beweis der non-malleability signieren, prüft die Signatur und schickt dies alles an das Schwarze Brett.

Das Schwarze Brett ist öffentlich lesbar, d. h. die Signaturen sind mit Hilfe der Liste der Wahlberechtigten öffentlichen Verifikationsschlüssel global verifizierbar. Ebenfalls sind die Zero-Knowledge-Beweise global verifizierbar, d. h. jeder kann die Wahlberechtigungen und die Unabhängigkeit der Stimmabgaben überprüfen.

Auszählungsphase

Nach Beendigung der Wahlphase werden die Signaturen überprüft. Stimmen mit ungültigen Signaturen und Stimmen ohne gültigen Zero-Knowledge-Beweis der Kenntnis der zur Verschlüsselung verwendeten Zufallszahlen werden ignoriert. Entsprechend einer Policy wird für jeden Wahlberechtigten nur eine Stimme gewertet, z. B. die erste abgegebene gültige Stimme[8]. Die nach diesen Überprüfungen verbliebenen Geheimtexte durchlaufen eine verifizierbare MIX-Kaskade, die die Stimmen permutiert und dabei unter h probabilistisch wiederverschlüsselt (vgl. Abschnitt 3.8.1). Die vom MIX-Netz ausgegebenen Geheimtexte werden, wie in Abschnitt 3.10 beschrieben[9], verifizierbar verteilt entschlüsselt und dabei deterministisch mit $\hat{s}$ geblendet. Die mit der „Zufallszahl" 0 standardverschlüsselten Wahloptionen aus der Liste $\mathbf{L}$ werden ebenfalls „entschlüsselt" und dabei deterministisch unter $\hat{s}$ geblendet. Schließlich werden die gültigen Stimmen, also die Werte, die in ihrer deterministischen Blendung jeweils einer deterministisch geblendeten gültigen Wahloption entsprechen, öffentlich ausgezählt.

5.3.4 Analyse des Wahlsystems

Wahlberechtigung

Die Überprüfung der Wahlberechtigung wird durch die digitale Signatur sichergestellt. In der Registrierungsphase werden die öffentlichen Signaturschlüssel der Wahlberechtigten auf einer Liste zertifiziert. Mit Hilfe der Zertifikate kann jeder feststellen, welche Nachrichten von Wahlberechtigten signiert wurden. Unberechtigt abgegebene Nachrichten werden nicht berücksichtigt.

Einmaligkeit

Wenn nur die zuerst veröffentlichten Stimmen mit korrekter Signatur in das MIX-Netz eingehen und eventuell später veröffentlichte Stimmen der gleichen Wähler für ungültig erklärt und verworfen werden, wird durch die Verifizierbarkeit des MIX-Netzes sichergestellt, dass jeder Wähler nur einmal seine Stimme abgeben kann. Die Einmaligkeit basiert auf der Fälschungssicherheit des verwendeten Signaturschemas.

[8] Dazu werden die Stimmen zeitgestempelt, sobald sie das Schwarze Brett erreichen.

[9] Man kann das in Abschnitt 3.10 beschriebene Verfahren mit $g = g_1 = g_2$ und $s = s_1 = s_2$ vereinfachen und so leicht auf die hier verwendete Standard-ElGamal-Verschlüsselung anpassen.

Fälschungssicherheit

Die Stimme eines Wähler ist fälschungssicher, da jeder Wähler die von ihm veröffentlichten Nachrichten digital signiert. Darüber hinaus kann kein Wähler seine Stimme nach deren Abgabe noch abändern, da nur die ersten veröffentlichten Stimmen gewertet werden. Die Fälschungssicherheit der Stimme leitet sich also direkt von der Fälschungssicherheit der digitalen Signatur ab.

Verifizierbarkeit

Da das Schwarze Brett öffentlich lesbar ist, kann jeder die Signaturen und Zero-Knowledge-Beweise überprüfen. Während der Auszählungsphase durchlaufen die Stimmen die MIX-Kaskade. Anschließend liegen die abgegebenen Stimmen im Klartext vor und sind öffentlich auszählbar. Das bedeutet, dass sich die Verifizierbarkeit des Wahlverfahrens aus der Verifizierbarkeit der Arbeit des MIX-Netzes ergibt.

Korrektheit

Die Korrektheit der Auszählung ist garantiert, wenn alle Wähler die Stimme ihrer Wahl abgeben können, also jeweils die Korrektheit der Verschlüsselung des Observers nachvollziehen können. Dies wird durch die Designated-Verifier-Beweise, die Verifizierbarkeit des MIX-Netzes und die öffentliche Auszählbarkeit der Klartextstimmen gewährleistet.

Ehrlichkeit, Robustheit

Ein unehrlicher Wähler kann keine ungültige Stimme in die Auszählung einfließen lassen, da diese am Ende des MIX-Netzes im Klartext vorliegt und ignoriert wird.

Die Verifizierbarkeit der Aktionen der MIX-Server ermöglicht es, betrügerische Autoritäten zu identifizieren und auszuschließen. Solange höchstens $n_A - t$ Personen des MIX-Netzes unehrlich sind, kann die Wahl mit Ausschluss dieser MIX-Server durchgeführt werden. Daher ist das Wahlverfahren robust.

Wahlaufwand

Im Unterschied zu [MBC01] und [Sch05a] muss kein Wiederverschlüsselungsbeweis für alle Wahloptionen durchgeführt werden. Daher ist der Aufwand auf Seiten des Wählers deutlich geringer.

Die weitere Komplexität der Kommunikation hängt stark von den verwendeten Beweisen, also vom Designated-Verifier-Beweis und vom Zero-Knowledge-Beweis der Unabhängigkeit ab. Diese Beweise sind effizient durchführbar und der Kommunikationsaufwand ist im Gegensatz zum Wahlverfahren von Hirt und Sako [HS00] unabhängig von der Anzahl der Autoritäten und im Gegensatz zu [Sch05a] auch unabhängig von der Anzahl der Wahloptionen! Darüber hinaus muss vom Wähler und vom Observer nur die Stimme und nicht jede Wahloption verschlüsselt werden.

Wahlgeheimnis, Anonymität

Die Anonymität eines jeden Wählers ist garantiert, wenn die verschlüsselte Stimme nicht durch eine außenstehende Person oder eine Gruppe von weniger als t Personen des Wahlausschusses entschlüsselt werden kann. Das trifft für dieses Wahlsystem zu.

Jeder kann feststellen, wer eine Stimme abgegeben hat, aber nur eine Menge von mindestens t MIX-Servern kann die Entschlüsselung der Wahloptionen vornehmen, um festzustellen, wie ein einzelner Wähler gewählt hat. Solange nicht mehr als $t - 1$ Personen des Wahlausschusses kooperieren, ist die Anonymität durch die Sicherheit des Verschlüsselungsverfahrens gewährleistet.

Unabhängigkeit

Es ist nicht möglich, eine Stimme eines anderen Wählers zu kopieren, da man für die verschlüsselte Wahloption einen Zero-Knowledge-Beweis liefern muss, die zur Verschlüsselung verwendete Zufallszahl zu kennen. Durch diese Eigenschaft der non-malleability ist es nicht möglich, Stimmen abzugeben, die in einer dem Angreifer bekannten Relation zu Stimmen anderer Wähler stehen.

Unüberprüfbarkeit

Das Wahlsystem ist unüberprüfbar, d. h. es ist nicht möglich, dass der Wähler einen Beleg erstellen kann, wie er gewählt hat. Die Unüberprüfbarkeit in diesem System basiert auf dem Einsatz des Observers, der einen Teil der zur Verschlüsselung verwendeten Zufallszahl generiert. Diese Zufallszahl ist dem Wähler unbekannt und kann daher auch nicht als Beleg für die abgegebene Stimme gelten. Der Designated-Verifier-Beweis ist nicht übertragbar. Der Wähler hat also keine Möglichkeit, andere beweisbar davon zu überzeugen, welche Wahloption in dem Geheimtext verborgen ist. Andererseits kann der Wähler zwar eine Stimme der Gestalt „Ich bin Wähler V und stimme für Partei A" verschlüsseln und abgeben, die Stimme wird aber in der Auszählung nicht in diesen Klartext entschlüsselt, sondern lediglich als ungültige Stimme aussortiert.

Erpressungsresistenz

Das Wahlsystem ist sicher gegen den Randomisationsangriff, da nur eine Wahloption verarbeitet wird.

Ein Impersonationsangriff kann abgewehrt werden, indem man zur Benutzung des Observers eine Authentifikation des Wählers gegenüber dem Observer vorsieht (vgl. Abschnitt 5.2.4).

Sofern der Erpresser die Zuordnung von Signaturschlüssel und Wähler bzw. Observer kennt, kann er feststellen, *ob* ein Wähler gewählt hat. Verlangt der Angreifer sogar vor der Wahlphase die Herausgabe des Observers, kann der Wähler nicht mehr abstimmen. Ein Enthaltungsangriff ist somit möglich.

Das Wahlsystem ist also nicht erpressungsresistent, da ein Enthaltungangriff möglich ist.

5.4 Vergleichende Analyse

Die in den Abschnitten 5.2 und 5.3 vorgestellten Wahlsysteme erfüllen die meisten Anforderungen an demokratische elektronische Wahlen (vgl. [Eur05]). Sie sind effizient und insbesondere bei dem in Abschnitt 5.3 beschriebenen System liegt der Wahlaufwand auf Seiten des Wählers in einem akzeptablen Rahmen.

Die Wahlsysteme sind allerdings nicht erpressungsresistent (vgl. [JCJ05]), da man durch die digitale Signatur feststellen kann, wer eine Stimme abgegeben hat. Nicht für alle muss die Erpressungsresistenz gefordert werden. Beispielsweise ist im Protection Profile des BSI für Personal- und Betriebsratswahlen (vgl. [BSI06]) kein solch hohes Maß an Sicherheit gefordert. In diesen Fällen sind die beiden hier vorgestellten Wahlschemata, effiziente sichere Systeme.

Es gibt aber Wahlen, bei denen eine solche Erpressbarkeit auszuschließen ist. Das in Abschnitt 5.3 vorgestellte Wahlsystem ist bereits sicher gegenüber dem Impersonations- und dem Randomisation-Angriff. Der Zwang zur Enthaltung ist ein Angriff, der auch in den bisher bekannten Systemen wie z. B. [HS00] oder [MBC01] durchführbar ist. Durch den in [JCJ05] verwendeten Ansatz wählerbezogener Credentials statt digitaler Signaturen zur Überprüfung der Wahlberechtigung lässt sich dieser Angriff verhindern. Daher wird in Kapitel 6 analysiert, wie sich Wahlsysteme mit Observer mit wählerbezogenen Credentials kombinieren lassen.

Denn die Gleichheit besteht darin, dass
Arme und Reiche in gleicher Weise regieren,
dass nicht Einzelne allein entscheiden,
sondern alle gleichmäßig ihrer Zahl nach.

Aristoteles

Kapitel 6

Ein allgemeines erpressungsresistentes elektronisches Wahlsystem mit Observer

Juels, Catalano und Jakobsson haben in [JCJ05] neue Angriffsmöglichkeiten auf die Sicherheit elektronischer Wahlen beschrieben. Ein Angreifer kann das Ziel haben, einen Wähler zu zwingen, nicht zu wählen. Weitere Angriffsszenarios sind der Randomisation-Angriff und der Impersonationsangriff. Letzterer kann durch zusätzliche Anforderungen an den Observer erfolgreich verhindert werden. Um den beiden ersten Angriffen - insbesondere dem zweiten - zu begegnen, muss die Verifikation der Wahlberechtigung entscheidend geändert werden. Wenn jeder nachvollziehen kann, wer gewählt hat, dann kann man einen Wähler auch zwingen, der Wahl fern zu bleiben. Die Wahlberechtigung muss also anonym bzw. pseudonym geprüft werden, d. h. man muss die Berechtigung gegenüber einer Pseudonymliste prüfen, die keine Rückschlüsse auf den Wähler zulässt. Da ein Angreifer jedoch den Wähler zwingen kann, sein Pseudonym zwischen Registrierung und Wahlphase zu verraten, muss es für den Wähler eine Möglichkeit geben, ein falsches Pseudonym an den Angreifer zu geben, ohne dass dieser das erkennt. Digitale Signaturen sind für diesen Zweck nicht geeignet, da die sonst gewünschte Unfälschbarkeit der Signaturen das Fälschen der Pseudonyme verhindert. Juels et. al haben daher vorgeschlagen, zu diesem Zweck von der Registrierungsbehörde gewählte, probabilistisch verschlüsselte und signierte Zufallszahlen als Credentials zu verwenden. Der Wähler erhält seine Zufallszahl als Klartext, während die zugehörigen probabilistisch verschlüsselten Werte auf einer Wählerliste vom Registrierungsbüro digital signiert veröffentlicht werden.

In diesem Kapitel wird ein allgemeines Wahlprotokoll mit Observer konstruiert, das alle Sicherheitsanforderungen an demokratische, elektronische Wahlen - einschließlich der Erpressungsresistenz - erfüllt. Das Verfahren basiert auf einem von mir in [Sch06b] vorgestellten Wahlsystem. Es ist effizienter als das ursprünglich von Juels, Catalano und Jakobsson vorgestellte Schema, erfüllt aber die gleichen Sicherheitsanforderungen.

In Abschnitt 6.1 wird zunächst das allgemeine Konzept des Wahlsystems und die Anforderungen an die verwendeten Bausteine vorgestellt. Dann wird in Abschnitt 6.2 ein konkretes Beispiel eines solchen erpressungsresistenten elektronischen Wahlsystems mit Observer erläutert. In Abschnitt 6.3 wird die Sicherheit des Wahlsystems analysiert.

6.1 Ein allgemeines sicheres Wahlsystem mit Observer

In diesem Abschnitt wird das Wahlsystem vorgestellt - ohne aber die kryptografischen Bausteine konkret festzulegen. Durch die Annahme idealer einzelner Komponenten ist es teilweise möglich, die Sicherheit des Systems auf die Sicherheit der Komponenten zurückzuführen wobei diese durch andere Komponenten mit gleichen Sicherheitseigenschaften austauschbar sind.

6.1.1 Grundsätzliche Anforderungen

- *Existenz einer PKI:* Dem Protokoll muss eine Public-Key-Infrastruktur (PKI) zugrunde liegen.
- *Non-malleability:* Das verwendete Verschlüsselungsverfahren für die Stimme bzw. das Credential muss non-malleable sein (vgl. Abschnitt 2.3), damit es einem Angreifer nicht möglich ist, abgegebene Stimmen zu kopieren oder modifiziert als seine eigene abzugeben. Andernfalls wäre die Unabhängigkeit der Stimmabgabe nicht gewährleistet.
- *Kommutative homomorphe Verschlüsselungen:* Die zur Verschlüsselung der Credentials eingesetzten Kryptosysteme müssen kommutativ sein, das heißt, dass die Reihenfolge der Verschlüsselungen und die Reihenfolge der Entschlüsselungen irrelevant sind. Darüber hinaus müssen sie homomorph sein. In Abschnitt 2.4.1 wurde ein solches Verschlüsselungsverfahren vorgestellt.
- *Simulierbarkeit im Rahmen von* $\boldsymbol{Exp}^{E.-res}_{\mathsf{WS},X}$ *und* $\boldsymbol{Exp}^{p.-E.-res.}_{\mathsf{WS},X'}$: Um die Erpressungresistenz nachzuweisen, müssen an das Verschlüsselungsschema noch weitere Anforderungen gestellt werden. Im Beweis der Erpressungsresistenz erhält der Simulator Werte, von denen er nicht weiß, ob sie eine vorgegebene Struktur aufweisen (z. B. ein Diffie-Hellman-Quadrupel) oder zufällig gewählt wurden. Der Simulator muss in beiden Fällen Stimmen bzw. Credentials entschlüsseln können, deren Verschlüsselung auf diesen Werten basiert. Dies ist beispielsweise mit dem Cramer-Shoup-Cryptosystem [CS98] oder mit dem in Abschnitt 6.2 verwendeten modifizierten-ElGamal-Verschlüsselungsschema (vgl. Abschnitt 2.4) möglich. Das herkömmliche ElGamal-Verschlüsselungsschema erlaubt dies jedoch nicht.
  Ausgenommen von dieser Anforderung ist die Verschlüsselung für das MIX-Netz $\tilde{\mathcal{A}}$ (vgl. Abschnitt 6.1.2).
- *Probabilistische Wiederverschlüsselung:* Damit Unüberprüfbarkeit erreicht werden kann, muss das Verschlüsselungsverfahren, mit dem die Wahloption verschlüsselt wird, eine probabilistische Wiederverschlüsselung R ermöglichen. Die Wiederverschlüsselung muss über der Menge der möglichen Chiffretexte gleichverteilt sein.
- *Existenz eines 1-von-n_L Witness-Indistinguishable-Wiederverschlüsselungsbeweises für zwei Prover:* Basierend auf R existiert ein 1-von-n_L WI-Wiederverschlüsselungsbeweis für zwei Prover (vgl. Abschnitt 3.6.1), dass ein Chiffretext c_k aus einer Liste von Chiffretexten $c_1, \ldots, c_{n_L}$ tatsächlich eine Wiederverschlüsselung des Chriffretextes c ist, ohne k zu offenbaren.

