SIMON GÖLZ, MICHAEL P. HEINL, CHRISTOPH BÖSCH

TRUSTWORTHY ELECTIONS?

EINE ÜBERSICHT AKTUELLER VERFAHREN & PROBLEME VON INTERNETWAHLEN IN UNKONTROLLIERTEN UMGEBUNGEN

Institut für Verteilte Systeme
Institute of Distributed Systems

University of Ulm
uulm.de/in/vs

März 2018

intentionally left blank

ABSTRACT

Remote electronic voting or informally called „Internet-Voting“ has been subject to research for several decades and is regularly part of the public debate. A special focus is thereby on remote voting procedures in uncontrolled environments. That is, election procedures which voters can use to cast their ballot conveniently from their own computer at home. Obviously, special security requirements are put on this kind of election procedures. This paper aims to present the legal requirements of general elections demanded by the example of the German constituion and derive appropriate cryptographical, technical, and organizational properties. Based on these, a methodology to assess remote electronic voting procedures regarding specific threat models is further developed. After having presented a broad survey of productively used as well as mostly academic voting procedures, those are evaluated and comparatively presented using the formerly developed methodology. Being the prototype of elections in uncontrolled environments, the postal voting procedure currently used in Germany is also assessed in order to embed the results into the overall discussion.

KURZZUSAMMENFASSUNG

Internetwahlverfahren, informell „Internet-Voting“ genannt, sind schon seit einigen Jahrzehnten im Fokus der Forschung und auch immer wieder Teil öffentlicher Debatten. Ein besonderes Hauptaugenmerk liegt dabei auf Internetwahlverfahren zur Nutzung in unkontrollierten Umgebungen, also Wahlverfahren, die Wählerinnen und Wähler bequem zu Hause vom eigenen Computer aus benutzen können, um ihre Stimme abzugeben. Selbstverständlich werden an Verfahren dieser Art besondere Anforderungen hinsichtlich ihrer Sicherheit gestellt. Diese Arbeit soll exemplarisch die rechtlichen Voraussetzungen vorstellen, welche vom deutschen Grundgesetz vorgegeben werden, um daraus technische, kryptografische sowie organisatorische Eigenschaften abzuleiten. Darauf aufbauend wird eine Vorgehensweise weiterentwickelt, die es ermöglichen soll, Internetwahlverfahren in unkontrollierten Umgebungen anhand spezifischer Angreifermodelle zu bewerten. Nachdem eine breite Übersicht praktisch eingesetzter sowie akademischer Verfahren vorgestellt wurde, sollen eben diese Verfahren anhand der vorher entwickelten Vorgehensweise bewertet und vergleichend dargestellt werden. Zum Abschluss wird das gegenwärtig in Deutschland eingesetzte Briefwahlverfahren, welches sozusagen die Urform von Wahlen in unkontrollierten Umgebungen darstellt, untersucht und die Ergebnisse in die Diskussion eingebettet.

Simon Gölz, Michael P. Heinl, Christoph Bösch: *Trustworthy Elections?*, Eine Übersicht aktueller Verfahren & Probleme von Internetwahlen in unkontrollierten Umgebungen, © März 2018 (inklusive geringfügiger Aktualisierungen bzgl. des Briefwahlprozesses vom September 2019).

WEBSITE:
uulm.de/in/vs

INHALTSVERZEICHNIS

TABELLENVERZEICHNIS

ABBILDUNGSVERZEICHNIS

1 EINLEITUNG

Wahlen stellen den Kern der Demokratie dar. Sie sind das zentrale Element der Legitimation politischer Entscheidungsträger. Eine hohe Wahlbeteiligung ist für eine moderne Demokratie deshalb unabdingbar. Diesem normativen Anspruch steht heutzutage die Realität gegenüber, in welcher die Wahlbeteiligungen immer weiter abnehmen. Schon lange gibt es deshalb Forderungen verschiedenster Akteure, das aktuelle Wahlsystem zu modernisieren und Onlinewahlen anzubieten, da man sich dadurch eine höhere Wahlbeteiligung verspricht. Ob die These, dass Onlinewahlen die Wahlbeteiligung erhöhen, stimmt oder nicht, wird nicht Gegenstand dieser Abhandlung sein. Viel mehr wird sie sich der Frage widmen, ob bestehende Verfahren, die Wahlen über das Internet ermöglichen, den rigiden Anforderungen des Grundgesetzes genügen.

Viele Staaten haben Internetwahlsysteme auf unterschiedlichen Ebenen bereits eingesetzt. Dazu gehört allen voran Estland, die nicht nur als erstes Land offizielle Online-Wahlen durchgeführt haben, sondern dies auch noch ununterbrochen tun und die Anwendung kontinuierlich von der Kommunal- bis hin zur Europawahl ausgeweitet haben. Aber auch Länder, wie z. B. die Schweiz, Norwegen, die Vereinigten Staaten von Amerika, das Vereinigte Königreich und viele andere haben bereits erste Versuche unternommen, Internetwahlsysteme einzuführen [1]. Die Schlüsse, die aus den vielen bereits durchgeführten Pilotprojekte gezogen wurden, waren unterschiedlicher Natur [1]. Das liegt zum einen daran, dass viele verschiedene Systeme zum Einsatz kamen. Zum anderen aber auch daran, dass es für die Bewertung der Systeme keinen einheitlichen Standard gab und nach wie vor nicht gibt. Eine Gemeinsamkeit aller bisher produktiv eingesetzten System war es, dass teilweise erhebliche Sicherheitsbedenken bestanden. Zwar lassen sich hierfür z. B. Standards der International Organization for Standardization (ISO) heranziehen und auf einzelne Teilaspekte anwenden [2], ein ganzheitlicher und speziell auf Internetwahlsysteme zugeschnittener Standard ist bei der kritischen Bedeutung, die Wahlen in demokratischen Gesellschaft spielen, mittelfristig jedoch unabdingbar. Hierfür wird in Kapitel 3 die von Langer et al. [3] eingeführte Methodik erweitert. Auch in Deutschland und speziell in Baden-Württemberg wird parallel zur Entstehung dieser Arbeit die Diskussion um die Einführung von Internetwahlen an Universitäten geführt [4]. Die Kernüberlegung, die ursprünglich zu dieser Projektarbeit führte, ist die Frage, warum Gremienwahlen an deutschen Universitäten nicht auch online möglich sein sollten, obwohl Estland sogar landesweit Onlinewahlen durchführen kann. Bereits im Jahre 2013 hat das Thüringer Oberlandesgericht außerdem - der in Unterabschnitt 2.2.3 erläuterten Karlsruher Entscheidung für Bundestagswahlen zum Trotz - Online-Wahlen im Hochschulbereich als zulässig erachtet, wenn die entsprechende Wahlordnung dies zulässt [5].

Neben den klassischen Papierwahlen wird in Deutschland auch vermehrt das Briefwahlverfahren eingesetzt, obwohl verschiedene Gruppen seit langer Zeit Sicherheitsbedenken bzgl. dieser Art der Wahl artikulieren. Da die Briefwahl trotz dieser Bedenken nach wie vor zugelassen ist, werden ihre Prozesse und Sicherheitsmerkmale in dieser Arbeit ebenfalls kurz erläutert und bewertet, um die entsprechenden Attribute anschließend mit denen der gängigen Internetwahlverfahren zu vergleichen.