- *Existenz eines Designated-Verifier-Wiederverschlüsselungsbeweises:* Es wird ein effizienter Designated-Verifier-Beweis benötigt (vgl. Abschnitt 3.5), der dem Wähler als Designated-Verifier beweist, dass ein Geheimtext eine gültige Wiederverschlüsselung eines anderen Geheimtextes darstellt.

- *Kenntnis des geheimen Schlüssels:* Jeder Wähler berechnet zwei öffentliche Schlüssel, wobei sichergestellt werden muss, dass er jeweils seinen zugehörigen privaten Schlüssel kennt.

- *Observer:* Jeder Wähler besitzt eine manipulationssichere Hardware, einen Observer, der die Wiederverschlüsselung R, den Designated-Verifier-Beweis und den Zero-Knowledge-Beweis der Kenntnis der zur Verschlüsselung verwendeten Zufallszahlen (non-malleability) durchführen kann. Zudem muss der Observer mit dem Wähler zusammen einen 2-Prover 1-von-n_L WI-Beweis der korrekten Wiederverschlüsselung erstellen können. Auf dem Observer müssen die Public-Keys der Autoritäten, das verschlüsselte Credential des Wählers, das ungültige verschlüsselte Credential und der für den Designated-Verifier-Beweis benötigte öffentliche Wert des Wählers gespeichert sein. Der Observer darf nur nach PIN-Eingabe arbeiten. Erhält er die korrekte PIN, so muss er mit dem korrekten Credential-Geheimtext arbeiten und sich korrekt verhalten. Wenn er mit einer anderen PIN benutzt wird, muss er den gespeicherten ungültigen Credential-Geheimtext verwenden. Nach dem Abschluss des Protokolls löscht der Observer sämtliche berechneten Werte.

- *Abhörsichere Kommunikation von den Registrierungsautoritäten zum Wähler:* Im Registrierungsbüro dürfen die Nachrichten von den Registrierungsautoritäten zum Wähler nicht abgehört werden können.

- *Stimmabgabe in einer virtuellen Wahlkabine:* Die Kommunikation zwischen Observer und Wähler darf während der Registrierungsphase und während der Wahlphase nicht durch Angreifer überwacht werden können. Wähler und Observer befinden sich sozusagen in einer virtuellen Wahlkabine.

- *Existenz eines Schwarzen Brettes:* Die Stimmen werden an ein Schwarzes Brett geschickt, auf dem jeder Leserechte und das Recht besitzt, Daten anzuhängen, aber keiner über das Recht verfügt, Daten zu löschen oder zu ändern.

- *Anonymer Kanal vom Wähler zum Schwarzen Brett:* Die vom Wähler an das Schwarze Brett versendeten Nachrichten dürfen nicht eindeutig dem Wähler zugeordnet werden können. Dies kann durch den Einsatz eines verifizierbaren robusten MIX-Netzes geschehen. Möchte man die Geheimhaltung der Stimmen dauerhaft sichern, so muss die Anonymität durch organisatorische Maßnahmen (Wahlkiosk, Wahlmöglickeiten von jedem geeigneten Terminal) ergänzt werden.

- *Verifizierbare, robuste Entschlüsselungs-MIX-Netze:* Zur Veröffentlichung der Liste wahlberechtigter verschlüsselter Credentials nach der Registrierung, bei der Stimmabgabe und in der Auszählungsphase werden MIX-Netze benötigt, die verifizierbar entschlüsseln und dabei so permutieren, dass der Zusammenhang zwischen den Geheim- und Klartexten verborgen bleibt. Jeder der n MIX-Server besitzt jeweils einen Anteil am geheimen Schlüssel des MIX-Netzes. Es darf jedoch nicht möglich sein, dass ein Zusammenschluss von weniger als t MIX-Servern die Nachrichten entschlüsselt. Das MIX-Netz sollte robust sein, damit es leicht möglich ist, bis zu $n - t$ korrupte MIX-Server zu erkennen und auszuschließen.

Im Folgenden wird der Ablauf eines solchen allgemeinen, erpressungsresistenten Wahlsystems mit Observer beschrieben und untersucht. In diesem Wahlsystem wird die Idee von Juels et al. aufgegriffen, Credentials anstelle von digitalen Signaturen zu verwenden. In [JCJ05] wird die Erzeugung der Credentials nur unzureichend beschrieben, so dass dieses System nicht mehr erpressungsresistent ist, sobald es nur eine einzelne ggf. korrupte Registrierungsautorität gibt. Diese Lücke wird geschlossen. Das hier vorgestellte System ist zudem effizienter, da es auf einen der beiden sehr aufwändigen Tests der Gleichheit von zugrundeliegenden Klartexten verzichtet. Auf die Durchführung des anderen Tests kann darüber hinaus auch verzichtet werden, wenn man stattdessen das in [Smi05] vorgestellte Verfahren (vgl. Abschnitt 3.10) verwendet. Der Einsatz des Observers stellt eine praktikable Lösung für den sicheren Transport der Credentialgeheimtexte durch den Wähler dar.

6.1.2 Wahlvorbereitung

Die Registrierungsautoritäten $R_1, \ldots, R_{n_R}$ erzeugen ein Signaturschlüsselpaar (SK_R, PK_R) gemeinsam, so dass jede Person R_j der Registrierungsautoritäten ihren Anteil sk_j^R von SK_R in einem $(t_R, n_R)-$Schwellensystem erhält und öffentlich auf diesen Anteil festgelegt wird.

Das Wahlsystem ist auf die Wahl einer Stimme aus n_L Kandidaten ausgelegt. Dies stellt keine Einschränkung da, da selbst komplizierte Kombinationen oder Reihenfolgen von Kandidaten als mögliche Wahloptionen dargestellt werden können. Um Wahlen erpressungsresistent durchführen zu können, muss man allerdings ausschließen, dass Wähler selbst Wahloptionen auf dem Wahlschein vorschlagen und eintragen können. Das heißt, die Liste der Wahloptionen muss feststehen, und der Wähler darf lediglich eine Wahloptionen auswählen.

Die Registrierungsautoritäten veröffentlichen die von ihnen signierte Liste $\mathbf{L}=(m_1, \ldots, m_{n_L})$ der möglichen Wahloptionen. Da die Stimmen am Ende durch das MIX-Netz $\mathcal{A}'$ entschlüsselt und einzeln im Klartext ausgegeben werden, kann eine beliebige Darstellung der Wahloptionen gewählt werden.[1] Es muss lediglich sichergestellt sein, dass man das Auszählungsergebnis effizient und korrekt eindeutig ermitteln kann.

Es gibt neben der Gruppe $\mathcal{R}$ der Registrierungsautoritäten $R_1, \ldots, R_{n_R}$ drei weitere Gruppen von Autoritäten. Jede Gruppe besitzt einen öffentlichen Schlüssel. Die zugehörigen privaten Schlüssel sind auf die Gruppenmitglieder in einem Schwellensystem aufgeteilt (vgl. Abschnitt 3.7.5). Zu der Gruppe der Autoritäten $\mathcal{A} = \{A_1, \ldots, A_{n_A}\}$ gehört das Schlüsselpaar

[1] Man kann die Wahloptionen z. B. in einem Zahlensystem zur Basis n_V darstellen: $\mathbf{L} = (1, n_V, n_V^2, \ldots, n_V^{n_L-1})$ (siehe Abschnitt 2.2.6)

(SK_A, PK_A), zur Gruppe $\mathcal{A}' = \{A'_1, \ldots, A'_{n_{A'}}\}$ das Schlüsselpaar $(SK_{A'}, PK_{A'})$ und schließlich zum MIX-Netz $\tilde{\mathcal{A}} = \{\tilde{A}_1, \ldots, \tilde{A}_{n_{\tilde{A}}}\}$ das Schlüsselpaar $SK_{\tilde{A}}, PK_{\tilde{A}}$.

Durch das Schwellensystem zum Schwellenwert t ist gewährleistet, dass nur eine Gruppe bestehend aus t oder mehr Autoritäten zusammen eine mit dem öffentlichen Schlüssel verschlüsselte Nachricht entschlüsseln können. In der Variante des Wahlsystems, bei der im Rahmen der Auszählung (siehe Abschnitt 6.1.5) eine deterministische Blendung der Credentials (vgl. Abschnitt 3.10) eingesetzt wird, müssen die Autoritäten $\mathcal{A}'$ einen gemeinsamen geheimen Wert $\hat{s}$ erzeugen, wie es beispielsweise in Abschnitt 3.7.5 beschrieben wird. Diese öffentlichen Schlüssel werden zusammen mit den anderen Systemparametern bekanntgegeben und PK_A und $PK_{A'}$ werden auf den Observern gespeichert.

6.1.3 Registrierungsphase

Jeder Wahlberechtigte V_i, $(i = 1, \ldots, n_V)$, wird vom Wahlausschuss benachrichtigt und begibt sich zum Registrierungsbüro, wo er sich gegenüber den Registrierungsautoritäten authentifiziert und seinen Observer erhält. Der Wahlberechtigte wählt einen geheimen Wert und berechnet den dazu gehörigen öffentlichen Wert, den er für die Erstellung der Designated-Verifier-Beweise an die Registrierungsautoritäten gibt. Jede Autorität R_j $(1 \leq j \leq n_R)$ wählt einen zufälligen Wert $\sigma_{i,j}$, verschlüsselt diesen probabilistisch unter dem öffentlichen Schlüssel PK_A der Autoritäten $\mathcal{A}$ zum Wert $E_{PK_A}(\sigma_{i,j})$. Sie verschlüsseln diese Geheimtexte $E_{PK_A}(\sigma_{i,j})$ erneut probabilistisch, diesmal allerdings unter dem öffentlichen Schlüssel der Autoritäten $\mathcal{A}'$. Die beiden Verschlüsselungen müssen vertauschbar sein, man muss also auch die zuerst angewendete Verschlüsselung als erstes entfernen, das heißt entschlüsseln können. Daher wird der durch diese beiden Verschlüsselungen entstandene Geheimtext mit $E_{PK_{A'},PK_A}(\sigma_{i,j}) = E_{PK_A,PK_{A'}}(\sigma_{i,j})$ bezeichnet. Neben den beiden Geheimtexten erstellt jede Autorität einen Designated-Verifier-Beweis für den Wähler, dass die Verschlüsselung unter $PK_{A'}$ korrekt ist. Die beiden Geheimtexte und der Beweis werden geheim von jeder Autorität zum Wähler übertragen. Dieser verifiziert die Beweise. Korrupte Autoritäten haben zwar die Möglichkeit statt der Zufallszahlen für $\sigma_{i,j}$ vom Angreifer vorgegebene Werte zu wählen, aber wenn es mindestens eine ehrliche Autorität gibt, kann nur der Wähler V_i das verschlüsselte Credential $E_{PK_A}(\sigma_i)$ berechnen, das als seine Wahlberechtigung fungiert:

$$E_{PK_A}(\sigma_i) := \prod_{j=1}^{n_R} E_{PK_A}(\sigma_{i,j}) = E_{PK_A}\left(\prod_{j=1}^{n_R} \sigma_{i,j}\right).$$

Das verschlüsselte Credential wird vom Wähler auf dem Observer gespeichert. Die Autoritäten berechnen öffentlich

$$\prod_{j=1}^{n_R} E_{PK_A,PK_{A'}}(\sigma_{i,j}) = E_{PK_A,PK_{A'}}\left(\prod_{j=1}^{n_R} \sigma_{i,j}\right) = E_{PK_A,PK_{A'}}(\sigma_i).$$

Dieser Wert wird auf die Liste der wahlberechtigten Credentials gesetzt. Am Ende der Registrierungsphase werden die Werte durch das verifizierbare Entschlüsselungs-MIX-Netz $\mathcal{A}$ geschickt und als Ausgabe die Liste der wahlberechtigten verschlüsselten Credentials $E_{PK_{A'}}(\sigma_i)$ (für $1 \leq i \leq n_V$) veröffentlicht. Der Wahlberechtigte wählt einen weiteren geheimen Wert und

berechnet den dazu gehörigen öffentlichen Wert, den er während der Wahlphase für die Erstellung des Designated-Verifier-Beweises der korrekten Wiederverschlüsselung auf dem Observer speichert.

Im Unterschied zum geheimen Signaturschlüssel in Abschnitt 5.3 sollte der Wähler das verschlüsselte Credential kennen, um auch ohne Observer abstimmen zu können. Der Observer dient hier zum sicheren Speichern und Transport des verschlüsselten Credentials sowie zur Absicherung der Unüberprüfbarkeit. Der Angreifer kann zwar verlangen, dass ihm der Wähler das Credential übermittelt, der Wähler hat aber die Möglichkeit, ihm ein falsches Credential zu geben. Zu diesem Zweck wählt der Wahlberechtigte eine weitere Zufallszahl, die er probabilistisch unter dem öffentlichen Schlüssel PK_A verschlüsselt und als falsches Credential auf dem Observer speichert bzw. im Fall eines Angriffes dem Angreifer aushändigt.

6.1.4 Wahlphase

Jeder Wähler verschlüsselt (non-malleable) seine Stimme m aus der Liste der gültigen Wahloptionen $\mathbf{L}$ probabilistisch unter dem Schlüssel $PK_{A'}$ und sendet den Geheimtext an den Observer. Dieser wählt eine Zufallszahl und verschlüsselt damit den Geheimtext wieder unter dem öffentlichen Schlüssel $PK_{A'}$ und erhält $E_{PK_{A'}}(m)$. Außerdem verschlüsselt er das Credential des Wählers probabilistisch - ebenfalls wieder unter dem öffentlichen Schlüssel $PK_{A'}$. Für die Unabhängigkeit der Stimmabgabe und die Prüfung der Wahlberechtigung ist es entscheidend, dass für die non-malleability die Geheimtexte von Credential und Stimme nicht getrennt voneinander betrachtet werden. Schließlich sendet der Observer die berechneten Werte und je einen Designated-Verifier-Beweis der korrekten Verschlüsselung der Stimme und des Credentials an den Wähler. Der Wähler verifiziert die Beweise. Gemeinsam mit dem Observer erstellt er einen 2-Prover-Witness-Indistinguishable-Beweis, dass sie eine korrekte Wahloption verschlüsselt haben, da sonst die Unüberprüfbarkeit verloren ginge (vgl. Abschnitt 3.6.1). Der Wähler verschlüsselt die Geheimtexte von Stimme und Credential, den WI-Beweis und den Beweis der non-malleability unter $PK_{\tilde{A}}$ und sendet diese Nachricht über das verifizierbare robuste Entschlüsselungs-MIX-Netz $\tilde{\mathcal{A}}$ an das Schwarze Brett. Am Schwarzen Brett wird jeweils die Wahlberechtigung $E_{PK_{A'},PK_A}(\sigma_i)$ zusammen mit der unter $PK_{A'}$ verschlüsselten Wahloption, dem WI-Beweis und dem Beweis der non-malleability mit einem Zeitstempel (z. B. einer fortlaufenden Nummer) versehen.

6.1.5 Auszählungsphase

Die Auszählungsphase lässt sich in fünf Schritte unterteilen:

Beweise prüfen: Nach Beendigung der Wahlphase werden die Beweise der non-malleability bzw. Unabhängigkeit und die WI-Beweise, dass korrekte Wahloptionen enthalten sind, überprüft. Stimmen ohne gültige Beweise werden ignoriert.

Duplikate entfernen: Die Autoritäten $\mathcal{A}$ entschlüsseln für alle anderen Stimmen verifizierbar das verschlüsselte Credential $E_{PK_{A'},PK_A}(\sigma_i)$ zu $E_{PK_{A'}}(\sigma_i)$. Die verschlüsselten Credentials werden nun paarweise von den Autoritäten $\mathcal{A}'$ verifizierbar auf Duplikate geprüft. Entsprechend

einer Policy wird für jedes Credential nur eine Stimme gewertet, z. B. die letzte abgegebene Stimme. Dazu werden die Zeitstempel miteinander verglichen.

- *Variante mit einem Test auf gleiche zugrundeliegende Klartexte*:
  Diese Überprüfung kann mit Hilfe eines Tests auf gleiche Klartexte geschehen, wie sie in Abschnitt 3.9 beschrieben wird.
- *Variante mit einer deterministischen Blendung der Credentials*:
  Effizienter ist die in Abschnitt 3.10 vorgestellte Methode der Entschlüsselung und gleichzeitigen deterministischen Blendung der Credentials σ_i zu Werten $\sigma_i^{\hat{s}}$.

MIX-Netzwerk $\mathcal{A}'$: Die duplikatfreie Liste der unter $PK_{A'}$ verschlüsselten Credentials und Stimmen wird durch das verifizierbare robuste MIX-Netzwerk $\mathcal{A}'$ geschickt, das die Paare aus Credential und Stimme beibehält, die Anordnung der Paare allerdings zufällig verändert und dabei Credential und Stimme schrittweise entschlüsselt. Die verschlüsselten Credentials durchlaufen dabei synchron das gleiche MIX-Netzwerk mit der gleichen Permutation. Die Ausgabe ist eine zufällig permutierte Liste der Klartextstimmen und der dazu gehörenden Klartextcredentials. Die Liste der nach der Registrierungsphase veröffentlichten wahlberechtigten und unter $PK_{A'}$ verschlüsselten Credentials wird ebenfalls verifizierbar von den Autoritäten $\mathcal{A}'$ entschlüsselt.

Credentials prüfen: Schließlich werden die Klartexte eines jeden abgegeben Credentials mit den Klartexten aus der Liste der wahlberechtigten Credentials verglichen. Die Stimmen und Credentials, die nicht wahlberechtigt sind, werden ignoriert.

Auszählung: Die verbleibenden gültigen Stimmen von wahlberechtigten Wählern können dann von jedem ausgezählt werden.

6.2 Beispiel eines erpressungsresistenten elektronischen Wahlsystems mit Observer

Betrachtet man die an das Wahlsystem gestellten Anforderungen, so erkennt man, dass die kanonische Wahl von ElGamal als Verschlüsselungsschema aus beweistechnischen Gründen nicht möglich ist. Um die Erpressungsresistenz des Wahlsystems unter der Diffie-Hellman-Entscheidungsannahme nachweisen zu können, muss ein spezielles Verschlüsselungsschema eingesetzt werden. Der Simulator muss im Beweis auch dann entschlüsseln können (siehe Seite 120), wenn die dem Simulator zu Beginn gestellte Diffie-Hellman-Challenge kein Diffie-Hellman-Quadrupel ist (siehe Seite 118). Daher wird in diesem Beispiel die in Abschnitt 2.4 beschriebene modifizierte-ElGamal-Verschlüsselung eingesetzt. Entsprechend werden die kryptografischen Bausteine und Beweise verwendet, die auf diesem modifizierten-ElGamal-Schema basieren und in den Kapiteln 2 und 3 vorgestellt wurden.

6.2.1 Wahlvorbereitung

Die Autoritäten bestimmen zusammen eine multiplikative Gruppe G mit Primzahlordnung q und drei erzeugende Elemente g, g_1, g_2 sowie eine kryptografische Hashfunktion $\mathcal{H}$. Anschließend erzeugen die Autoritäten aus $\mathcal{A}$ einen ElGamal-Schlüssel, bestehend aus den geheimen Schlüsseln s_1, s_2 und dem öffentlichen Schlüssel

$$g_1,\ g_2,\ h = g_1^{s_1} g_2^{s_2}$$

in einem $(t_A, n_A)-$Schwellensystem (vgl. Abschnitt 3.7.6). Der geheime Schlüssel s_1, s_2 wird dabei nicht explizit berechnet und ist auf die Autoritäten $A_1, \ldots, A_{n_A}$ aufgeteilt (vgl. Abschnitt 3.7.5). Entsprechend erzeugen die Autoritäten $A'_1, \ldots, A_{n_{A'}}$ Schlüssel

$$s'_1,\ s'_2,\ h' := g_1^{s'_1} g_2^{s'_2}$$

und die MIX-Server $\tilde{A}_1, \ldots, \tilde{A}_{n_{\tilde{A}}}$ das ElGamal-Schlüsselpaar

$$\tilde{s},\ \tilde{h} = g^{\tilde{s}}.$$

Die Registrierungsautoritäten $R_1, \ldots, R_{n_R}$ erzeugen ein Signaturschlüsselpaar s_R, $h_R := g^{s_R}$. Die Systemparameter $\mathcal{H}$, q, g, g_1, g_2 und die öffentlichen Schlüssel h, h' und $\tilde{h}$ werden veröffentlicht und auf den n_V Observern der Wahlberechtigten gespeichert. Der öffentliche Signaturschlüssel h_R der Registierungsautoritäten wird veröffentlicht.

Die Registrierungsautoritäten veröffentlichen die von ihnen unter s_R signierte Liste

$$\mathbf{L} = \left(m_1 = 1, \ldots, m_{n_L} = n_V^{n_L - 1}\right)$$

der möglichen Wahloptionen (vgl. Abschnitt 2.5.4).

In der Variante des Wahlsystems, bei der während der Auszählung (siehe Abschnitt 6.2.4) eine deterministischen Blendung der Credentials (vgl. Abschnitt 3.10) eingesetzt wird, müssen die Autoritäten $\mathcal{A}'$ einen gemeinsamen geheimen Wert $\hat{s} \in \mathbb{Z}_q$ erzeugen, wie es in Abschnitt 3.7.5 beschrieben wird.

6.2.2 Registrierungsphase

Die Wahlberechtigten V_i $(1 \leq i \leq n_V)$ werden benachrichtigt und begeben sich zum Registrierungsbüro. Dort authentifizieren sie sich gegenüber den Registrierungsautoritäten. Sie wählen sich eine Zufallszahl $z_{1,i} \in_R \mathbb{Z}_q$ und berechnen $h_{1,i} := g^{z_{1,i}}$. Der Wert $h_{1,i}$ wird den Registrierungsautoritäten mitgeteilt. Jede Autorität R_j $(1 \leq j \leq n_R)$ wählt zufällige Werte $\sigma_{i,j}, a_{i,j}, b_{i,j} \in \mathbb{Z}_q$ und erstellt die Verschlüsselung

$$(x_{1,i,j}, x_{2,i,j}, y_{i,j}) := (g_1^{a_{i,j}}, g_2^{a_{i,j}}, h^{a_{i,j}} \sigma_{i,j}),$$

die sie geheim an den Wähler überträgt. Dann erstellt sie die mit der obigen Verschlüsselung kommutative Verschlüsselung (vgl. Abschnitt 2.4.1)

$$\left(x_{1,i,j}, x_{2,i,j}, x'_{1,i,j}, x'_{2,i,j}, y'_{i,j}\right) := \left(g_1^{a_{i,j}}, g_2^{a_{i,j}}, g_1^{b_{i,j}}, g_2^{b_{i,j}}, h^{a_{i,j}} \cdot h'^{b_{i,j}} \sigma_{i,j}\right)$$

und den nicht-interaktiven Designated-Verifier-Beweis, dass

$$\left(x_{1,i,j} \cdot x_{2,i,j} \cdot x'_{1,i,j} \cdot x'_{2,i,j}, y'_{i,j}\right) = \left(\left(g_1 \cdot g_2\right)^{\xi} \left(x_{1,i,j} \cdot x_{2,i,j}\right), h'^{\xi} y_{i,j}\right)$$

für ein $\xi \in \mathbb{Z}_q$ gilt (vgl. Abschnitt 3.5).
Der Beweis und $(x_{1,i,j}, x_{2,i,j}, x'_{1,i,j}, x'_{2,i,j}, y'_{i,j})$ werden geheim an den Wähler übertragen.
Der Wähler V_i verifiziert die Designated-Verifier-Beweise der Autoritäten und berechnet das verschlüsselte Credential

$$\begin{aligned}
(x_{1,i}, x_{2,i}, y_i) &:= \left(\prod_{j=1}^{n_R} x_{1,i,j}, \prod_{j=1}^{n_R} x_{2,i,j}, \prod_{j=1}^{n_R} y_{i,j}\right) \\
&= \left(\prod_{j=1}^{n_R} g_1^{a_{i,j}}, \prod_{j=1}^{n_R} g_2^{a_{i,j}}, \prod_{j=1}^{n_R} h^{a_{i,j}} \sigma_{i,j}\right) \\
&\underset{\sigma_i := \prod_{j=1}^{n_R} \sigma_{i,j}}{=} \left(g_1^{\sum_{j=1}^{n_R} a_{i,j}}, g_2^{\sum_{j=1}^{n_R} a_{i,j}}, h^{\sum_{j=1}^{n_R} a_{i,j}} \sigma_i\right) \\
&\underset{a_i := \sum_{j=1}^{n_R} a_{i,j}}{=} \left(g_1^{a_i}, g_2^{a_i}, h^{a_i} \sigma_i\right).
\end{aligned}$$

Das Credential dient als Wahlberechtigung des Wählers und wird auf dem Observer gespeichert.
Die Autoritäten berechnen öffentlich

$$\begin{aligned}
\left(x_{1,i}, x_{2,i}, x'_i, x'_{2,i}, y'_i\right) &:= \left(\prod_{j=1}^{n_R} x_{1,i,j}, \prod_{j=1}^{n_R} x_{2,i,j}, \prod_{j=1}^{n_R} x'_{1,i,j}, \prod_{j=1}^{n_R} x'_{2,i,j}, \prod_{j=1}^{n_R} y'_{i,j}\right) \\
&= \left(g_1^{a_i}, g_2^{a_i}, \prod_{j=1}^{n_R} g_1^{b_{i,j}}, \prod_{j=1}^{n_R} g_2^{b_{i,j}}, \prod_{j=1}^{n_R} h^{a_{i,j}} h'^{b_{i,j}} \sigma_{i,j}\right) \\
&= \left(g_1^{a_i}, g_2^{a_i}, g_1^{\sum_{j=1}^{n_R} b_{i,j}}, g_2^{\sum_{j=1}^{n_R} b_{i,j}}, h^{\sum_{j=1}^{n_R} a_{i,j}} h'^{\sum_{j=1}^{n_R} b_{i,j}} \sigma_i\right) \\
&\underset{b_i := \sum_{j=1}^{n_R} b_{i,j}}{=} \left(g_1^{a_i}, g_2^{a_i}, g_1^{b_i}, g_2^{b_i}, h^{a_i} h'^{b_i} \sigma_i\right).
\end{aligned}$$

Dieser Wert wird auf die Liste der wahlberechtigten Credentials gesetzt. Am Ende der Registrierungsphase werden die Werte durch das verifizierbare robuste Entschlüsselungs-MIX-Netz $\mathcal{A}$ geschickt. Die Ausgabe des MIX-Netzes $\mathcal{A}$ ist die Liste der wahlberechtigten, unter h' verschlüsselten Credentials $(g_1^{b_i}, g_2^{b_i}, h'^{b_i} \sigma_i)$, für $1 \leq i \leq n_V$.

Der Wahlberechtigte wählt geheim einen weiteren Wert $z_{2,i} \in_R \mathbb{Z}_q$ und berechnet $h_2 := g^{z_{2,i}}$. Der Wert h_2 wird auf dem Observer gespeichert.

Für den Fall einer Erpressung wählt der Wahlberechtigte weitere Zufallszahlen $a'_i, \sigma'_i \in \mathbb{Z}_q$ und berechnet $(g_1^{a'_i}, g_2^{a'_i}, h^{a'_i} \sigma'_i)$. Dieser Geheimtext, der als falsches Credential dient, wird vom Wähler auf dem Observer gespeichert bzw. im Fall eines Angriffes dem Angreifer ausgehändigt.

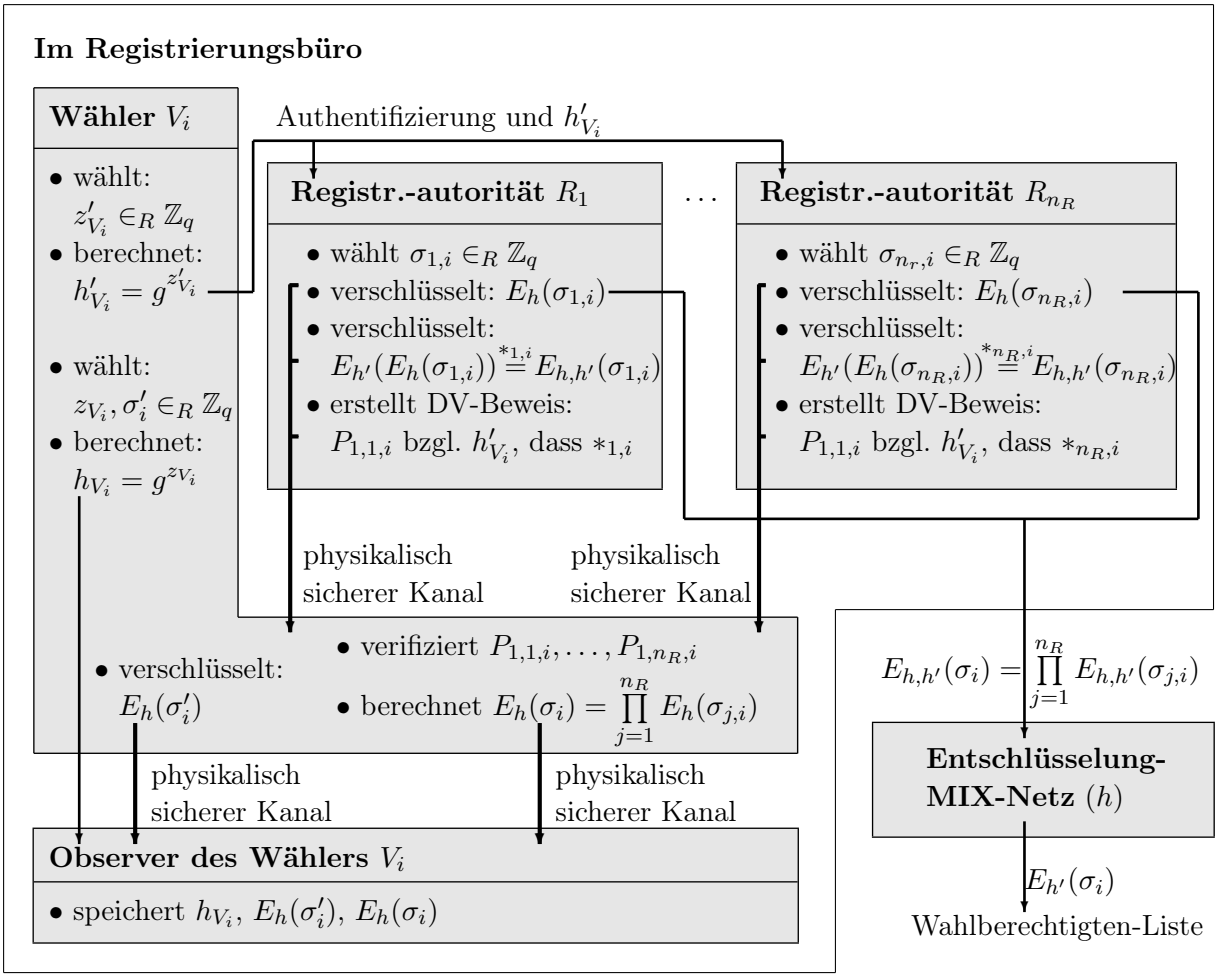

Abbildung 6.1: Registrierungsphase im erpressungsresistenten Wahlsystem mit Observer.