1.1 ABGRENZUNG

Der Begriff *e-Voting* umfasst zwar Internetwahlverfahren, schließt jedoch auch ortsgebundene elektronische Wahlmaschinen ein, die beispielsweise die Auszählung der Stimmen erleichtern und beschleunigen sollen. Diese Arbeit behandelt explizit nur ortsunabhänige Verfahren, die Wahlen über das Internet ermöglichen sollen (in der Fachliteratur häufig *Remote Electronic Voting*, *Internet Voting* oder kurz *I-Voting* genannt). Diese Internetwahlsysteme können weiterhin in zwei größere Unterkategorien unterteilt werden. Zum einen gibt es Systeme, deren Endpunkte bzw. Clients sich in einer kontrollierten Umgebung, also z. B. einem Wahllokal, befinden (*Controlled Environment*). Ein Beispiel hierfür ist das Okaloosa Distance Balloting Project (ODBP), welches im Ausland stationierten US-amerikanischen Militärangehörigen die Wahl über das Internet ermöglichte, indem sie spezielle Wahlcomputer in Hotels benutzten. Die andere Kategorie umfasst Endgeräte, die von Wahlberechtigten zur Abgabe ihrer Wahl benutzt werden könnten, sich jedoch jeglicher Kontrolle der Wahlbehörden entzieht (*Uncontrolled Environment*) [1]. Wahlmaschinen und andere elektronische Systeme, die ortsgebundenen Wahlen zuzuordnen sind (also auch sämtliche Internetwahlsysteme in kontrollierten Umgebungen), sind explizit nicht Bestandteil dieser Arbeit.

Stattdessen werden, wie in Abbildung 1 schematisch dargestellt, ausschließlich Internetwahlverfahren betrachtet, die für den Einsatz in unkontrollierten Umgebungen spezifiziert sind. Dies können zu einen Verfahren sein, die auf Seite des Benutzers auf einer Software beruhen, die lokal auf dem Rechner installiert wird (*Application-based*). Zum anderen ist es aber auch möglich, dass der Benutzer zur Wahl lediglich seinen Web-Browser ohne zusätzlich installierte Software benötigt (*Web-based*).

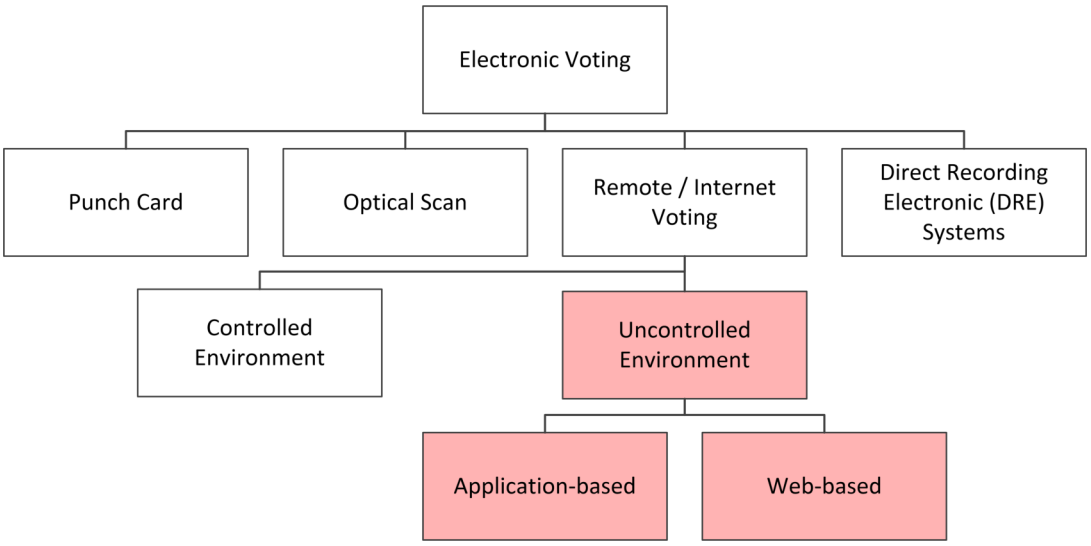


Abbildung 1: Einordnung der Arbeit (rot markiert) in Anlehnung an die Kategorisierungen der U.S. Election Assistance Commission [1] und Mursi et al. [6].

1.2 DEFINITIONEN

Im Folgenden sollen die in dieser Arbeit eingesetzten Begrifflichkeiten erläutert werden.

Internetwahlprotokoll: *Theoretisch / akademisches Verfahren, welches zwar in einer wissenschaftlichen Arbeit beschrieben und evtl. als Proof-of-Concept implementiert ist, es aber noch keine produktiv nutzbare Implementierung bzw. keine praktische Erfahrung im Rahmen eines tatsächlichen, nennenswerten Einsatzes gibt.*

Internetwahlsystem: *Für den produktiven Einsatz implementiertes und eingesetztes Verfahren, über das bereits praktischen Erfahrungswerte vorliegen.*

Internetwahlverfahren: *Wird in der vorliegenden Arbeit als zusammenfassender Oberbegriff für beide, oben spezifizierten Begrifflichkeiten verwendet.*

Stimmzettel: *Von der Wählerin[1] erstellte Nachricht, die die Wahlentscheidung enthält. Kann verschlüsselt sein.*

Wahlentscheidung: *Die von der Wählerin tatsächlich getroffene Entscheidung über eine im Wahlvorgang verfügbare Wahloption (z. B. Kandidatin X).*

1 Die in dieser Arbeit in jedem Kapitel abwechselnd verwendeten Feminina und Maskulina sind ausdrücklich generisch zu verstehen.

1.3 FORSCHUNGSFRAGEN / VORGEHENSWEISE

Diese Arbeit bietet einen Überblick über die aktuellen Entwicklungen im Forschungsbereich der Internet-Wahlsysteme, also der Durchführung demokratischer Wahlen über das Internet. Als Grundlage für die Auswahl und Bewertung bestimmter Verfahren, die später genauer analysiert werden, werden zunächst juristische Anforderungen an Wahlen allgemein ermittelt und daraus so weit möglich technische, kryptografische und organisatorische Anforderungen an Internet-Wahlen im Speziellen abgeleitet. Zur Unterstützung wird die Methodik von Langer et al. [3] übernommen und erweitert. Aus einer aktuellen Übersicht publizierter Forschungsprojekte und -ergebnisse werden auf Basis der erörterten Anforderungen und der erweiterten Methodik die vielversprechendsten Lösungen ausgewählt und näher beleuchtet. Die Forschungsfragen, denen diese Arbeit zu Grunde liegt, lauten demnach:

- Wie sieht der momentane Forschungsstand im Bereich Internetwahlverfahren für unkontrollierte Umgebungen aus?
- Wie könnte eine Erweiterung der von Langer et al. eingeführten Bewertungsmethodik aussehen?
- Gibt es ein Internetwahlsystem, das unter Annahme des Angreifermodells für Wahlen erster Ordnung sicherer, mindestens aber genauso sicher wie der momentan eingesetzte Briefwahlprozess ist?

1.4 VERWANDTE ARBEITEN

Der folgende Abschnitt fasst die während der Literaturrecherche gefundenen Arbeiten zusammen, die sich mit ähnlichen Fragestellungen befassen, wie die vorliegende Arbeit. Der erste Abschnitt befasst sich dabei mit Übersichten von Internetwahlverfahren, der zweite Abschnitt mit verschiedenen Frameworks zur Beurteilung der Sicherheit eben dieser. Da die vorgestellten Arbeiten teilweise Anteile beider Kategorien enthalten, ist eine eindeutige Einordnung nicht immer möglich. Die Kategorisierung findet deshalb auf Basis des dominierenden Anteils statt.