6.2.3 Wahlphase

Der Wähler wählt seine Wahloption $m_i \in \mathbf{L}$ und zwei Zufallszahlen $\alpha_i, \alpha'_i \in \mathbb{Z}_q$, berechnet

$$(x_{1,m_i}, x_{2,m_i}, y_{m_i}) = (g_1^{\alpha_i}, g_2^{\alpha_i}, h'^{\alpha_i} m_i), \quad g_1^{\alpha'_i}, g_2^{\alpha'_i}$$

und sendet diese Werte an den Observer. Der Observer wählt Zufallszahlen $\beta_i, \beta'_i, \beta_{i,\sigma}, \beta'_{i,\sigma} \in \mathbb{Z}_q$ und berechnet die Wiederverschlüsselung

$$\left(x'_{1,m_i}, x'_{2,m_i}, y'_{m_i}\right) := \left(g_1^{\beta_i} x_{1,m_i}, g_2^{\beta_i} x_{2,m_i}, h'^{\beta_i} y_{m_i}\right) = \left(g_1^{\alpha_i+\beta_i}, g_2^{\alpha_i+\beta_i}, h'^{\alpha_i+\beta_i} m_i\right)$$

und die kommutative Verschlüsselung (vgl. Abschnitt 2.4.1)

$$\left(x_{1,i}, x_{2,i}, g_1^{\beta_{i,\sigma}}, g_2^{\beta_{i,\sigma}}, h'^{\beta_{i,\sigma}} y_i\right).$$

Mittels der Hashfunktion $\mathcal{H}$ erstellt der Observer seinen Anteil am NIZK-Beweis der non-malleability

$$(\beta_i+\beta_{i,\sigma}) \cdot \mathcal{H}\left(g_1, g_2, x'_{1,m_i}, x'_{2,m_i}, y'_{m_i}, g_1^{\alpha'_i+\beta'_i+\beta'_{i,\sigma}}, g_2^{\alpha'_i+\beta'_i+\beta'_{i,\sigma}}, x_{1,i}, x_{2,i}, g_1^{\beta_{i,\sigma}}, g_2^{\beta_{i,\sigma}}, h'^{\beta_{i,\sigma}} y_i\right) + \beta'_i + \beta'_{i,\sigma}.$$

Er schickt diesen sowie

$$\left(x'_{1,m_i}, x'_{2,m_i}, y'_{m_i}\right),\ g_1^{\alpha'_i+\beta'_i+\beta'_{i,\sigma}},\ g_2^{\alpha'_i+\beta'_i+\beta'_{i,\sigma}} \text{ und } \left(x_{1,i}, x_{2,i}, g_1^{\beta_{i,\sigma}}, g_2^{\beta_{i,\sigma}}, h'^{\beta_{i,\sigma}} y_i\right)$$

an den Wähler. Außerdem schickt der Observer je einen nicht-interaktiven Designated-Verifier-Beweis bezüglich h_2 an den Wähler, dass

$$\left(x'_{1,m_i} \cdot x'_{2,m_i}, y'_{m_i}\right) = \left((g_1 \cdot g_2)^{\xi} \cdot (x_{1,m_i} \cdot x_{2,m_i}), h^{\xi} y_{m_i}\right)$$

für ein $\xi \in \mathbb{Z}_q$ und dass

$$\left(\left(g_1^{\beta_{i,\sigma}} \cdot g_2^{\beta_{i,\sigma}}\right) \cdot (x_{1,i} \cdot x_{2,i}), h'^{\beta_{i,\sigma}} y_i\right) = \left((g_1 \cdot g_1)^{\xi'} \cdot (x_{1,i} \cdot x_{2,i}), h'^{\xi'} y_i\right)$$

für ein $\xi' \in \mathbb{Z}_q$ gilt (vgl. Abschnitt 3.5). Der Wähler verifiziert die Designated-Verifier-Beweise und komplettiert den NIZK-Beweis der non-malleability:

$$P_{1,i} = (\alpha_i + \beta_i + \beta_{i,\sigma}) \cdot \mathcal{H}\Big(g_1, g_2, x'_{1,m_i}, x'_{2,m_i}, y'_{m_i}, g_1^{\alpha'_i+\beta'_i+\beta'_{i,\sigma}}, g_2^{\alpha'_i+\beta'_i+\beta'_{i,\sigma}}, x_{1,i}, x_{2,i}, g_1^{\beta_{i,\sigma}}, g_2^{\beta_{i,\sigma}}, h'^{\beta_{i,\sigma}} y_i\Big) + \alpha'_i + \beta'_i + \beta'_{i,\sigma}.$$

Observer und Wähler erstellen gemeinsam einen Beweis $P_{2,i}$, dass sie eine korrekte Wahloption verschlüsselt haben. Das geschieht mittels eines nicht-interaktiven Witness-Indistinguishable-Beweises mit zwei Provern, dass für $x'_{1,m_i}, x'_{2,m_i}, y'_{m_i}$ eine „Wiederverschlüsselung"[2] in der Menge der Wahloptionen existiert, da sonst die Unüberprüfbarkeit verloren ginge (vgl. Abschnitt 3.6.3). Der Wähler V_i erzeugt eine Zufallszahl $d_i \in \mathbb{Z}_q$ und verschlüsselt nun

$$M_i := \Big(\text{„an das Schwarze Brett"}, \left(x'_{1,m_i}, x'_{2,m_i}, y'_{m_i}\right), g_1^{\alpha'_i+\beta'_i+\beta'_{i,\sigma}}, g_2^{\alpha'_i+\beta'_i+\beta'_{i,\sigma}}, \left(x_{1,i}, x_{2,i}, g_1^{\beta_{i,\sigma}}, g_2^{\beta_{i,\sigma}}, h'^{\beta_{i,\sigma}} y_i\right), P_{1,i}, P_{2,i}\Big)$$

unter dem öffentlichen Schlüssel $\tilde{h}$ des MIX-Netzes $\tilde{\mathcal{A}}$. Diesen Geheimtext $(g_1^{d_i}, g_2^{d_i}, \tilde{h}^{d_i} M_i)$ sendet der Wähler über das robuste verifizierbare Entschlüsselungs-MIX-Netz $\tilde{\mathcal{A}}$. Dieses MIX-Netz verarbeitet auch andere Nachrichten mit anderen Empfängeradressen. Die Ausgabe von $\tilde{\mathcal{A}}$ an das Schwarze Brett ist eine zufällig permutierte Liste von Nachrichten M_i, die mit einem Zeitstempel t_i versehen bzw. fortlaufend nummeriert werden. Wenn der Wähler seine Stimme durch eine andere ersetzen möchte, kann er erneut eine Stimme abgeben. Er sollte nur darauf achten, dass eine vom MIX-Netz festgelegte Zeitspanne zwischen zwei MIX-Durchläufen liegt, damit die Nachrichten Zeitstempel in der Reihenfolge der Absendungen erhalten. Die Senderanonymität kann auch dadurch erreicht werden, dass es den Wählern ermöglicht wird, von beliebigen Endgeräten aus ihre Stimmen abzugeben. In diesem Fall ist die Anonymität der Stimmen nicht nur rechnerisch sicher.

[2] Dazu werden die Wahloptionen in der Form $(1, 1, m_1), \ldots, (1, 1, m_{n_L})$ dargestellt. Im Vergleich zu Abschnitt 3.6.1 ist hier die Rolle von Wiederverschlüsselung und Verschlüsselung vertauscht. Das ist möglich, da für eine Wiederverschlüsselung $(g_1^{a+b}, g_2^{a+b}, h^{a+b} m)$ von $(g_1^a, g_2^a, h^a m)$ auch umgekehrt $(g_1^a, g_2^a, h^a m)$ eine Wiederverschlüsselung von $(g_1^{a+b}, g_2^{a+b}, h^{a+b} m)$ unter dem Witness $-b$ ist.

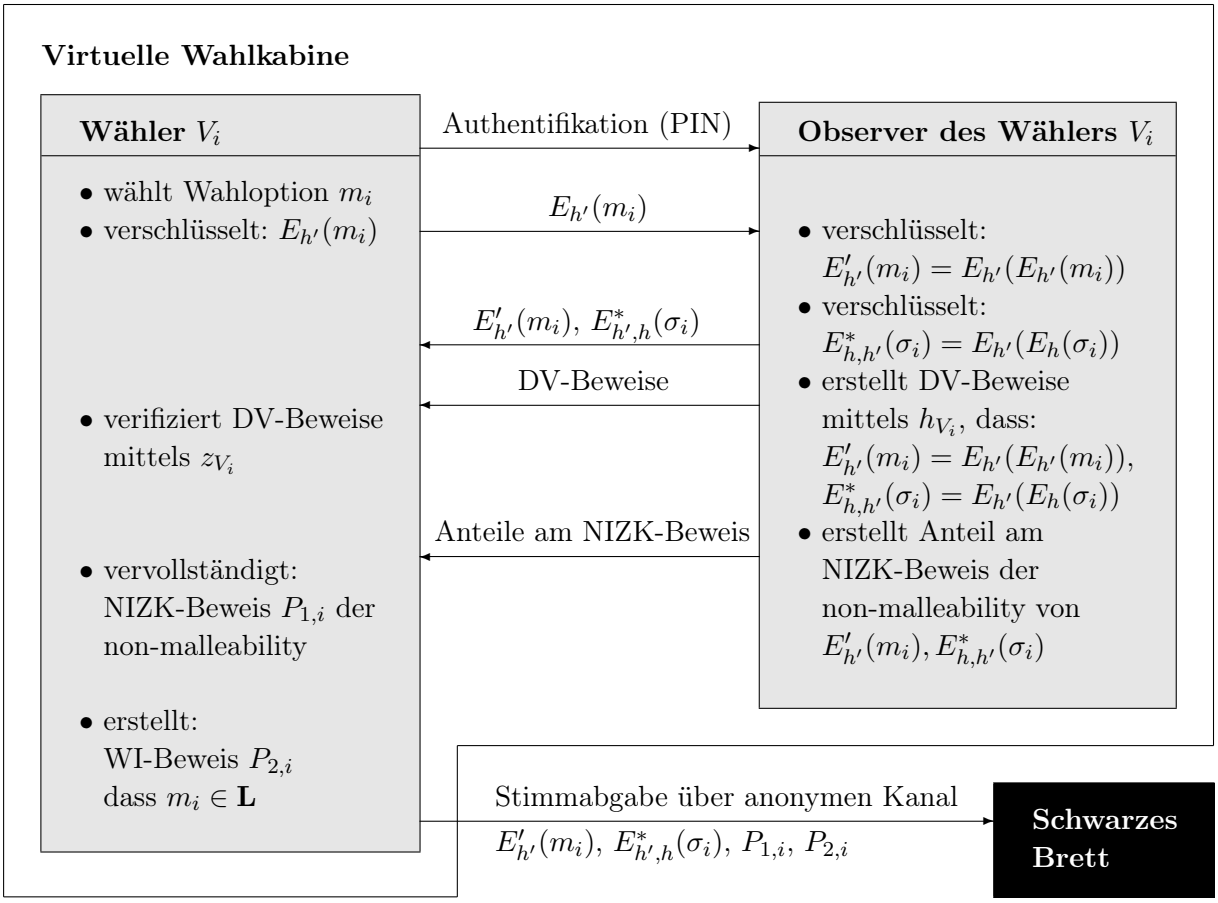

Abbildung 6.2: Wahlphase im erpressungsresistenten Wahlsystem mit Observer.

6.2.4 Auszählung

Beweise prüfen: Auf dem Schwarzen Brett kann jeder die Gültigkeit der Beweise $P_{1,i}$ und $P_{2,i}$ verifizieren.

Duplikate entfernen: Zu den Nachrichten M_i mit korrekten Beweisen $P_{1,i}$ und $P_{2,i}$ erstellen[3] die Autoritäten $\mathcal{A}$ aus dem kommutativ unter h' und h verschlüsselten Credential

$$\left(x_{1,i}, x_{2,i}, g_1^{\beta_{i,\sigma}}, g_2^{\beta_{i,\sigma}}, h'^{\beta_{i,\sigma}} y_i\right)$$

das lediglich noch unter dem Schlüssel h' verschlüsselte Credential

$$\left(g_1^{\beta_{i,\sigma}}, g_2^{\beta_{i,\sigma}}, h'^{\beta_{i,\sigma}} \sigma_i\right).$$

Die verschlüsselten Credentials werden nun paarweise von den Autoritäten $\mathcal{A}'$ verifizierbar auf Duplikate geprüft. Entsprechend einer Policy wird pro Credential nur eine Stimme gewertet, z. B. die letzte abgegebene Stimme. Dazu werden die Zeitstempel verglichen.

- *Variante mit Test auf gleiche zugrundeliegende Klartexte*:
  Aus jeweils zwei Geheimtexten $\left(g_1^{\beta_{k,\sigma}}, g_2^{\beta_{k,\sigma}}, h'^{\beta_{k,\sigma}} \sigma_k\right)$ und $\left(g_1^{\beta_{\ell,\sigma}}, g_2^{\beta_{\ell,\sigma}}, h'^{\beta_{\ell,\sigma}} \sigma_\ell\right)$ wird nun

[3] Sie entschlüsseln verteilt und verifizierbar $x_{1,i}^{-s_1} \cdot x_{2,i}^{-s_2} h'^{\beta_{i,\sigma}} y_i = g_1^{-a_i s_1} \cdot g_2^{-a_i s_2} \cdot h'^{\beta_{i,\sigma}} g_1^{a_i s_1} g_2^{a_i s_2} \sigma_i$ (vgl. Abschnitt 3.7.6).

der Wert $\left(g_1^{\beta_{k,\sigma}-\beta_{\ell,\sigma}}, g_2^{\beta_{k,\sigma}-\beta_{\ell,\sigma}}, h'^{\beta_{k,\sigma}-\beta_{\ell,\sigma}}\sigma_k\sigma_\ell^{-1}\right)$ berechnet. Die Autoritäten $\mathcal{A}'$ prüfen gemeinsam wie in Abschnitt 3.9 beschrieben, ob $\sigma_k = \sigma_\ell$ gilt, dies also eine Verschlüsselung des Klartextes 1 ist oder nicht.

- *Variante mit einer deterministischen Blendung der Credentials*:
  Die Autoritäten $\mathcal{A}'$ entschlüsseln den Geheimtext $\left(g_1^{\beta_{i,\sigma}}, g_2^{\beta_{i,\sigma}}, h'^{\beta_{i,\sigma}}\sigma_i\right)$ und blenden die zugrundeliegende Nachricht σ_i zugleich nach dem in Abschnitt 3.10 beschriebenen Verfahren. Die Ausgabe dieser verifizierbaren Berechnung ist das deterministisch geblendete Credential $\sigma_i^{\hat{s}}$. Anhand der Tabelle der Hashwerte von deterministisch geblendeten Credentials können nun effizient Nachrichten zu gleichen Credentials herausgefunden werden.

MIX-Netzwerk $\mathcal{A}'$: Die duplikatfreie Liste der unter dem Schlüssel h' verschlüsselten Credentials $\left(g_1^{\beta_{i,\sigma}}, g_2^{\beta_{i,\sigma}}, h'^{\beta_{i,\sigma}}\sigma_i\right)$ und Stimmen $\left(x'_{1,m_i}, x'_{2,m_i}, y'_{m_i}\right)$ wird vom verifizierbaren robusten MIX-Netzwerk $\mathcal{A}'$ entschlüsselt und permutiert. Paare von Credential und Stimme bleiben dabei zusammen, lediglich die Anordnung der Paare wird zufällig verändert. Die Ausgabe des MIX-Netzes ist eine Liste von Klartextstimmen m_i und zugehörigen Credentials σ_i.

Die nach der Registrierungsphase vom MIX-Netz $\mathcal{A}$ veröffentlichte Liste der wahlberechtigten unter h' verschlüsselten Credentials wird ebenfalls verifizierbar von den Autoritäten $\mathcal{A}'$ entschlüsselt.

Credentials prüfen: Schließlich werden die Klartexte eines jeden abgegeben Credentials σ_i mit den Klartexten aus der Liste der wahlberechtigten Credentials verglichen. Übrig bleibt eine Liste I von Wahloptionen, die zu wahlberechtigten Credentials gehören.

Auszählung: Die gültigen Stimmen von wahlberechtigten Wählern können dann von jedem ausgezählt werden.

6.3 Analyse des Wahlsystems

Der Schwerpunkt bei der Analyse des Wahlsystems liegt auf die für das Wahlsystem besondere und charakteristische Eigenschaft der Erpressungsresistenz.

Wahlberechtigung

Die Überprüfung der Wahlberechtigung wird durch die Credentials sichergestellt. In der Registrierungsphase werden die Credentials der Wahlberechtigten in verschlüsselter und permutierter Form auf einer Liste zertifiziert. Durch die verifizierbaren Tests auf Gleichheit der Klartexte bzw. deterministischer Blendung in der Auszählung und den Einsatz verifizierbarer MIX-Netze kann jeder überzeugt werden, welche Nachrichten von Wahlberechtigten stammen und welche nicht. Unberechtigt abgegebene Nachrichten werden nicht berücksichtigt.

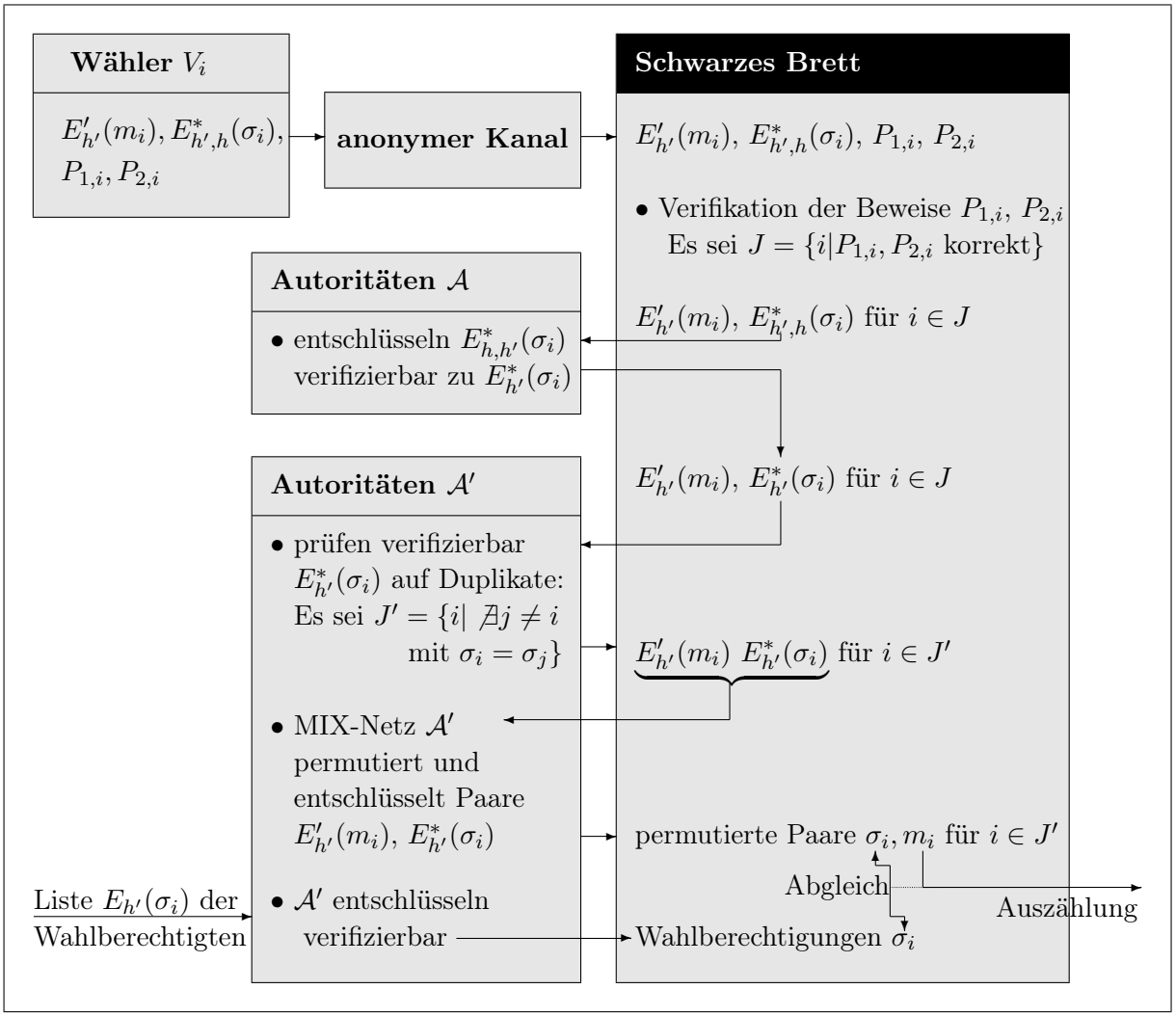

Abbildung 6.3: Auszählung im erpressungsresistenten Wahlsystem mit Observer.

Einmaligkeit

Wenn nur die zuerst veröffentlichten Stimmen mit korrektem Credential in das MIX-Netz eingehen und eventuell später veröffentlichte Stimmen für ungültig erklärt und verworfen werden, wird durch die Verifizierbarkeit des MIX-Netzes sichergestellt, dass jeder Wähler nur einmal seine Stimme abgeben kann. Die Einmaligkeit basiert auf der Fälschungssicherheit der Credentials.

Fälschungssicherheit

Die Stimme eines Wählers ist fälschungssicher, da jeder Wähler zu der von ihm gewählten Wahloption sein Credential hinzufügen muss. Die non-malleability der Verschlüsselung verbindet Credential und Wahloption somit miteinander.

Verifizierbarkeit

Da das Schwarze Brett öffentlich lesbar ist, kann jeder die Verarbeitung der eingegangenen Nachrichten verfolgen. Nach jedem in Abschnitt 6.2.4 beschriebenen Schritt der Auszählung werden

das Ergebnis der Berechnungen und die Beweise der korrekten Berechnung auf dem Schwarzen Brett veröffentlicht. Um ein falsches Auszählungsergebnis zu provozieren, müsste es dem Angreifer bei mindestens einem dieser Schritte gelingen von der korrekten Durchführung abzuweichen - ohne dass dies von jedem überprüft werden kann. Selbst wenn der Angreifer die Credentials aller Wähler kennt und alle Nachrichten, die auf dem Schwarzen Brett eingestellt werden kontrolliert oder sogar selbst produziert, kann er nur dann ein falsches Auszählungsergebnis herbeiführen, wenn er mindestens einen der Beweise der Korrektheit der einzelnen Auszählungsschritte fälschen kann. Unter der Annahme der Sicherheit der Witness-Indistinguishable- und der Zero-Knowledge-Beweise der korrekten Berechnung bzw. der Annahme, dass nur eine Minderheit der Auszählungsautoritäten korrupt ist, ist somit für einen polynomiell zeitbeschränkten Angreifer die Erfolgswahrscheinlichkeit $\mathbf{Succ}^{Ver.}_{\mathsf{WS},X}$ vernachlässigbar. Somit folgt die globale Verifizierbarkeit des Wahlsystems im Sinne von Definition 4.2.

Korrektheit

Die Korrektheit der Auszählung ist garantiert, wenn alle Wähler die Stimme ihrer Wahl abgeben können, und es dem Angreifer nicht gelingt, Stimmen hinzuzufügen oder gültige Stimmen zu löschen. Die Definition der Korrektheit des Wahlsystems setzt die Verifizierbarkeit des Wahlsystems, also die Korrektheit der Auszählung voraus. Das bedeutet in diesem Wahlsystem, dass es höchstens vor Eintreffen der Nachrichten auf dem Schwarzen Brett Angriffsmöglichkeiten geben kann. Um eine Stimme für einen ehrlichen Wähler abzugeben, die in die Auszählung einfließt, benötigt ein Angreifer ein gültiges Credential eines ehrlichen Wählers. Dazu bieten sich die folgenden Möglichkeiten, die aber alle aufgrund der Protokolleigenschaften der verwendeten kryptografischen Bausteine verhindert werden.

- Der Angreifer X könnte ein Credential von der veröffentlichten Liste der wahlberechtigten verschlüsselten Credentials nehmen. Da dieses Credential unter dem öffentlichen Schlüssel h' bereits verschlüsselt ist, kann der Angreifer den Beweis der non-malleability nicht erbringen, da er die zur Verschlüsselung unter h' verwendete Zufallszahl nicht kennt.

- Er könnte ein von einem Wähler abgegebenes unter h' verschlüsseltes Credential als sein eigenes ausgeben. Aber auch in diesem Fall kann er den notwendigen Beweis der non-malleability nicht führen.

- X könnte Stimmen eines Wählers abfangen oder manipulieren. Da der Wähler aber seine Stimme kennt, kann er die am Schwarzen Brett eingegangenen Nachrichten überprüfen und seine Stimme nochmals, gegebenenfalls über ein anderes Endgerät abgeben.

- Der Angreifer könnte versuchen mit Hilfe korrupter Registrierungsautoritäten, einem Wähler ein verschlüsseltes Credential zu geben, dass nicht einem unter h' verschlüsselten Credential in der Liste der Wahlberechtigungen entspricht. Dies wird aber durch die zu führenden Designated-Verifier-Beweise bei der Registrierung verhindert.

- Eine weitere Angriffsmöglichkeit, die aber bei genauerer Betrachtung entfällt, stellt ein Versuch des Angreifers dar, einen Observer vor der Ausgabe an den Wähler so zu manipulieren, dass dieser nicht korrekt arbeitet. Da der Observer aber die Korrektheit der Verschlüsselung in Designated-Verifier-Beweisen dem Wähler beweisen muss, ist es durch den Wähler erkennbar. Da der Observer zur Stimmabgabe nicht benötigt wird, kann der Wähler auch ohne Observer seine Stimme abgeben.

Ehrlichkeit, Robustheit

Ein Wahlsystem erfüllt die Anforderung der Ehrlichkeit, wenn ein unehrlicher Wähler keine ungültige Stimme abgeben kann. Dies ist durch die Sicherheit des Witness-Indistinguishable-Beweises der korrekten Wiederverschlüsselung einer der gültigen Wahloptionen gewährleistet. Eine Manipulation von Stimmen oder ein Löschen von Stimmen auf dem Weg zum Schwarzen Brett kann vom Wähler erkannt werden. Der Wähler hat die Möglichkeit seine Stimme nochmals abzugeben.

Denial-of-Service Angriffe auf den anonymen Kanal zum Schwarzen Brett und auf das Schwarze Brett selbst können kryptografisch nicht verhindert werden und erfordern technische oder organisatorische Auswege. Daher wird hier vorausgesetzt, dass entsprechende Alternativen existieren.

Die Verifizierbarkeit der Aktionen der MIX-Server und der anderen an der Registrierung und Auszählung beteiligten Instanzen ermöglicht es, betrügerische Autoritäten zu identifizieren und auszuschließen. Solange höchstens $n_A - t$ Personen bzw. $n'_A - t$ bzw. $n_{\tilde{A}} - t$ Autoritäten unehrlich sind, kann die Wahl mit Ausschluss dieser Autoritäten durchgeführt werden. Daher ist das Wahlverfahren robust.

Wahlaufwand

Der Aufwand auf Seiten des Wählers und auf Seiten der Autoritäten ist groß, die Auszählung erfolgt aber deutlich effizienter als im einzigen weiteren erpressungsresistente Wahlsystem [JCJ05]. Es verzichtet auf einen der beiden aufwändigen paarweisen Tests auf Gleichheit der den abgegebenen verschlüsselten Credentials zugrundeliegenden Klartexte. Mit Hilfe des in Abschnitt 3.10 vorgestellten Verfahrens kann sogar auf den zweiten verzichtet werden. Dazu entschlüsselt das MIX-Netz $\mathcal{A}'$ bei der Auszählung zwar die Stimmen, erstellt aber synchron dazu keine Entschlüsselung der Credentials, sondern eine Wiederverschlüsselung derselben (vgl. Abbildung 6.4). So kann man anschließend mit einem von $\hat{s}$ verschiedenen Geheimnis $\hat{\hat{s}}$ deterministisch geblendete Credentials erhalten. Wendet man nun das gleiche Verfahren auf die Liste der wahlberechtigten Credentials an, erhält man eine Liste deterministisch mittels $\hat{\hat{s}}$ geblendeter wahlberechtigter Credentials. Da die Credentials auch am Ende der Auszählung nicht im Klartext vorliegen, können sie bei einer weiteren Wahl erneut als Wahlberechtigung fungieren[4].

[4] Allerdings sollte man bei der Auszählung in den weiteren Wahlen jeweils die Blendungswerte $\hat{s}$ und $\hat{\hat{s}}$ ändern, um Korrelationen zu Auszählungen vorheriger Wahlen zu verschleiern.

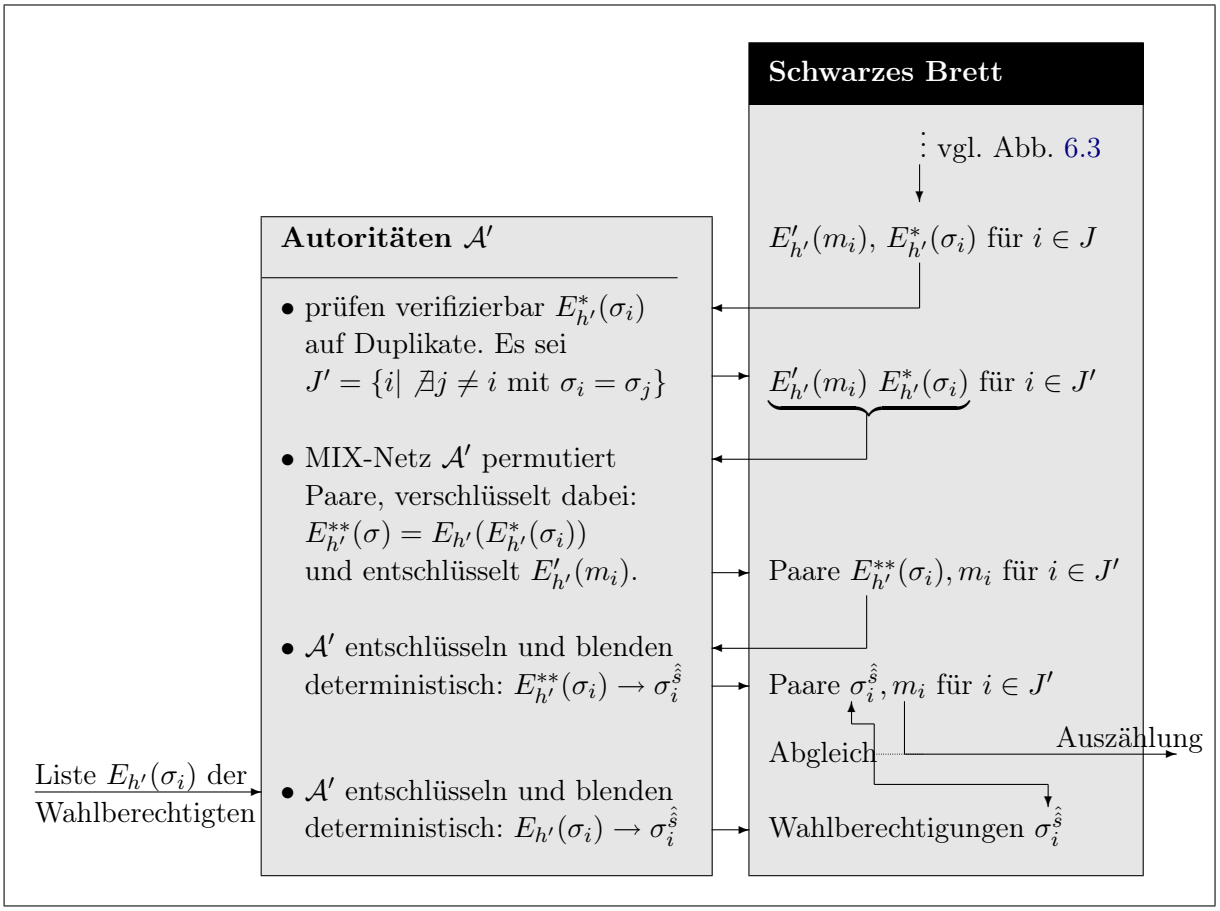

Abbildung 6.4: Auszählung mit deterministischer Blendung der Wahlberechtigungen.