1.4.1 Surveys

In ihrer zum Zeitpunkt der Entstehung der vorliegenden Arbeit jungen Übersicht „Survey on Remote Electronic Voting“ [7] beschreiben Schneider, Meter und Hagemeister die Basis-Anforderungen an und die wichtigsten Kategorien von Internetwahlverfahren. Exemplarisch werden einige Internetwahlprotokolle und -systeme sowie etwaige Schwachstellen kurz erläutert. Weldemariam und Villafiorita gehen in ihrer Arbeit „A Survey: Electronic Voting Development and Trends“ [8] auch auf die Schwierigkeiten von Internetwahlverfahren ein und verweisen auf einige konkrete Protokolle und Systeme. Der Fokus der Arbeit liegt jedoch nicht auf Internetwahlverfahren, sondern auf elektronischen Wahlmaschinen. Mursi et al. [6] geben einen Überblick über Stärken und Schwächen sämtlicher Kategorien von elektronischen Wahlverfahren. Internetwahlverfahren machen dabei nur einen sehr kleinen Teil aus. Die Election Assistance Commission (EAC) der Vereinigten Staaten von Amerika listet in ihrer 149 Seiten starken „Survey of Internet Voting“ [1] die in den USA, Kanada, Europa und Ozeanien zwischen Januar 2000 und November 2011 offiziell eingesetzten Internetwahlsysteme auf und fasst die Ein- bzw. Durchführung der Wahlen aus legislativer, organisatorischer und technischer Perspektive kurz zusammen. In ihrem Fazit formuliert die EAC gleich am Anfang, dass insbesondere eine nähere Betrachtung unterschiedlicher Internetwahlsysteme sowie ein internationaler Standard für Internetwahlsysteme in unkontrollierten Umgebungen notwendig wäre.

1.4.2 Frameworks

Neumann und Volkamer [2] präsentieren ein ausführliches Framework zur Bewertung von Internetwahlsystemen. Dazu leiten sie aus rechtlichen Anforderungen und Prinzipien technische Anforderungen ab und beschreiben bzw. verweisen auf geeignete Methoden, wie z. B. ISO Standards, um zu ermitteln, wie ausgeprägt die jeweilige Anforderung beim vorliegenden Internetwahlverfahren ist. In seiner Dissertation [9] beschreibt Neumann weiterhin eine Methode, um die Erfüllung der technischen Anforderungen durch Internetwahlverfahren mit Hilfe von Monte-Carlo-Simulationen weiter zu quantifizieren. Sampigethaya und Poovendran [10] entwickeln ein System zur Klassifikation verschiedener Sicherheits-Anforderungen (*General Security Requirements*, *Adversary Counter-attack Requirements* und *System Implementation Requirements*) und präsentieren darauf aufbauend eine Übersicht, welche (ausschließlich kryptografischen) Internetwahlsysteme bzw. -protokolle welche Anforderungen zu wel-

chem Grad erfüllen. Langer et al. [3] entwerfen ein Framework, das es unter Einbeziehung sämtlicher relevanter sicherheitskritischer Annahmen ermöglichen soll, die Erfüllung von Sicherheitsanforderungen durch Internetwahlverfahren möglichst genau zu spezifizieren. Die Arbeit beschränkt sich dabei auf die beiden exemplarischen Sicherheitsanforderungen des Wahlgeheimnisses sowie der (individuellen und universellen) Verifizierbarkeit. Ondrisek liefert in ihrer Dissertation [11] eine Übersicht verschiedener elektronischer Wahlverfahren. Neben Wahlcomputern behandelt sie zwar auch Internetwahlverfahren, geht jedoch nicht im einzelnen auf die jeweiligen Protokolle und Systeme ein, sondern beschreibt diese jeweils nur in Kategorien. Auch die von ihr eingeführte Methode der *E-Voting-System Security Optimization* (EVSSO) bezieht sich hauptsächlich auf Wahlmaschinen und weniger auf Internetwahlverfahren.

2 ANFORDERUNGEN AN WAHLEN

In diesem Kapitel sollen die Anforderungen an Internetwahlen ausgearbeitet werden. Dabei dienen die juristischen Voraussetzungen für politische Wahlen als Grundlage für diese Anforderungen und werden als erstes vorgestellt. Die rechtlichen Voraussetzungen für politische Wahlen unterscheiden sich abhängig vom jeweiligen Nationalstaat, in dem die Wahl durchgeführt werden soll. Da es unmöglich ist, die juristischen Anforderungen aller Nationalstaaten zu betrachten, soll in diesem Kapitel exemplarisch die deutsche Rechtslage als Ausgangspunkt dienen. Die Wahlgrundsätze des deutschen Wahlrechts sind allerdings identisch zu allen Mitgliedstaaten der EU. Somit unterscheiden sich die Anforderungen höchstens in deren Auslegung. Im zweiten Schritt werden aus diesen allgemeinen Anforderungen spezielle Anforderungen an Internetwahlen abgeleitet.

2.1 KATEGORISIERUNG VON WAHLEN

Neben den politischen Wahlen, die den Grundpfeiler der Demokratie darstellen, existieren selbstverständlich auch andere Arten von Wahlen, deren Einflussbereiche sehr unterschiedlich und meistens weitaus kleiner sind, als der Einfluss offiziell durchgeführter staatlicher Wahlen. Zwar basieren auch andere Wahlen nicht selten auf gewissen rechtlichen Grundlagen, zu nennen sind hier z. B. Betriebsratswahlen nach dem Betriebsverfassungsgesetz, jedoch sind die rechtlichen Vorschriften, die die Durchführung dieser Wahlen reglementieren, bei weitem nicht so umfangreich und restriktiv wie die Wahlen zu staatlichen Organen wie z. B. dem Bundestag.

Helbach [12] kategorisiert Wahlen nach einem in der Wahlforschung gängigen System, das sie entsprechend ihres Einflussbereichs in insgesamt drei unterschiedliche Ebenen einordnet. Politische Wahlen, wie z. B. Wahlen zum Bundestag, den Landesparlamenten oder dem Europaparlament, fasst er als *Wahlen erster Ordnung* zusammen. Andere gesetzlich vorgeschriebene Wahlen mit eher eingeschränktem Einfluss und weniger resktriktiven Anforderungen, wie z. B. Sozialwahlen oder Personal- bzw. Betriebsratswahlen, nennt er *Wahlen zweiter Ordnung*. Die letzte Kategorie der *Wahlen dritter Ordnung* umfasst zum z. B. Vereins- und Verbandswahlen, die zwar teilweise rechtlich vorgeschrieben sind, deren Rahmenbedingungen aber in einer Satzung individuell festgelegt werden können, sodass z. B. das Geheimnis der Wahl nicht zwingend vorgeschrieben ist und die Wahl auch offen erfolgen kann.

Andere Formen der Wahl können durchaus weitere Eigenschaften bzw. Anforderungen mit sich bringen. Bei Abstimmungen im Rahmen der Hauptversammlung von Aktiengesellschaften können Stimmen beispielsweise auf andere natürlich oder juristische Personen übertragen werden [13]. Solche Sonderformen der Wahl werden im Verlauf der Arbeit nicht weiter betrachtet. Zwar erhebt diese Arbeit vor allem vom juristischen Blickwinkel aus betrachtet keinen Anspruch auf Vollständigkeit, um die rechtlichen Anforderungen trotzdem so ausführlich wie im Rahmen dieser Arbeit möglich abzudecken, beschäftigt sich dieses Kapitel mit Wahlen erster Ordnung, für deren Durchführung im Allgemeinen die strengsten Regeln gelten.