Wahlgeheimnis, Anonymität

Die Anonymität eines jeden Wählers ist garantiert, wenn die verschlüsselte Stimme nicht durch eine außenstehende Person oder eine Gruppe von weniger als t Personen des Wahlausschusses entschlüsselt werden kann. Das trifft für dieses Wahlsystem zu. Das Wahlsystem ist sogar unüberprüfbar, d. h. selbst bei einer Kooperation von Wähler und Angreifer ist es dem Wähler nicht möglich dem Angreifer nachzuweisen, wie er gewählt hat. Es ist darüber hinaus sicher gegen Enthaltungsangriffe (siehe hierzu die Analyse der Erpressungsresistenz), d. h. ein Angreifer weiß nicht einmal, *ob* ein Wähler eine Stimme abgegeben hat.

Unabhängigkeit

Es ist nicht möglich, eine Stimme eines anderen Wählers zu kopieren, da man für die verschlüsselte Wahloption einen Zero-Knowledge-Beweis liefern muss, die zur Verschlüsselung verwendete Zufallszahl zu kennen. Bei identischen Nachrichten, die auf dem Schwarzen Brett eingehen, wird zur Vermeidung von Replay-Attacken nur die zuerst eingegangene gewertet. Durch diese Eigenschaft der non-malleability ist es nicht möglich, Stimmen abzugeben, die in einer dem Angreifer bekannten Korrelation zu Stimmen anderer Wähler stehen.

Unüberprüfbarkeit

Das Wahlsystem ist unüberprüfbar, d. h. der Wähler kann keinen Beleg erstellen, wie er gewählt hat. Die Unüberprüfbarkeit in diesem System basiert auf dem Einsatz der Credentials. Aufgrund der Erpressungsresistenz (s.u.) ist es einem Angreifer noch nicht einmal möglich, festzustellen, ob ein Wähler eine Stimme abgegeben hat.

6.3.1 Erpressungsresistenz

Der Nachweis der Erpressungsresistenz in diesem elektronischen Wahlsystem ergibt sich unter der Diffie-Hellman-Entscheidungsannahme (vgl. Abschnitt 2.2.4). Dazu wird ein polynomiell zeitbeschränkter Simulator $\mathcal{S}$ konstruiert, der das Wahlsystem im Experiment $\mathbf{Exp}^{E.-res}_{\mathsf{WS},X}$ simuliert. Ist diese Simulation für einen Angreifer X ununterscheidbar von den tatsächlichen, oben beschriebenen Komponenten des Wahlsystems und der Angreifer kann die Simulation nicht von der korrekten Durchführung abhalten, so ist das Auszählungsergebnis die einzige Information, die der Angreifer erhält. Der Angreifer X ist dann höchstens so mächtig wie der Angreifer X' im Experiment $\mathbf{Exp}^{p.-E.-res.}_{\mathsf{WS},X'}$, d. h. das Wahlsystem ist erpressungsresistent.

Der Simulator $\mathcal{S}$ simuliert das Hashfunktionsorakel $O_{\mathcal{H}}$, das als Eingabe Werte aus $\{0,1\}^*$ erhält. Die Ausgaben von $O_{\mathcal{H}}$ sind Zufallswerte einer vorgegebenen Länge (Sicherheitsparameter), d. h. gleiche Eingaben bedingen immer gleiche Ausgabewerte. $O_{\mathcal{H}}$ repräsentiert also eine ideale kryptografische Hashfunktion. Die Fähigkeit des Simulators die Hashfunktion zu simulieren entspricht der Simulation der Zufallswerte bei nicht-interaktiven Zero-Knowledge-Beweisen (siehe hierzu [BR93]).

Simulation

Zu Beginn der Simulation wird ein Bit $d \in \{0,1\}$ gewählt und der Simulator $\mathcal{S}$ erhält ein Quadrupel $(g_1, g_2, h_1, h_2,)$, das für $d = 1$ ein Diffie-Hellman-Quadrupel mit $h_1 = g_1^{s_d}$, $h_2 = g_2^{s_d}$ für ein $s_d \in \mathbb{Z}_q$ ist oder für $d = 0$ aus $g_1, g_2, g_1^{s_d}, g_2^c$ für $s_d, c \in \mathbb{Z}_q$ besteht. Das Ziel des Simulators ist es, herauszufinden, ob $d = 0$ oder $d = 1$ gilt. Der Simulator verfügt dabei während der ganzen Simulation über das Hashfunktionsorakel $O_{\mathcal{H}}$.

Wahlvorbereitung: Der Simulator wählt Zufallszahlen $s_1, s_2, s_1', s_2' \in_R \mathbb{Z}_q$, berechnet die Werte $h = g_1^{s_1} g_2^{s_2}$, $h' = g_1^{s_1'} g_2^{s_2'}$ und veröffentlicht g_1, g_2 und h, h'. Des Weiteren erzeugt der Simulator eine zufällige Liste der Wahloptionen $\mathbf{L} = (m_1, \ldots, m_{n_L})$. Aus beweistechnischen Gründen müssen die Wahloptionen hier Zufallszahlen sein.

Registrierung: Der Simulator $\mathcal{S}$ simuliert die Registrierungsautoritäten $\mathcal{R}$. Er erzeugt für $1 \le i \le n_V$ und $1 \le j \le n_R$ die Zufallszahlen $\sigma_{i,j}, a_{i,j}, b_{i,j} \in_R \mathbb{Z}_q$, die Verschlüsselungen

$$(x_{1,i,j}, x_{2,i,j}, y_{i,j}) := \left(h_1^{a_{i,j}}, h_2^{a_{i,j}}, h_1^{s_1 a_{i,j}} h_2^{s_2 a_{i,j}} \sigma_{i,j}\right)$$

und die kommutativen Verschlüsselungen

$$\left(x_{1,i,j}, x_{2,i,j}, x'_{1,i,j}, x'_{2,i,j}, y'_{i,j}\right) := \left(h_1^{a_{i,j}}, h_2^{a_{i,j}}, h_1^{b_{i,j}}, h_2^{b_{i,j}}, h_1^{s_1 a_{i,j}} h_2^{s_2 a_{i,j}} \cdot h_1^{s_1' b_{i,j}} h_2^{s_2' b_{i,j}} \sigma_{i,j}\right).$$

Er generiert außerdem die Zufallszahlen $z_{1,i} \in_R \mathbb{Z}_q$, berechnet $h_{1,i} = g^{z_{1,i}}$ und simuliert die nicht-interaktiven Designated-Verifier-Beweise, dass

$$\left(x_{1,i,j} \cdot x_{2,i,j} \cdot x'_{1,i,j} \cdot x'_{2,i,j}, y'_{i,j}\right) = \left(\left(g_1 \cdot g_2\right)^{\xi} \left(x_{1,i,j} \cdot x_{2,i,j}\right), h'^{\xi} y_{i,j}\right)$$

für ein $\xi \in \mathbb{Z}_q$ gilt.
Da die zugehörigen Challenges Werte des Hashfunktionsorakels $O_{\mathcal{H}}$ sind, kann der Simulator sich diese vorher bestimmen lassen. Außerdem kennt er die Werte $z_{1,i}$, so dass er die Beweise simulieren kann. Der Simulator berechnet nun die geheimen Credentials der Wähler

$$\left(x_{1,i}, x_{2,i}, y_i\right) := \left(\prod_{j=1}^{n_R} x_{1,i,j}, \prod_{j=1}^{n_R} x_{2,i,j}, \prod_{j=1}^{n_R} y_{i,j}\right)$$

und die zugehörige öffentliche Liste

$$\left(x_{1,i}, x_{2,i}, x'_i, x'_{2,i}, y'_i\right) := \left(\prod_{j=1}^{n_R} x_{1,i,j}, \prod_{j=1}^{n_R} x_{2,i,j}, \prod_{j=1}^{n_R} x'_{1,i,j}, \prod_{j=1}^{n_R} x'_{2,i,j}, \prod_{j=1}^{n_R} y'_{i,j}\right).$$

Nun simuliert $\mathcal{S}$ das Entschlüsselungs-MIX-Netz $\mathcal{A}$ durch Kenntnis der geheimen Schlüssel s_1, s_2 und veröffentlicht für $1 \leq i \leq n_V$ die Liste der wahlberechtigten, verschlüsselten Credentials $\left(h_1^{b_i}, h_2^{b_i}, h_1^{s'_1 b_i} h_2^{s'_2 b_i} \sigma_i\right)$. Mittels $O_{\mathcal{H}}$ erstellt er die zugehörigen Beweise der korrekten Entschlüsselung.

Korruption durch den Angreifer: Der Angreifer wählt eine Menge U von n_U korrupten Wählern aus und legt sich auf den zu erpressenden Wähler V_j sowie den für V_j gewünschten Kandidaten m fest. Sollte eine dieser Auswahlen nicht zulässig sein, d. h. falls $|U| \neq n_U$ oder $V_j \notin V \setminus U$ oder $m \notin \mathbf{L}$ gilt, so wird die Simulation abgebrochen.

Münzwurf: Es wird ein Bit $b \in_R \{0, 1\}$ gewählt.

Credentials übergeben: Der Simulator gibt dem Angreifer die Credentials $(x_{1,i}, x_{2,i}, y_i)$ für die korrupten Wähler $V_i \in U$. Der Angreifer erhält nicht nur die vollständigen Credentials, sondern zudem alle Werte, die der Wähler $V_i \in U$ während der Registrierungsphase erhalten hätte. Dazu gehören

$$\left(x_{1,i,j}, x_{2,i,j}, y_{i,j}\right) := \left(h_1^{a_{i,j}}, h_2^{a_{i,j}}, h_1^{s_1 a_{i,j}} h_2^{s_2 a_{i,j}} \sigma_{i,j}\right),$$

$$\left(x_{1,i,j}, x_{2,i,j}, x'_{1,i,j}, x'_{2,i,j}, y'_{i,j}\right) := \left(h_1^{a_{i,j}}, h_2^{a_{i,j}}, h_1^{b_{i,j}}, h_2^{b_{i,j}}, h_1^{s_1 a_{i,j}} h_2^{s_2 a_{i,j}} \cdot h_1^{s'_1 b_{i,j}} h_2^{s'_2 b_{i,j}} \sigma_{i,j}\right)$$

und die Designated-Verifier-Beweise.
Für $b = 1$ erhält der Angreifer das Credential des erpressten Wählers V_j. Ist $b = 0$, so erhält der Angreifer stattdessen zufällige Werte.

Simulation ehrlicher Wähler: Der Simulator erstellt für jeden ehrlichen Wähler $V_i \in V \setminus U$ eine Stimme, bestehend aus den beiden Geheimtexten

$$\left(x'_{1,m_i}, x'_{2,m_i}, y'_{m_i}\right) := \left(h_1^{\beta_i} h_1^{\alpha_i}, h_2^{\beta_i} h_2^{\alpha_i}, h_1^{s'_1\beta_i} h_2^{s'_2\beta_i} h_1^{s'_1\alpha_i} h_2^{s'_2\alpha_i} m_i\right)$$

und

$$\left(x_{1,i}, x_{2,i}, h_1^{\beta_{i,\sigma}}, h_2^{\beta_{i,\sigma}}, h_1^{s'_1\beta_{i,\sigma}} h_2^{s'_2\beta_{i,\sigma}} y_i\right)$$

für $\alpha_i, \alpha'_i \in \mathbb{Z}_q$. Außerdem simuliert $\mathcal{S}$ jeweils mittels des Hashfunktionsorakels $O_{\mathcal{H}}$ die zugehörigen NIZK-Beweise $P_{1,i}$ der non-malleability. Ebenfalls mit Hilfe des Hashfunktionsorakels $O_{\mathcal{H}}$ kann der Simulator die WI-Beweise $P_{2,i}$ der korrekten Wahloption simulieren. Mit Hilfe der Kenntnis von $\tilde{h}, \tilde{s}$ kann der Simulator dann unter $\tilde{h}$ verschlüsseln und das Entschlüsselungs-MIX-Netz simulieren. Mittels $O_{\mathcal{H}}$ erstellt er die zugehörigen Beweise der korrekten Entschlüsselung. Die Stimmen, bestehend aus

$$\left(x'_{1,m_i}, x'_{2,m_i}, y'_{m_i}\right), h_1^{\alpha'_i+\beta'_i+\beta'_{i,\sigma}}, h_2^{\alpha'_i+\beta'_i+\beta'_{i,\sigma}}, \left(x_{1,i}, x_{2,i}, h_1^{\beta_{i,\sigma}}, h_2^{\beta_{i,\sigma}}, h_1^{s'_1\beta_{i,\sigma}} h_2^{s'_2\beta_{i,\sigma}} y_i\right), P_{1,i}, P_{2,i},$$

werden auf das Schwarze Brett geschrieben, wo sie mit einem Zeitstempel versehen werden.

Stimmabgabe durch den Angreifer: Der Angreifer gibt eine Menge von Stimmen und den zugehörigen WI- und NIZK-Beweisen ab.

Entschlüsselung durch Simulator: Der Simulator berechnet zu den Stimmen und Credentials mit gültigen NIZK- und WI-Beweisen die Klartexte. Dazu nutzt er die Kenntnis seiner geheimen Schlüssel s_1, s_2, s'_1, s'_2.

Auszählung: Der Simulator $\mathcal{S}$ simuliert das Verhalten der ehrlichen Mehrheit der Auszählungsautoritäten. Die Minderheit, die unter Kontrolle des Angreifers stehen könnte, wird ignoriert. Die Auszählung läuft in den folgenden Schritten ab.

- **Überprüfung der Stimmen:** Der Simulator überprüft die abgegebenen Beweise. Stimmen mit ungültigen Beweisen werden gekennzeichnet und in der Folge ignoriert.
- **Teilentschlüsselung der Credentials:** Nun simuliert der Simulator mit seinen geheimen Schlüsseln s_1, s_2 die verteilte Entschlüsselung der Credentials

  $$\left(x_{1,i}, x_{2,i}, h_1^{\beta_{i,\sigma}}, h_2^{\beta_{i,\sigma}}, h_1^{s'_1\beta_{i,\sigma}} h_2^{s'_2\beta_{i,\sigma}} y_i\right)$$

  zu nur noch unter $h_1^{s'_1} h_2^{s'_2}$ verschlüsselten Credentials

  $$\left(h_1^{\beta_{i,\sigma}}, h_2^{\beta_{i,\sigma}}, h_1^{s'_1\beta_{i,\sigma}} h_2^{s'_2\beta_{i,\sigma}} \sigma_i\right).$$

  Mittels $O_{\mathcal{H}}$ erstellt er die zugehörigen Beweise der korrekten Entschlüsselung.
- **Duplikate entfernen:** Der Simulator entfernt die Duplikate. Dazu simuliert $\mathcal{S}$ die Durchführung der paarweisen Tests auf zugrundeliegende gleiche Credentials bzw. die deterministische Blendung der Credentials mit Hilfe der berechneten Klartexte. Die entsprechenden Beweise der korrekten Durchführung kann er mittels $O_{\mathcal{H}}$ simulieren. Bei den Stimmen und Credentials, die mehrfach auftreten, werden alle außer dem zuletzt abgegebenen Credential ignoriert.

- **MIX-Netz:** Nun simuliert der Simulator $\mathcal{S}$ das Entschlüsselungs-MIX-Netz, indem er die verbliebenen Stimmen und Credentials zu den Werten m_i und σ_i entschlüsselt und dabei permutiert. Darüber hinaus entschlüsselt er die Liste der wahlberechtigten, verschlüsselten Credentials $\left(h_1^{b_i}, h_2^{b_i}, h_1^{s'_1 b_i} h_2^{s'_2 b_i} \sigma_i\right)$ zu σ_i (für $1 \leq i \leq n_V$). Mittels $O_{\mathcal{H}}$ erstellt er die zugehörigen Beweise der korrekten Entschlüsselung.
- **Stimmauszählung:** Die Klartextstimmen mit zugehörigen wahlberechtigten Klartextcredentials werden ausgezählt.

Der Angreifer muss nun ermitteln, ob sein Angriff Erfolg hatte. Er gibt ein Bit b' aus, das dann 1 ist, wenn er Grund zu der Vermutung hat, dass der Erpresste auf seine Forderung eingegangen ist. Der Simulator gibt b' als Vermutung für die Diffie-Hellman-Entscheidungs-Challenge aus.

Falls der Simulator zu Beginn ein Diffie-Hellman-Quadrupel erhalten hat (Fall $d = 1$), dann ist mit

$$g_1 = g, \;\; g_2 = g^a, \;\; h_1 = g^b, \;\; h_2 = g^{ab}$$

das Tripel

$$(h_1^r, h_2^r, h_1^{rs_1} h_2^{rs_2} m) = \left(g^{br}, g^{abr}, g^{brs_1} g^{abrs_2} m\right) = \left(g_1 br, g_2^{br}, h^{br} m\right)$$

eine Verschlüsselung im modifizierten ElGamal-Verschlüsselungsschema unter der Zufallszahl $b \cdot r$. Entsprechendes gilt für die Verschlüsselungen unter den anderen modifizierten ElGamal-Schlüsselpaaren. Somit ist die obige Simulation ununterscheidbar vom Experiment $\mathbf{Exp}_{\mathsf{WS},X}^{E.-res}$ und es gilt:

$$Pr\left(b' = 1 | d = 1\right) = Pr\left(\mathbf{Exp}_{\mathsf{WS},X}^{E.-res} = 1\right) = \mathbf{Succ}_{\mathsf{WS},X}^{E.-res.}.$$

Ist andererseits die Eingabe des Simulators kein Diffie-Hellman-Quadrupel (Fall $d = 0$), dann ist mit

$$g_1 = g, \;\; g_2 = g^a, \;\; h_1 = g^b, \;\; h_2 = g^c \;\text{ für ein } c \in \mathbb{Z}_q$$

ein Tripel der Form

$$\begin{aligned}
(h_1^r, h_2^r, h_1^{rs_1} h_2^{rs_2} m) &= \left(g^{br}, g^c r, g^{brs_1} g^{crs_2} m\right) \\
&\underset{c' := ca^{-1}}{=} \left(g_1^{br}, g_1^{c'r}, g_1^{brs_1} g_2^{c'rs_2} m\right) \\
&\underset{c'' := c' - b}{=} \left(g_1^{br}, g_1^{c'r}, g_1^{brs_1} g_2^{brs_2} g_2^{c''rs_2} m\right) \\
&= \left(g_1^{br}, g_1^{c'r}, h^{br}\left(g_2^{c''rs_2} m\right)\right)
\end{aligned}$$

eine modifizierte ElGamal-Verschlüsselung unter der Zufallszahl br, allerdings von $g_2^{c''rs_2} m$. Die Nachricht m ist durch $g_2^{c''rs_2}$ perfekt geblendet.

Somit ist die vom Simulator erzeugte Sicht, die der Angreifer von der Simulation hat, perfekt ununterscheidbar von der im Experiment $\mathbf{Exp}_{\mathsf{WS},X'}^{p.-E.-res.}$ und es gilt:

$$Pr\left(b' = 1 | d = 0\right) = Pr\left(\mathbf{Exp}_{\mathsf{WS},X'}^{p.-E.-res.} = 1\right) = \mathbf{Succ}_{\mathsf{WS},X'}^{p.-E.-res.}.$$

Das bedeutet, dass die Möglichkeit einer Erpressung für den Angreifer so groß ist wie die Möglichkeit für den Simulator, die Diffie-Hellman-Entscheidungsannahme zu brechen:

$$\mathbf{Adv}_S^{DDH} = |Pr\left(b' = 1 | d = 1\right) - Pr\left(b' = 1 | d = 0\right)| = \mathbf{Adv}_{\mathsf{WS},X}^{E.-res.}$$

Die Wahrscheinlichkeit einer Erpressung im Wahlsystem ist somit unter der Diffie-Hellman-Entscheidungsannahme vernachlässigbar.

The highest measure of democracy is neither the 'extent of freedom' nor the 'extent of equality', but rather the highest measure of participation.

Robert Orben

Kapitel 7

Zusammenfassende Analyse

In Abschnitt 1.2 wurde ein Überblick über den Stand der Forschung bei elektronischen Wahlsystemen gegeben und die aktuellen Probleme beim Erreichen der Unüberprüfbarkeit und Erpressungsresistenz erläutert. Um quittungsfreie und erpressungsresistente Wahlsysteme zu konstruieren, wurde in den Kapiteln 5 und 6 ein Observer eingesetzt.

Die für diese Wahlsysteme verwendeten kryptografischen Bausteine sowie der kryptografische und mathematische Hintergrund wurden in den Kapiteln 2 und 3 konstruiert und beschrieben. Dabei wurden unter anderem Witness-Indistinguishable- und Zero-Knowledge-Beweise so angepasst, dass sie für Wähler und Observer als Prover funktionieren.

In Kapitel 5 wurde zunächst ein bestehendes Wahlsystem mit Observer untersucht. Es wurde gezeigt, dass das Wahlsystem [MBC01] keine Unüberprüfbarkeit bietet und die Verschlüsselung malleable ist. Anschließend wurden diese Lücken geschlossen. Das daraus resultierende verbesserte Wahlsystem mit Observer, das ich in [Sch05a] veröffentlicht habe, wurde in Abschnitt 5.2 vorgestellt. Im Gegensatz zu [MBC01] ist die Unabhängigkeit der Stimmabgabe gewährleistet. Darüber hinaus ist es das erste Wahlsystem, das Unüberprüfbarkeit auch ohne die unrealistische Annahme eines physikalisch sicheren Kanals von jeder Autorität im Wahlausschuss zu jedem Wähler erreicht hat.

In Abschnitt 5.3 wurde ein weiteres Wahlsystem mit Observer vorgestellt, das ebenfalls Unüberprüfbarkeit bietet, aber deutlich effizienter ist. Insbesondere wurde der Aufwand auf der Seite des Wählers deutlich reduziert. Dieses Wahlsystem habe ich in [Sch06a] veröffentlicht.

Mit den beiden Wahlsystemen wurde gezeigt, dass es möglich ist, mit Hilfe eines Observers effiziente elektronische Wahlsysteme aufzustellen, die beinahe allen Anforderungen elektronischer Wahlen genügen. Lediglich ein Enthaltungsangriff ist möglich, da ein Angreifer feststellen kann, welcher Wähler eine Stimme abgegeben hat, sofern er die Signatur dem Wähler zuordnen kann.

Schließlich wurde in Kapitel 6 ein erpressungsresistentes Wahlsystem mit Observer vorgestellt. Das Wahlsystem ist deutlich aufwändiger als das in Abschnitt 5.3. Es ist aber sicher gegen den Enthaltungsangriff. Um dieses Sicherheitsziel zu erreichen, kann ein Observer nicht mehr im ursprünglichen Sinn von Chaum und Pedersen bzw. Cramer und Pedersen (vgl. [CP92], [CP93]) obligatorisch an den Berechnungen zur Stimmabgabe beteiligt werden, da ein Wähler sonst nach der Aushändigung des Observers an einen Erpresser nicht mehr abstimmen könnte.

Der Observer dient in diesem System im Wesentlichen als sicheres Transportmedium des in der Registrierungsphase generierten Credentials. Deren Erzeugung wurde in [JCJ05] nicht beschrieben. Diese Lücke wurde in Abschnitt 6.1.3 geschlossen. Das vorgestellte Wahlsystem ist deutlich effizienter als das von Juels et al. vorgeschlagene System, da es auf einen der beiden aufwändigen Tests auf Gleichheit der zugrundeliegenden Klartexte verzichtet. In Abschnitt 6.2.4 wurde zudem gezeigt, wie man mit einer verteilten Entschlüsselung und gleichzeitigen Blendung der Credentials durch einen festen Wert auch auf den zweiten dieser Tests verzichten kann. Das Wahlsystem habe ich in [Sch06b] veröffentlicht.

Welches der vorgestellten Wahlsysteme zum Einsatz kommen sollte, hängt von den individuellen Anforderungen ab, die an die verschiedenen Typen von Wahlen (politische Wahlen, Vereinswahlen, Betriebsratswahlen etc.) zu stellen sind. Bislang ist noch nicht gesetzlich festgelegt, welche konkreten Anforderungen an Wahlsysteme für die unterschiedlichen Wahlen in Deutschland zu stellen sind. Zurzeit erstellt das Bundesamt für Sicherheit in der Informationstechnik ein Schutzprofil, das die grundlegenden Anforderungen an die Wahlphase und Auszählung bei elektronischen Wahlen spezifiziert. In diesen Anforderungen wird nichts darüber ausgesagt, ob es einem Angreifer unmöglich sein muss, feststellen zu können, *ob* ein Wähler eine Stimme abgegeben hat. Kann man auf die Sicherheit gegen einen Enthaltungsangriff verzichten, so hilft ein Observer die übrigen Anforderungen in einem Wahlsytem effizient zu erfüllen.

Elektronische Wahlen haben insbesondere in den letzten Jahren mit den Wahlen in Estland und in der Schweiz zunehmend an Bedeutung gewonnen. Auch in Deutschland wurden einige elektronische Wahlen durchgeführt. Beispielsweise wurde vom 9. bis 11. Mai 2006 die Betriebsratswahl in einigen Bereichen der Deutschen Telekom elektronisch durchgeführt. Die Steigerung der Wahlbeteiligung um fast 100% gegenüber der vorherigen Wahl ist ein Indiz, dass Wähler durch elektronische Wahlen motiviert werden können, aktiv an der Demokratie teilzunehmen (siehe [Die06]). Auch bei gesetzlich geregelten politischen Wahlen wurde bereits elektronisch gewählt (siehe [MSM+01]).

Da nicht jeder Wahlberechtigte die Möglichkeit hat, elektronisch über das Internet zu wählen, und zudem viele Wahlberechtigte - gerade ältere Menschen - nicht über die notwendigen Kenntnisse verfügen, bieten sich elektronische Wahlen als zusätzliche Alternative zur Briefwahl an.

Es ist nun vor allem an der Zeit, die grundlegenden Anforderungen, die das Bundesamt für Sicherheit in der Informationstechnik für elektronische Wahlen spezifiziert hat, für die verschiedenen Wahlen in Deutschland zu konkretisieren. Darüber hinaus sollte der Gesetzgeber nun unter Beachtung der technischen Möglichkeiten gesetzliche Regelungen treffen, die elektronische Wahlsysteme zu erfüllen haben.

Die Aufgabe der kryptografischen Forschung wird es in Zukunft sein, die Sicherheitsziele, die an elektronische Wahlen zu stellen sind, formaler zu definieren und dann die bestehenden Systeme mit Hilfe formaler Sicherheitsbeweise genau auf die Erfüllung ihrer Sicherheitsziele hin zu analysieren. Die Formulierung der Anforderungen von Seiten der Gesetzgebung und die formale

Untersuchung bestehender Systeme im Hinblick auf die Erfüllung dieser Anforderungen von Seiten der Kryptografie sind notwendige Schritte, damit Versuche mit unsicheren Systemen oder Wahlmaschinen nicht das Vertrauen der Wähler in elektronische Wahlen unnötig erschüttern.

Literaturverzeichnis

[Abe98] Abe, Masayuki: Universally verifiable mix-net with verification work independent of the number of mix-servers. In: [Nyb98], S. 437–447

[Ahr05] Ahrens, Katharina: *Initiative D21 wählt neuen Gesamtvorstand online.* Presseinformation Nr. 39/2005, November 2005. – `http://www.initiatived21.de/presse/presseinformationen/pages/show.prl?params=recent%3D1%26type%3D10%26all%3Dall%26keyword%3D%26laufzeit%3D&id=13102&currPage=1`

[BB06] Braun, Nadja ; Brändli, Daniel: Swiss E-Voting Pilot Projects: Evaluation, Situation Analysis and How to Proceed. In: [Kri06], S. 27–36

[Bec01] Becker, Harms: *Gremienwahlen an der Hochschule Bremerhaven.* `http://www.signatur.uni-bremen.de/online-wahl/ivote-in-bremerhaven.pdf`, Oktober 2001

[BNS05] Beutelspacher, Albrecht ; Neumann, Heike B. ; Schwarzpaul, Thomas: *Kryptografie in Theorie und Praxis - Mathematische Grundlagen für elektronisches Geld, Internetsicherheit und Mobilfunk.* 1. verbesserte Auflage. Vieweg-Verlag, 2005. – ISBN 3528031689

[BR93] Bellare, Mihir ; Rogaway, Phillip: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In: Denning, Dorothy (Hrsg.) ; Pyle, Ray (Hrsg.) ; Ganesan, Ravi (Hrsg.) ; Sandhu, Ravi (Hrsg.) ; Ashby, Victoria (Hrsg.): *ACM Conference on Computer and Communications Security*, ACM Press, 1993 (CCS). – ISBN 0897916298, S. 62–73

[BR01] Bobenheim-Roxheim, Jugendgemeinderat: *Jugendgemeinderatswahl 2001 in Bobenheim-Roxheim - Das neue Wahlverfahren: Wahl per Internet.* `http://www.bobenheim-roxheim.de/jugendrat/wahlen/neue_wahlverfahren.htm`, November 2001

[BSI06] BSI, Bundesamt für Sicherheit in der Informationstechnik: *Common Criteria Protection Profile - Mindestanforderungen für Online-Wahlen.* 2006

[BSW06] Beutelspacher, Albrecht ; Schwenk, Jörg ; Wolfenstetter, Klaus-Dieter: *Moderne Verfahren der Kryptographie - Von RSA zu Zero-Knowledge.* 6. verbesserte Auflage. Vieweg-Verlag, 2006. – ISBN 383480083X

[BT94] Benaloh, Josh C. ; Tuinstra, Dwight: Receipt-free secret-ballot elections (extended abstract). In: *STOC 26*, ACM, 1994, S. 544–553

[CDS94] Cramer, Ronald ; Damgård, Ivan ; Schoenmakers, Berry: Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In: Desmedt, Yvo (Hrsg.): *CRYPTO '94*, Springer, 1994 (LNCS 839). – ISBN 3540583335, S. 174–187

[CEG87] Chaum, David ; Evertse, Jan-Hendrik ; Graaf, Jeroen van d.: An Improved Protocol for Demonstrating Possession of Discrete Logarithms and Some Generalizations. In: Chaum, David (Hrsg.) ; Price, Wyn L. (Hrsg.): *EUROCRYPT '87*, Springer, 1987 (LNCS 304). – ISBN 354019102X, S. 127–141

[CFSY96] Cramer, Ronald ; Franklin, Matthew K. ; Schoenmakers, Berry ; Yung, Moti: Multi-Authority Secret-Ballot Elections with Linear Work. In: [Mau96], S. 72–83

[CGS97] Cramer, Ronald ; Gennaro, Rosario ; Schoenmakers, Berry: A Secure and Optimally Efficient Multi-Authority Election Scheme. In: Fumy, Walter (Hrsg.): *EUROCRYPT '97* Bd. 1233, Springer, 1997 (LNCS). – ISBN 3540629750, S. 103–118

[Cha81] Chaum, David: Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms. In: *Communications of the ACM, Annual Symposium on the Theory of Computing* 24 (1981), Nr. 2, S. 84–88

[CP92] Chaum, David ; Pedersen, Torben P.: Wallet Databases with Observers. In: Brickell, Ernest F. (Hrsg.): *CRYPTO '92*, Springer, 1992 (LNCS 740). – ISBN 3540573402, S. 89–105

[CP93] Cramer, Ronald ; Pedersen, Torben P.: Improved Privacy in Wallets with Observers (Extended Abstract). In: [Hel94], S. 329–343

[CS98] Cramer, Ronald ; Shoup, Victor: A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In: Krawczyk, Hugo (Hrsg.): *CRYPTO '98* Bd. 1462, Springer, 1998 (LNCS 1462). – ISBN 3540648925, S. 13–25

[Cyb03] Cybervote, Projektkonsortium: *Cybervote - An innovative cyber voting system for internet terminals and mobile phones. The Trials - Bremen, Germany.* `http://www.eucybervote.org/trials.html`, Januar 2003