2.2 RECHTLICHE GRUNDLAGEN

Im Folgenden sollen aus dem deutschen Grundgesetz, der Bundeswahlordnung, dem Bundeswahlgesetz sowie der höchstrichterlichen Rechtsprechung des Bundesverfassungsgerichts Anforderungen abgeleitet werden, um ein grundlegendes Verständnis für die Beurteilung der Sicherheit von Wahlsystemen zu schaffen. Um zu verdeutlichen, dass die deutschen bzw. europäischen Voraussetzungen wesentlich restriktiver sind, als die des Völkerrechts, wird das Völkerrecht ebenfalls kurz betrachtet.

2.2.1 Grundgesetz

Die Rahmenbedingungen und grundsätzlichen Anforderungen des deutschen Wahlrechts werden im Grundgesetz festgelegt. Ein wichtiger Bestandteil ist der Grundsatz der Volkssouveränität. Durch diesen Grundsatz wird festgeschrieben, wozu Wahlen dienen sollen. So heißt es im Artikel 20 Absatz 2 des deutschen Grundgesetzes:

> Alle Staatsgewalt geht vom Volke aus. Sie wird vom Volke in Wahlen und Abstimmungen und durch besondere Organe der Gesetzgebung, der vollziehenden Gewalt und der Rechtsprechung ausgeübt.

Das bedeutet, dass die Gewalt von repräsentativ gewählten Organen ausgeht. Dieser Artikel sichert somit jedem Bürger das Wahlrecht zu. Grundsätzlich ist im Grundgesetz nicht spezifiziert, welche Staatsorgane vom Volk gewählt werden. Allerdings wird die Besetzung eines besonders wichtigen Staats- bzw. Verfassungsorgans, die des Bundestages, durch das Grundgesetz geregelt [14]. So heißt es in Artikel 38 Grundgesetz:

> 1.) Die Abgeordneten des deutschen Bundestages werden in allgemeiner, unmittelbarer, freier, gleicher und geheimer Wahl gewählt. Sie sind Vertreter des ganzen Volkes, an Aufträge und Weisungen nicht gebunden und nur ihrem Gewissen unterworfen.
> 2.) Wahlberechtigt ist, wer das achtzehnte Lebensjahr vollendet hat; wählbar ist, wer das Alter erreicht hat, mit dem die Volljährigkeit eintritt.
> 3.) Das Nähere bestimmt ein Bundesgesetz.

Wie aus Absatz 1 hervorgeht, sind an die Wahl die Anforderungen allgemein, unmitelbar, frei, gleich und geheim zu stellen. Diese Anforderungen werden auch Wahlrechtsgrundsätze oder Wahlgrundsätze genannt. Im Folgenden soll auf die Bedeutung dieser Grundsätze weiter eingegangen werden.

- **Allgemein:** Damit ist gemeint, dass alle wahlberechtigten Personen ihre Wahl abgeben können. Dies muss unabhängig von Faktoren wie Bildung, Beruf, Sprache, körperlicher Verfassung, Einkommen oder Besitz sein. Die grundsätzlichen Voraussetzungen für das Wahlrecht nach dem Bundeswahlgesetz Paragraf 12 sind davon jedoch ausgenommen. Im Allgemeinen muss eine Wählerin also das Wahlalter erreicht haben und die deutsche Staatsangehörigkeit besitzen.
- **Unmittelbar:** Unmittelbar heißt, dass die Wählerin ihre Stimme direkt abgibt. Es darf also keine Zwischenschaltung fremden Willens zwischen der Stimmabgabe und dem Willen der Wählerin stehen. Dies schließt Wahlmänner oder Vertreter bei der Wahl aus [14, 15].
- **Frei:** Jede Wählerin gibt ihre Stimme frei von öffentlicher Gewalt und privatem Druck ab. Das bedeutet, es muss sichergestellt werden, dass die Wählerin während der Stimmabgabe nicht von außen beeinflusst wird und ihren eigenen unverfälschten Willen äußern kann. Um dies zu gewährleisten, geht das Prinzip der freien Wahl Hand in Hand mit dem Prinzip der Geheimhaltung.
- **Gleich:** Jede Stimme zählt gleich viel. Um dies zu gewährleisten, darf jede wahlberechtigte Person genau eine Stimme mit gleichem Zählwert abgeben. Außerdem müssen die äußeren Einflüsse gleich sein. So muss beispielsweise jeder die gleichen Wahlmöglichkeiten besitzen.
- **Geheim:** Die Wählerin muss ihre Stimme so abgeben, dass andere keine Kenntnis darüber erhalten, was sie wählen möchte oder was sie gewählt hat. Die Wählerin kann nicht nur geheim wählen, sondern sie ist dazu verpflichtet, da dies neben dem Schutz der Wählerin, auch dem Schutz des Prinzips der freien Wahl dient .

Durch das Prinzip der Volkssouveränität, wie in Artikel 28 Absatz 1 Grundgesetz festgeschrieben, werden die Wahlgrundsätze aus Artikel 38 Absatz 1 für alle politischen Wahlen innerhalb Deutschlands als Maßstab festgeschrieben und gelten deshalb für alle politischen Wahlen auf Bundes-, Länder-, Kreis- und Kommunalebene [14, 15]. Des Weiteren wird in Absatz 3 beschrieben, was genau in speziellen Bundesgesetzen geregelt wird. Das wichtigste dieser Bundesgesetze ist das Bundeswahlgesetz, welches wiederum auf die Bundeswahlordnung verweist. Sie beschreiben unter anderem Sicherheitsanforderungen, wie sie für die jetzige Papierwahl gelten. Diese soll im nächsten Kapitel betrachtet werden.

2.2.2 Bundeswahlgesetz und Bundeswahlordnung

Das Bundeswahlgesetz und die Bundeswahlordnung legen für die genannten, im Grundgesetz verankerten, eher abstrakten Wahlgrundsätze tatsächliche Vorschriften für die Durchführung von regulären Papierwahlen fest. Da die Auflistung aller Maßnahmen den Umfang dieser Arbeit sprengen würde, konzentriert sie sich im Folgenden auf die dort aufgeführte Anforderung der öffentlichen Verifizierbarkeit, die aufgrund ihrer Übertragbarkeit auf

elektronische Wahlen (siehe dazu Unterabschnitt 2.2.3) für die vorliegende Arbeit von besonderer Bedeutung ist. In Paragraf 54 der Bundeswahlordnung wird jedem, egal ob wahlberechtigt oder nicht, das Recht eingeräumt, den Wahlvorgang zu beobachten. Der Wahlvorgang umfasst die Stimmabgabe sowie die Stimmauszählung. So heißt es im Paragraf 54 der Bundeswahlordnung:

> Während der Wahlhandlung sowie der Ermittlung und Feststellung des Wahlergebnisses hat jedermann zum Wahlraum Zutritt, soweit das ohne Störung des Wahlgeschäfts möglich ist.

Durch diesen Paragrafen soll es jedem möglich gemacht werden, sich von der Korrektheit der Wahl zu überzeugen. Offensichtlich ist es aber einer einzelnen Person nicht möglich, die komplette Wahl zu verifizieren, da diese Person sich sonst zur gleichen Zeit an mehreren Orten aufhalten müsste. Also kann die Wahl nur durch die Korrektheit der einzelnen Teilergebnisse verifiziert werden, weswegen zur vollständigen Verifikation eine Personengruppe nötig ist [16].