[DH76] Diffie, Whitfield ; Hellman, Martin E.: New Directions in Cryptography. In: *IEEE Transactions on Information Theory* 22, No. 6 (1976), S. 644–654

[Die06] Diehl, Klaus: *Signifikante Erhöhung der Wahlbeteiligung durch Einsatz von eVoting am Arbeitsplatz.* Presseerklärung T-Systems Enterprise Services GmbH, ITO, AL Region, Online Focus Solutions, Projekt Onlinewahlen `klaus.diehl@t-systems.com`, Mai 2006

[DK00] Desmedt, Yvo ; Kurosawa, Kaoru: How to Break a Practical MIX and Design a New One. In: [Pre00], S. 557–572

[Elg85] Elgamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: *IEEE Transactions on Information Theory* 31 (1985), Nr. 4, S. 469–472

[Eur05] Europe, Council of: *Legal, operational and technical standards for e-voting - Recommendation Rec(2004)11 and explanatory memorandum*. Council of Europe, 2005. – ISBN 9287156352

[Fel87] Feldman, Paul: A Practical Scheme for Non-interactive Verifiable Secret Sharing. In: *FOCS*, IEEE, 1987, S. 427–437

[Fel01] Fellbach, Stadt: *Jugendgemeinderatswahl 2001 in Fellbach - online*. `http://www.fellbach.de/kommunalpolitik/jugendgemeinderat/Dokumentation_JGROnlinewahl.PDF`, September 2001

[Fil01] Filderstadt, Stadtverwaltung: *Wahlen und Sitzungen des Jugendgemeinderates*. `http://www.filderstadt.de/servlet/PB/menu/1002789/1002789.html`, Dezember 2001

[FS86] Fiat, Amos ; Shamir, Adi: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: [Odl87], S. 186–194

[FS90] Feige, Uriel ; Shamir, Adi: Witness Indistinguishable and Witness Hiding Protocols. In: *STOC 22*, ACM, 1990, S. 416–426

[GI04] GI, Gesellschaft für Informatik e.V.: *Gesellschaft für Informatik hat Onlinewahl erfolgreich erprobt*. `http://www.gi-ev.de/presse/pressemitteilungen-thematisch/pressemitteilung-vom-10-dezember-2004/`, Dezember 2004

[GK96] Goldreich, Oded ; Krawczyk, Hugo: On the Composition of Zero-Knowledge Proof Systems. In: *SIAM Journal on Computing* 25 (1996), Nr. 1, S. 169–192

[GM84] Goldwasser, Shafi ; Micali, Silvio: Probabilistic Encryption. In: *Journal of Computer and System Sciences*. 28 (1984), Nr. 2, S. 270–299

[Hel94] Helleseth, Tor (Hrsg.): *Advances in Cryptology - EUROCRYPT '93, Lofthus, Norway, May 23-27, 1993, Proceedings*. Springer, 1994 (LNCS 765). – ISBN 3540576002

[HS00] Hirt, Martin ; Sako, Kazue: Efficient Receipt-Free Voting Based on Homomorphic Encryption. In: [Pre00], S. 539–556

[Jak98] Jakobsson, Markus: A Practical Mix. In: [Nyb98], S. 448–461

[JCJ05] Juels, Ari ; Catalano, Dario ; Jakobsson, Markus: Coercion-Resistant Electronic Elections. In: *WPES '05*, ACM CCS, November 2005, S. 61–70

[JJ00] Jakobsson, Markus ; Juels, Ari: Mix and Match: Secure Function Evaluation via Ciphertexts. In: Okamoto, Tatsuaki (Hrsg.): *ASIACRYPT '00*, Springer, 2000 (LNCS 1976). – ISBN 3540414045, S. 162–177

[JSI96] Jakobsson, Markus ; Sako, Kazue ; Impagliazzo, Russell: Designated Verifier Proofs and Their Applications. In: [Mau96], S. 143–154

[Kri06] Krimmer, Robert (Hrsg.): *Electronic Voting 2006 - 2nd International Workshop, Bregenz, Austria, Aug 2-4, 2006, Proceedings.* Bd. 77. GI, August 2006 (LNI). – ISBN 9783885791805

[Mau96] Maurer, Ueli M. (Hrsg.): *Advances in Cryptology - EUROCRYPT '96, Saragossa, Spain, May 12-16, 1996, Proceeding.* Springer, 1996 (LNCS 1070). – ISBN 354061186X

[MBC01] Magkos, Emmanouil ; Burmester, Mike ; Chrissikopoulos, Vassilios: Receipt-Freeness in Large-Scale Elections without Untappable Channels. In: Schmid, Beat (Hrsg.) ; Stanoevska-Slabeva, Katarina (Hrsg.) ; Tschammer, Volker (Hrsg.): *I3E '01*, Kluwer, 2001 (IFIP Conference Proceedings 202). – ISBN 0792375297, S. 683–694

[ME01] MediaKomm, Steinbeis-Transferzentrum ; Esslingen, MediaKomm: *Internet-Briefwahl zum Jugendgemeinderat Esslingen a.N. 2001 - Beschreibung der Komponenten und des Ablaufs.* `http://www.jgrwahl.esslingen.de/40606_beschreibung_internet_briefwahl_jgrwahl_es_mamr.pdf`, Juni 2001

[MM06] Madise, Ülle ; Martens, Tarvi: E-voting in Estonia 2005. The first practice of country-wide binding Internet voting in the world. In: [Kri06], S. 15–26

[MOV97] Menezes, Alfred J. ; Oorschot, Paul C. ; Vanstone, Scott A.: *Handbook of Applied Cryptography.* CRC Press LLC, 1997. – ISBN 0849385237

[MRS86] Micali, Silvio ; Rackoff, Charles ; Sloan, Bob: The Notion of Security for Probabilistic Cryptosystems. In: [Odl87], S. 381–392

[MSM+01] Marburg, Philipps-Universität ; Software, Berninger ; Marburg, Wahlleiter ; Marburg-Biedenkopf, Kreiswahlleiter ; Hessen, Landeswahlleiter ; Datenschutzbeauftragter, Hessischer ; Hessen, KIV: *Projekt ESI - Elektronische Stimmabgabe im Internet - Erfahrungsbericht zur Internet-Testwahl - Landratswahl des Landkreises Marburg-Biedenkopf am 16. September 2001.* `http://www.wahlen.hessen.de/Internetwahl.doc`, September 2001

[Nyb98] Nyberg, Kaisa (Hrsg.): *Advances in Cryptology - EUROCRYPT '98, Espoo, Finland, May 31 - June 4, 1998, Proceeding.* Springer, 1998 (LNCS 1403). – ISBN 3540645187

[Odl87] Odlyzko, Andrew M. (Hrsg.): *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*. Bd. *263*. Springer, 1987 (Lecture Notes in Computer Science)

[Oka96] Okamoto, Tatsuaki: An electronic voting scheme. In: *IFIP World Conference on IT Tools*, 1996, S. 21–30

[Oka97] Okamoto, Tatsuaki: Receipt-Free Electronic Voting Schemes for Large Scale Elections. In: Christianson, Bruce (Hrsg.) ; Crispo, Bruno (Hrsg.) ; Lomas, T. Mark A. (Hrsg.) ; Roe, Michael (Hrsg.): *Security Protocols Workshop* Bd. 1361, Springer, 1997 (Lecture Notes in Computer Science). – ISBN 3540640401, S. 25–35

[OKST97] Ogata, Wakaha ; Kurosawa, Kaoru ; Sako, Kazue ; Takatani, Kazunori: Fault tolerant anonymous channel. In: Han, Yongfei (Hrsg.) ; Okamoto, Tatsuaki (Hrsg.) ; Qing, Sihan (Hrsg.): *ICICS*, Springer, 1997 (LNCS 1334). – ISBN 354063696X, S. 440–444

[Ped91] Pedersen, Torben P.: Non-interactive and information-theoretic secure veriable secret sharing. In: Feigenbaum, Joan (Hrsg.): *CRYPTO '91*, Springer, 1991 (LNCS 576). – ISBN 3540551883, S. 129–140

[Pet00] Peters, Frens: *Internetwahlen an der FHH*. `http://www.fh-hannover.de/imperia/md/content/zentral/forschung/21.pdf`, Dezember 2000

[PG02] Pintor, Rafael L. ; Gratschew, Maria: *Voter Turnout since 1945 - A Global Report*. 3. Buch der Voter Turnout Reihe. International Institute for Democracy and Electoral Assistance (International IDEA), 2002. – ISBN 9189098617

[PIK93] Park, Choonsik ; Itoh, Kazutomo ; Kurosawa, Kaoru: Efficient Anonymous Channel and All/Nothing Election Scheme. In: [Hel94], S. 248–259

[PP89] Pfitzmann, Birgit ; Pfitzmann, Andreas: How to Break the Direct RSA-Implementation of Mixes. In: Quisquater, Jean-Jacques (Hrsg.) ; Vandewalle, Joos (Hrsg.): *EUROCRYPT '89*, Springer, 1989 (LNCS 434). – ISBN 3540534334, S. 373–381

[Pre00] Preneel, Bart (Hrsg.): *Advances in Cryptology - EUROCRYPT '00, Bruges, Belgium, May 14-18, 2000, Proceedings*. Springer, 2000 (LNCS 1807). – ISBN 3540675175

[PS96] Pointcheval, David ; Stern, Jacques: Security Proofs for Signature Schemes. In: [Mau96], S. 387–398

[Sah99] Sahai, Amit: Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In: *FOCS*, IEEE, 1999, S. 543–553

[Sch91] Schnorr, Claus-Peter: Efficient Signature Generation by Smart Cards. In: *Journal of Cryptology* 4 (1991), Nr. 3, S. 161–174

[Sch05a] Schweisgut, Jörn: Elektronische Wahlen mit Observer. In: Weinmann, Ralf-Philipp (Hrsg.): *3. Krypto-Tag der Gesellschaft für Informatik e.V. - Fachgruppe Krypto, Darmstadt, Germany. Technical Report No. TI-1/05* Bd. TI-1, Technische Universität Darmstadt - FB Informatik, September 2005 (Technical Report No. TI-1/05), S. 14

[Sch05b] Schwenk, Jörg: *Sicherheit und Kryptographie im Internet. Von sicherer E-Mail bis zu IP-Verschlüsselung.* 2. Auflage. Vieweg-Verlag, 2005. – ISBN 3834800422

[Sch06a] Schweisgut, Jörn: Effiziente elektronische Wahlen mit Observer. In: Dittmann, Jana (Hrsg.): *Sicherheit 2006* Bd. 77, GI, Februar 2006 (LNI). – ISBN 3885791714, S. 306–316

[Sch06b] Schweisgut, Jörn: Coercion-Resistant Electronic Elections with Observer. In: [Kri06], S. 171–178

[Sha79] Shamir, Adi: How to Share a Secret. In: *Commununications of the ACM, Annual Symposium on the Theory of Computing* 22 (1979), Nr. 11, S. 612–613

[Sig97] *Gesetz zur digitalen Signatur (Signaturgesetz - SigG).* In der Fassung vom 1. Mai 2001. Bundesgesetzblatt I., S. 1870 und 1872, 13. Juni 1997. – Artikel 3 des Gesetzes zur Regelung der Rahmenbedingungen für Informations- und Kommunikationsdienste

[Sig01] *Gesetz über Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften (Signaturgesetz - SigG).* In der Fassung vom 16. Mai 2001. Bundesgesetzblatt I., S. 876 und 884, 21. Mai 2001

[SJ99] Schnorr, C.P. ; Jakobsson, M.: Security Of Discrete Log Cryptosystems in the Random Oracle and Generic Model. In: *Conference on The Mathematics of Public-Key Cryptography* The Fields Institute, 1999, S. 1–15

[SK95] Sako, Kazue ; Kilian, Joe: Receipt-Free Mix-Type Voting Scheme - A Practical Solution to the Implementation of a Voting Booth. In: Guillou, Louis C. (Hrsg.) ; Quisquater, Jean-Jacques (Hrsg.): *EUROCRYPT '95*, Springer, 1995 (LNCS 921). – ISBN 3540594094, S. 393–403

[Smi05] Smith, Warren D.: *New cryptographic election protocol with best-known theoretical properties.* Frontiers in Electronic Elections (FEE 2005) - Workshop on the 10th European Symposium On Research In Computer Security - ESORICS 2005, Milan, Italy., September 12-14, 2005

[TY98] Tsiounis, Yiannis ; Yung, Moti: On the Security of ElGamal Based Encryption. In: Imai, Hideki (Hrsg.) ; Zheng, Yuliang (Hrsg.): *PKC '98*, Springer, 1998 (LNCS 1431). – ISBN 3540646930, S. 117–134

[Wed06] Weddeling, Sonja: *Abschlussbericht - Elektronische Betriebsratswahl bei der T-Systems im September 2005.* Version: 0.1, Stand: 21.03.2006 `sonja.weddeling@t-systems.com`, März 2006

[Wik05] Wikström, Douglas: A Sender Verifiable Mix-Net and a New Proof of a Shuffle. In: Roy, Bimal K. (Hrsg.): *ASIACRYPT '05*, Springer, 2005 (LNCS 3788). – ISBN 3540306846, S. 273–292

[Win05] Winter, Cornelia: *GI-Wahlen 2005 beendet.* `http://www.gi-ev.de/gi-wahlen2005/`, Dezember 2005

Index

Bezeichnungen

Die Bezeichnungen, die in dieser Arbeit verwendet werden, werden im jeweiligen Kapitel bzw. Abschnitt definiert. Zur besseren Übersicht, werden hier die wichtigsten Bezeichnungen aufgelistet.

$\mathcal{R} = \{R_1, \ldots, R_{n_R}\}$	Menge der Registrierungsautoritäten $R_1, \ldots, R_{n_R}$
$\mathcal{A} = \{A_1, \ldots, A_{n_A}\}$	Menge der Autoritäten $A_1, \ldots, A_{n_A}$ im Wahlausschuss
$\mathcal{A}' = \{A'_1, \ldots, A'_{n_{A'}}\}$	Menge der Autoritäten $A'_1, \ldots, A'_{n_{A'}}$ im Wahlausschuss
$\tilde{\mathcal{A}} = \{\tilde{A}_1, \ldots, \tilde{A}_{n_{\tilde{A}}}\}$	Menge der Autoritäten $\tilde{A}_1, \ldots, \tilde{A}_{n_{\tilde{A}}}$ im Wahlausschuss
$\mathcal{V} = \{V_1, \ldots, V_{n_V}\}$	Menge der Wähler $V_1, \ldots, V_{n_V}$
$\mathcal{W} = \{W_1, \ldots, W_{n_W}\}$	Menge der ehrlichen Wähler $W_1, \ldots, W_{n_W}$
$\mathcal{U} = \{U_1, \ldots, U_{n_U}\}$	Menge der korrupten Wähler $U_1, \ldots, U_{n_U}$
$\mathbf{L} = (m_1, \ldots, m_{n_L})$	geordnete Liste der Wahloptionen $m_1, \ldots, m_{n_L}$
h, h', $\tilde{h}$	öffentlicher ElGamal-Schlüssel von $\mathcal{A}$, bzw. $\mathcal{A}'$, bzw. $\tilde{\mathcal{A}}$
Σ_n	symmetrische Gruppe der Ordnung n
Σ^*	Menge der endlichen Bitstrings
L	Sprache ($L \subseteq \Sigma^*$)
k	Sicherheitsparameter des Kryptosystems
N	Sicherheitsparameter (Anzahl der Protokolldurchführungen)
G	endliche Untergruppe mit Primzahlordnung von $\mathbb{Z}_p^*$
q	Gruppenordnung von G
$g, g_1, g_2, \ldots, g_K, \gamma$	Generatoren von G
$P[\cdot]$	Wahrscheinlichkeitsmaß
E	Verschlüsselung
PK, SK	öffentlicher Schlüssel, privater Schlüssel
ν, μ	vernachlässigbare Funktionen
X, X'	Angreifer
$\in_R$	gemäß einer Gleichverteilung ausgewählt
sig	Signatur
$\mathcal{H}$	Hashfunktion
ID	Identität des Senders
$\oplus$	additive Verknüpfung / exklusives Oder (XOR)
$\otimes$	multiplikative Verknüpfung