2.2.3 Urteil des Bundesverfassungsgerichts

Wie bereits in Unterabschnitt 2.2.2 dargestellt, müssen Wahlen durch die allgemeine Öffentlichkeit verifizier- und nachvollziehbar sein. Mit seinem Urteil vom 3. März 2009 [17] entschied das Bundesverfassungsgericht, dass der durch die Bundeswahlgeräteverordnung[1] ermöglichte, vereinzelte Einsatz von Wahlmaschinen während der Bundestagswahl 2005 verfassungswidrig war. Als Begründung führten die Richter an, dass es den Bürgerinnen und Bürgern auch ohne spezielle Fachkenntnisse möglich sein muss, die vom Gerät tatsächlich gewertete Stimme zu überprüfen. Eine Anzeige auf einem Display oder ein Papierausdruck allein reichen hierfür nicht aus, da diese kein eindeutiger Beleg dafür sind, dass die Maschine tatsächlich korrekt gerechnet, das heißt die abgegebenen Stimmen korrekt aufaddiert hat. Nun bezieht sich dieses Urteil zwar lediglich auf physische Wahlmaschinen, die die Auszählung des Wahlergebnisses vor Ort elektronisch unterstützen sollen. Da die von den Richtern geschilderten Anforderungen jedoch direkt aus der Verfassung abgeleitet sind, kann davon ausgegangen werden, dass sie ebenfalls (und aufgrund des für Wählerinnen und Wähler noch weniger greifbaren Wesens des Internets wohl insbesondere) für Internetwahlverfahren gelten.

2.2.4 Völkerrecht

Im Völkerrecht ist das Recht auf freie Wahlen in Artikel 3 Zusatzprotokoll (ZP) 1 spezifiziert. Demnach ist es die Verpflichtung eines Staates, für gesetzgebende Körperschaften freie Wahlen zu gewährleisten. Mit gesetzgebenden Körperschaften sind Parlamente und sonstige gesetzgebende Institutionen gemeint, nicht aber Präsidentschaftswahlen. Artikel 3 ZP 1 nennt dabei die Wahlgrundsätze Freiheit sowie Geheimheit, allerdings gelten darüber hinaus auch Allgemeinheit und Gleichheit der Wahl als anerkannte Wahlgrundsätze. Des Weiteren ist im Völkerrecht erwähnt, dass die Wahlgrundsätze auch über den eigentlichen Wahlakt hinaus eingehalten werden müssen. Das eigentliche Wahlrecht darf der Staat dabei nach seinen eigenen Vorstellungen ausgestalten, was allerdings nicht zu unverhältnismäßigen Beschränkungen sowie zur Verletzung der Wahlgrundsätze führen darf [18]. Wie man sehen kann, sind die völkerrechtlichen Anforderungen an Wahlen wesentlich schwächer ausgelegt, als es der deutschen Rechtsprechung für Bundestagswahlen entspricht, weswegen für die weitere Arbeit die deutschen Wahlgrundsätze als Basis für die technischen Anforderungen herangezogen werden.

2.3 ALLGEMEINE ANFORDERUNGEN AN WAHLVERFAHREN

Nachdem im vorherigen Abschnitt die rechtlichen Grundlagen erklärt wurden, sollen in diesem Abschnitt nun daraus resultierende Anforderungen noch einmal zusammengefasst werden. Die Einteilung geschieht grob anhand der Wahlgrundsätze. Eine allgemeingültige Einordnung ist allerdings nicht möglich, da manche Anforderungen aus mehreren Wahlrechtsgrundsätzen abgeleitet werden können. Zu beachten ist, dass nicht alle der Anforderungen technisch umgesetzt werden können, sondern teilweise auch durch organisatorische Maßnahmen gewährleistet werden müssen.

1 Da der Fokus der Bundeswahlgeräteverordnung auf elektronischen Wahlgeräten liegt und sie in ihrer derzeitigen Fassung als verfassungswidrig [17] gilt, wird in dieser Arbeit nicht weiter auf sie eingegangen.

2.3.1 Geheim und frei

Um eine geheime und freie Wahl durchführen zu können, muss offensichtlich das **Wahlgeheimnis** eingehalten werden. Das bedeutet, dass sichergestellt werden muss, dass während sowie nach der Wahl keine Zuordnung zwischen der Identität der Wählerin und der Wahlentscheidung möglich sein darf. Es ist davon auszugehen, dass die Wahlentscheidung der Wählerin über deren komplette Lebenszeit geheim bleiben muss.

Eine Anforderung mit ganz ähnlichen Gründen ist, dass **keine vorläufigen Zwischenergebnisse**[2] preisgegeben werden. Wenn z. B. eine bestimmte Wahlmöglichkeit zum Zeitpunkt der Veröffentlichung des Zwischenergebnisses bereits weitaus mehr Stimmen auf sich vereinen kann, kann einer Wählerin mit höherer Wahrscheinlichkeit als normal eine gewisse Wahloption zugeordnet werden, unter der Voraussetzung, dass ein Dritter weiß, wer wann gewählt hat. Bei Präsenzwahlen ist dies relativ einfach zu ermitteln, indem man kontrolliert, wer ins Wahllokal geht. Im Extremfall hat zum Zeitpunkt der Veröffentlichung des Zwischenergebnisses nur eine Person gewählt oder alle Personen haben dasselbe gewählt. Dann ist das Wahlgeheimnis direkt gebrochen.

Außerdem muss das Wahlverfahren die **Quittungsfreiheit** (manchmal auch Belegfreiheit genannt) gewährleisten. Ein Wahlverfahren ist quittungsfrei, wenn es der Wählerin nicht möglich ist (selbst mit ihrer Mithilfe), ihre Wahl gegenüber Dritten zu beweisen. Es ist damit allerdings nicht gemeint, dass die Wählerin keine Quittung in irgendeiner Form bekommen darf. Eine Anwendung einer zulässigen Quittung ist beispielsweise, dass die Wählerin (und nur die Wählerin) später nachprüfen kann, ob ihre Stimme in das Wahlergebnis eingeflossen ist (siehe individuelle Verifizierbarkeit).

Für die Freiheit ist es wichtig, dass die Wählerin frei von körperlichem, physischem und finanziellem Druck wählen kann. Deswegen darf es nicht möglich sein, dass eine Wählerin ihre Stimme verkaufen kann (**Unmöglichkeit von Stimmenverkauf**) oder dass die Wählerin durch Erpressung zu einer bestimmten Wahl gezwungen wird (**Nicht-Erpressbarkeit**). Im Unterschied zur Quittungsfreiheit soll es unmöglich sein, dass eine Dritte die Wahlentscheidung einer Wählerin in irgendeiner Weise beeinflussen kann. Dabei muss die Angreiferin nicht unbedingt die tatsächliche Wahl erfahren. Eine Möglichkeit ist es, der Wählerin eine oder mehrere Wahlmöglichkeiten vorzuenthalten oder sie dazu zu zwingen, eine zufällige Wahl zu treffen. Ist die Wählerin bei der Anwendung eines gegebenen Wahlverfahrens jedoch nicht in der Lage, ein bestimmtes Verhalten, welches die Wahl beeinflusst, nachzuweisen, kann der Erpresser bzw. der Stimmenkäufer nicht nachvollziehen, ob die Wählerin seine Anweisungen befolgt hat. Ein Beispiel ist ein Wahlverfahren, das die Wahlmöglichkeiten $W = \{1, 2, \ldots, n\}$ in zufälliger Reihenfolge (wird unter Gleichverteilung gezogen) untereinander anordnet und zur individuellen Verifizierbarkeit eine Quittung ausgibt, auf welcher die Position der Wahl auf dem Stimmzettel ersichtlich wird. Wenn der Erpresser die Wählerin anweist, die Position $k \in W$ zu wählen, kann der Erpresser dieses Verhalten später kontrollieren und die Wählerin mit einer Wahrscheinlichkeit von $1 - \frac{1}{n}$ dazu zwingen, nicht ihre eigene Wahl zu treffen. Ist die Wählerin nicht in der Lage, ein bestimmtes Verhalten nachzuweisen, kann sie trotz dieser Bedrohungen frei wählen, was daran liegt, dass der Stimmenkäufer nicht weiß, ob es sich lohnt, zu bezahlen bzw. der Erpresser nicht weiß, ob er seine Drohung wahr machen soll.

Ein Sonderfall, der nicht zur Erpressbarkeit zählt, ist, dass die Wählerin nicht gezwungen werden kann, sich zu enthalten (**keine erzwungene Enthaltung**). Diese Anforderung ist schwer zu realisieren, da es immer möglich ist, Wählerinnen während der Wahlphase an der Teilnahme an der Wahl zu hindern. Dies gilt insbesondere bei den derzeitig eingesetzten Präsenzwahlen. Bei Onlinewahlen kann dies jedoch auch nicht komplett ausgeschlossen werden. Für dieses Problem gibt es allerdings Lösungsansätze, welche dafür sorgen, dass der Angriff für große Personenzahlen nicht skalierbar ist. Beispielsweise indem dafür gesorgt wird, dass die Wahlphase über einen längeren Zeitraum möglich ist.

Eine weitere Gefahr für das Treffen einer freien Entscheidung ist der Einfluss durch Wahlwerbung (**keine Wahlwerbung**). Dies zu vermeiden, ist bei Präsenzwahlen durch ein Verbot von Wahlwerbung im Wahllokal einfach umzusetzen. Bei Onlinewahlen ist die Umsetzung allerdings schwer. Man kann zwar Wahlwerbung auf der Website verbieten, allerdings kann die Wählerin von überall aus wählen und so durch ihre Umgebung entsprechend beeinflusst werden [16, 19–21].

2.3.2 Unmittelbar und öffentlich verifizierbar

Um eine unmittelbare und verifizierbare Wahl zu gewährleisten, muss sichergestellt werden, dass jede Wählerin direkt eine Wahlmöglichkeit erhält. Zwischen den einzelnen abgegebenen Stimmen und dem Wahlergebnis steht nur die Aufsummierung aller dieser abgegebenen Stimmen. Diese Aufsummierung hat fehlerlos zu geschehen,

2 Diese Anforderung wird unter dem Aspekt der Gleichheit und der Geheimheit der Wahl betrachtet, da die gleiche Anforderung für die Einhaltung beider Kriterien wichtig ist.

weswegen die Anforderung der **Korrektheit** erfüllt sein muss. Dazu gehört auch die **Integrität** der Wahl. Die Integrität gewährleistet, dass jede Stimme, die korrekt abgegeben wurde, in das Ergebnis einfließt. Natürlich ist es wünschenswert, dass eine Wahl auch **verifizierbar** ist. Hierbei wird zwischen individueller Verifizierbarkeit und universeller Verifizierbarkeit unterschieden. Mit individueller Verifizierbarkeit ist gemeint, dass die Wählerin überprüfen kann, ob ihre Wahl korrekt in das Ergebnis eingeflossen ist. Das heißt jede Wählerin kann die Integrität der Wahl prüfen. Universelle Verifizierbarkeit heißt, dass jeder, auch eine Nichtwählerin, die Integrität des Wahlergebnisses überprüfen kann. Außerdem ist es wünschenswert, dass die Wahl auch nach der Auszählung der Stimmen verifizierbar bleibt. Dazu ist eine **Archivierung** der einzelnen Stimmen nötig. Ein weiterer Grund, der für eine Archivierung spricht, ist, bei einer neu entdeckten Sicherheitslücke nachvollziehen zu können, ob und ab wann vorangegangene Wahlen manipuliert wurden. Eine weitere essentielle Anforderung für den praktischen Einsatz ist die **Korrigierbarkeit**. Dies begründet sich dadurch, dass wenn die Korrektheit nachweislich nicht gegeben ist und die Wahl nicht korrigierbar ist, eine Neuwahl erfolgen muss. Somit könnte eine Angreiferin dies ausnutzen, um die **Robustheit** der Wahl anzugreifen [16, 20, 21].

2.3.3 Gleich

Damit eine Wahl den Wahlgrundsatz der Gleichheit erfüllt, wird vorausgesetzt, dass jede Wählerin das **gleiche Stimmgewicht** besitzt. Dazu muss sichergestellt werden, dass jede Stimme nur einmal gezählt wird und dass jede Wählerin genau eine Stimme abgeben kann. Hierzu zählt die Unmöglichkeit von Stimmenkauf sowie die nicht Erpressbarkeit, welche bereits in Unterabschnitt 2.3.1 erklärt wurden. Dies wird offensichtlich dadurch begründet, dass durch Erpressung und Stimmenkauf das Stimmgewicht beeinträchtigt werden kann. Ein weiterer Aspekt der Gleichheit ist, dass jede Wählerin unter gleichen Voraussetzungen wählen kann. Deswegen ist sicherzustellen, dass jede Wählerin die **gleichen Wahlmöglichkeiten** besitzt und diese auch gleich angezeigt werden. Ein Problem bei Onlinewahlen ist die unterschiedliche Auflösung der Geräte. So kann es sein, dass Wahlmöglichkeiten übersehen werden. Des Weiteren muss die Wahl unter den gleichen Voraussetzungen stattfinden. Das bedeutet, das Wahlsystem darf **keine vorläufigen Zwischenergebnisse**[3] preisgeben. Ansonsten kann für Wählerinnen, die später wählen, durch Taktieren ein Vorteil entstehen. Wenn beispielsweise zwei Wahlmöglichkeiten bereits weitaus mehr Stimmen auf sich vereinigen konnten, sodass die anderen Wahlmöglichkeiten keine oder nur eine geringe Chance haben würden, die Wahl zu gewinnen, könnte die Wählerin von ihrer eigentlichen Wahl abweichen und die Wahl so betrachten, als gäbe es nur noch diese zwei Alternativen. Dadurch hätte die Wählerin einen Vorteil gegenüber anderen an der Wahl Teilnehmenden [16, 19–21].

2.3.4 Allgemein

Die Allgemeinheit besagt, dass jeder unabhängig von Bildung und Besitz wählen darf. Deswegen ist es wichtig, dass der Wahlvorgang möglichst **benutzerfreundlich** ist. Dies umfasst zum einen das Benutzerinterface, aber auch den Wahlprozess sowie die spätere Verifikation. Diese sollten so intuitiv sein, dass jeder wählen kann und möglichst alle Personen die Möglichkeit haben, das Wahlergebnis zu verifizieren. Wichtig zu beachten ist, dass die Benutzerfreundlichkeit die Benutzung von Sicherheitsmechanismen und kryptografischen Methoden einschließt, da diese den Wahlvorgang sowie die Verifizierung des Wahlergebnisses im Vergleich zu einer Papierwahl oft erschweren. Eine weitere Anforderung ist, dass niemand, der stimmberechtigt ist, von der Wahl ausgeschlossen werden darf (**kein Auschluss**). Das bedeutet, dass nicht vorausgesetzt werden darf, dass die Wählerin mit dem Computer zurechtkommt. Speziell für Internetwahlen bedeutet dies, dass nicht vorausgesetzt werden kann, dass jeder Zugang zum Internet hat, weswegen die austragende Instanz Geräte und Internetanschlüsse bereitstellen muss. Außerdem ist die **Vollständigkeit** sicherzustellen. Das bedeutet, dass einerseits nur wahlberechtigte Personen wählen können, aber auch, dass keine wahlberechtigten Personen abgewiesen werden [14, 16, 19–21].

2.3.5 Sonstige Anforderungen

In diesem Abschnitt sollen alle Anforderungen genannt werden, die nicht direkt einem Wahlgrundsatz zugeordnet werden können. Eine dieser Anforderungen ist die **Robustheit**. Damit ist gemeint, dass niemand einen Abbruch der Wahl oder eine Wahlwiederholung provozieren kann. Dazu ist es nötig, dass zwischen behaupteter und tatsächlicher Manipulation unterschieden werden kann. Ist die Robustheit nicht gegeben, können Angrei-

3 Diese Anforderung wird auch unter dem Aspekt Gleichheit der Wahl betrachtet, da die gleiche Anforderung zu der Einhaltung beider Kriterien betrachtet werden muss.

ferinnen dies dazu nutzen, die Wahl zu verhindern oder zu verschieben. Beispielsweise wäre es denkbar, dass falls einer der Kandidatinnen laut den Umfragen keine Chance hat, gewählt zu werden, er die Ungültigkeit der Wahl erzwingen und darauf hoffen kann, dass die Wahlwiederholung zu einem für ihn günstigeren Zeitpunkt stattfindet [16, 21].

2.4 ANFORDERUNGEN AN INTERNETWAHLVERFAHREN

Nachdem alle vorgeschriebenen Anforderungen ausführlich und auf Basis ihrer rechtlichen Grundlage erörtert wurden, wird der Anforderungskatalog für die Beurteilung von Internetwahlverfahren jeweils zusammen mit einer kurzen Definition und einer Kategorie noch einmal in der folgenden Tabelle zusammengefasst. Die Kategorie beschreibt dabei, wie die entsprechende Eigenschaft eines Wahlverfahrens aufgrund ihrer jeweiligen Natur optimalerweise in einem praktisch einsetzbaren Wahlsystem realisiert werden sollte. „Kryptografie“ ist dabei die stärkste Kategorie und beinhaltet alle Anforderungen, die durch den Einsatz kryptografischer Methoden im Wahlprotokoll realisiert werden können. „Implementierung“ umfasst Anforderungen, die durch die konkrete Implementierung erreicht werden und unter „Organisatorisch“ sind alle nicht technisch, sondern durch organisatorische oder sonstige Maßnahmen zu gewährleistenden Anforderungen zusammengefasst. Manche Anforderungen sind aufgrund der später folgenden, weiteren Untergliederungen teilweise in zwei Kategorien angesiedelt.

Tabelle 1: Definition und Kategorisierung der Anforderungen an Internetwahlverfahren.

Anforderung	Definition	Kategorie
Individuelle Verifizierbarkeit	Ein System ist individuell verifizierbar, wenn die Wählerin überprüfen kann, dass ihre Stimme mit ins Ergebnis eingeflossen ist.	Kryptografie
Universelle Verifizierbarkeit	Im Gegensatz zur individuellen ermöglicht die universelle Verifizierbarkeit jedem, auch Personen, die nicht selbst gewählt haben, zu überprüfen, ob alle abgegebenen Stimmen korrekt erfasst wurden und die Integrität des Wahlergebnis gewährleistet ist.	Kryptografie
Wahlgeheimnis (Anonymität)	Es werden keinerlei Informationen preisgegeben, die mehr über die Wahl einer einzelnen Person bekanntgeben, als das Wahlergebnis.	Kryptografie
Quittungsfreiheit	Die Wählerin erhält keinerlei Quittung, die es ermöglichen würde, Dritten gegenüber zu beweisen, wie sie gewählt hat.	Kryptografie
Nicht-Erpressbarkeit & Unmöglichkeit von Stimmenkauf	Es darf der wählenden Person nicht ermöglicht werden, ihre Wahl Dritten gegenüber zweifelsfrei zu belegen, um zum einen Erpressbarkeit und zum anderen die Möglichkeit des Stimmenverkaufs auszuschließen.	Kryptografie / Organisatorisch
Gleiches Stimmgewicht	Alle abgegebenen, gültigen Stimmen fließen zu gleichen Teilen in das Gesamtergebnis ein. Keine wählende Person wird bevor- oder benachteiligt.	Kryptografie
Robustheit	Die Durchführbarkeit der Wahl ist nicht von einzelnen Personen oder kleinen Gruppen von Personen abhängig und trotzt auch so gut es geht äußeren Umständen.	Kryptografie / Implementierung
Gleiche Wahlmöglichkeiten	Es muss sichergestellt werden, dass allen Wählerinnen die selben Auswahlmöglichkeiten zur Verfügung stehen.	Implementierung
Korrigierbarkeit	Wählerinnen müssen die Möglichkeit haben, ihre Stimme, die entweder unabsichtlich oder z. B. aufgrund einer Erpressung entgegen ihrem freien Willen abgegeben wurde, innerhalb des Wahlzeitraums zu korrigieren.	Implementierung
Archivierung	Damit das Ergebnis der Wahl auch nach der finalen Auszählung noch verifiziert werden kann, können die dafür nötigen Daten archiviert werden.	Implementierung

Flexible Anwendungsmöglichkeit	Um die Akzeptanz durch die Wählerinnen zu gewährleisten, sollte das Wahlsystem an verschiedene Arten von Wahlen angepasst werden können, sodass keine ständige Umgewöhnung stattfinden muss.	Implementierung
Benutzbarkeit	Das System muss selbsterklärend und benutzerfreundlich sein, um die Akzeptanz der Wählerinnen sicherzustellen.	Implementierung
Verständnis	Das Wahlsystem kann von den Wählenden ohne spezielle Fachkenntnisse verstanden werden.	Organisatorisch
Keine Zwischenergebnisse	Vor Ablauf des definierten Wahlzeitraums darf das System keinerlei Information über die Verteilung der bisher abgegebenen Stimmen preisgeben.	Kryptografie / Organisatorisch
Keine erzwungene Enthaltung	Es soll nicht möglich sein, eine wahlberechtigte Person zwingen zu können, sich der Wahl zu enthalten.	Kryptografie / Organisatorisch
Kein Ausschluss	Das Wahlsystem darf keine Person, die Wahlrecht besitzt, von der Wahl ausschließen.	Organisatorisch
Keine Wahlwerbung	Am Wahltag darf keine Wahlwerbung stattfinden, die die freie Meinungsbildung der wahlberechtigten Personen beeinflusst.	Organisatorisch
Korrektheit	Es muss sichergestellt sein, dass die Liste aller wahlberechtigten Personen (z. B. Melderegister) fehlerlos ins System eingespeist wird.	Organisatorisch
Integrität	Es dürfen nur die Stimmen der wahlberechtigten Personen gezählt werden, die tatsächlich gewählt haben. Es dürfen keine Stimmen hinzugefügt, verändert oder gelöscht werden.	Organisatorisch
Vollständigkeit	Es muss sichergestellt sein, dass die Liste aller wahlberechtigten Personen (z. B. Melderegister) komplett ins System eingespeist wird.	Organisatorisch

3 BEWERTUNGSMETHODIK

Um die nachfolgende Evaluierung praktikabler zu gestalten, werden die in Tabelle 1 aufgelisteten Anforderungen zusammengefasst und auf Basis individueller Abhängigkeiten reduziert. Hierfür wird ein Bottom-Up-Prozess benutzt, der wie in Abbildung 2 aufgezeigt, einzelne, teilweise andere Anforderungen implizierende Kern-Anforderungen herausstellt, deren Erfüllung für die Realisierung der vorgegebenen Wahlgrundsätze des Grundgesetzes unbedingt notwendig ist. Aufeinander aufbauende, artverwandte Anforderungen, deren Kanten aneinander liegen, stellen dabei Anforderungen dar, deren Erfüllbarkeit direkt mit der Erfüllbarkeit der darunterliegenden Anforderungen zusammenhängt. Bei durch Pfeilen miteinander verbundenen Anforderungen trägt die Anforderung am hinteren Ende des Pfeils dazu bei, dass die Anforderung an der Spitze des Pfeils erreicht wird. Geschwungene Klammern am Ende eines Pfeiles bedeuten, dass die von den Klammern erfassten Anforderungen zur Erfüllung der Anforderung an der Spitze beiträgt, geschwungene Klammern an der Spitze des Pfeils bedeuten, dass die Anforderung am Ende des Pfeils zur Erfüllung der durch die Klammer erfassten Anforderungen beiträgt.

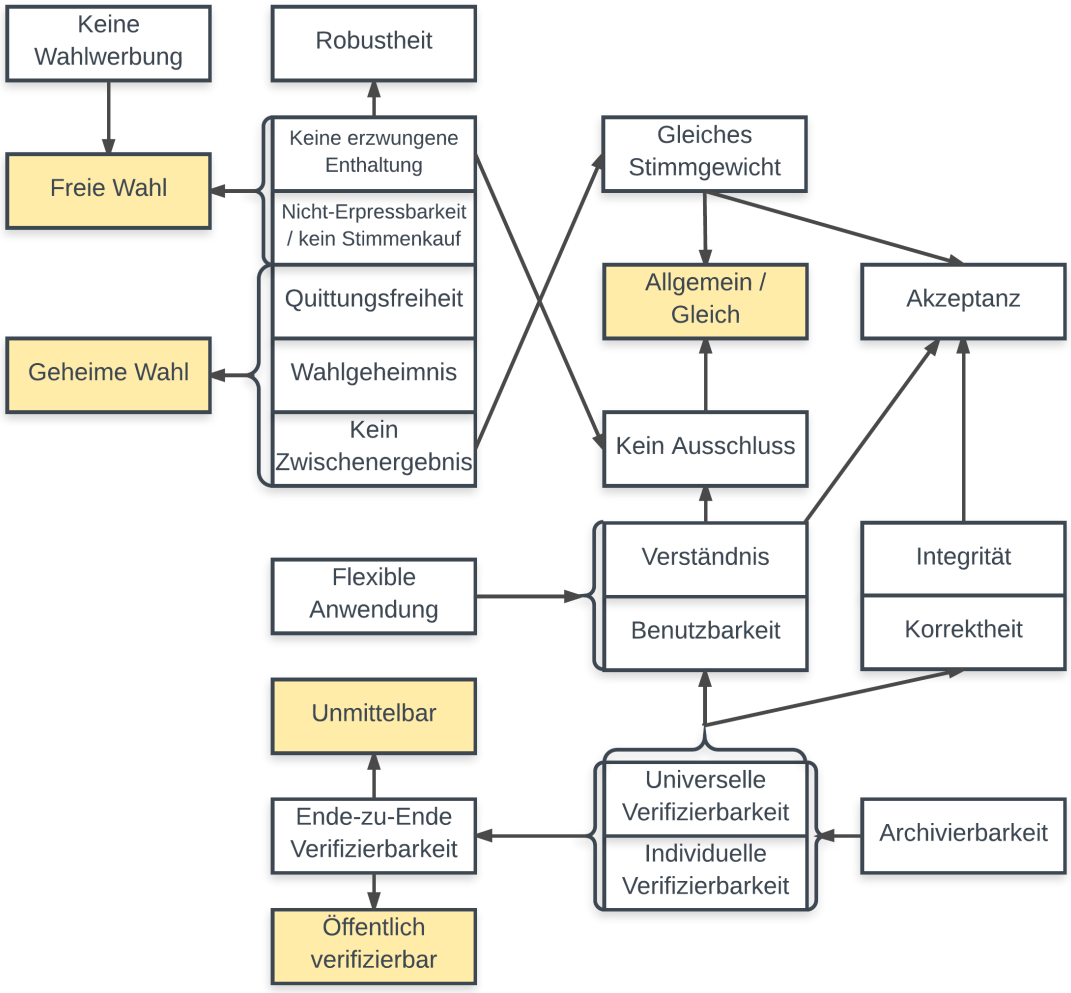


Abbildung 2: Bottom-Up-Prozess zur systematischen Reduzierung der Anforderungen.

3.1 KRITERIEN

Im Folgenden werden die Eigenschaften, nach denen die in dieser Arbeit behandelten Verfahren bewertet werden sollen, noch einmal aufgeführt und begründet, warum gerade diese Eigenschaften ausgewählt wurden.

ROBUSTHEIT: Die Robustheit gewährleistet, dass die Wahl auch unter widrigen Umständen zu einem gültigen Ergebnis führt und ist deshalb Grundvoraussetzung für eine erfolgreiche Wahl.

VERIFIZIERBARKEIT: Individuelle und universelle Verifizierbarkeit führen dazu, dass sich die wahlberechtigten Personen sowohl von der Integrität ihrer eigenen als auch von der Integrität aller Stimmen überzeugen können. Des Weiteren führt die Verifizierbarkeit dazu, dass die Korrektheit des Wahlergebnisses sichergestellt werden kann. Hierdurch wird sowohl die Unmittelbarkeit, die Gleichheit als auch die öffentliche Verifizierbarkeit der Wahl gewährleistet.

WAHLGEHEIMNIS: Dadurch, dass es keinem Dritten ermöglicht wird, die tatsächliche Wahl zu erfahren, wird das Wahlgeheimnis gewahrt und die Wahl findet somit geheim statt.

QUITTUNGSFREIHEIT: Durch die Einhaltung der Quittungsfreiheit können die Wählenden nicht beweisen, welche Wahl sie tatsächlich getroffen haben. Um diese Eigenschaft zu erfüllen, muss das Wahlgeheimnis eingehalten werden. Dies verbessert die Geheimheit in dem Sinne, dass die wählende Person im Nachhinein nicht einmal beweisen könnte, was sie gewählt hat, wenn sie wollte. Dadurch wird der glaubhafte Verkauf der Stimme erschwert und somit die Freiheit der Wahl begünstigt.

NICHT-ERPRESSBARKEIT: Die Nicht-Erpressbarkeit gewährleistet der wählenden Person, dass sie selbst im Falle einer Erpressung frei und geheim wählen kann. Offensichtlich muss dazu die Quittungsfreiheit vorrausgesetzt sein, da dem Erpresser keinerlei Anhaltspunkt auf Plausibilität einer getroffenen Wahl gegeben werden kann.

BENUTZBARKEIT: Die Benutzbarkeit der Wahl ist Voraussetzung, dass die wahlberechtigten Personen das System überhaupt nutzen. Weiterhin ist es wichtig, die Hemmschwelle für die Teilnahme an der Wahl so gering wie möglich zu halten, damit praktisch alle Wahlberechtigten tatsächlich wählen können, was zur Allgemeinheit und Gleichheit der Wahl beiträgt. Des Weiteren muss auch der Prozess für die individuelle Verifizierbarkeit benutzbar sein, was wiederum zur Integrität der Wahl beiträgt.

3.2 BEWERTUNGSSCHEMA

In diesem Kapitel wird für die oben genannten Anforderungen ein Bewertungsschema erarbeitet, mit welchem anschließend die verschiedenen Wahlverfahren bewertet werden. Dieses Bewertungsschema basiert auf dem Framework von Langer et al. [3, 22]. Es wird um die Punkte Robustheit und Benutzbarkeit erweitert. Normalerweise werden diese beiden Kriterien für die Bewertung von theoretischen Protokollen nicht berücksichtigt, allerdings wurde bei der Betrachtung einiger theoretischer Ansätze festgestellt, dass die grundsätzliche Erfüllbarkeit dieser beiden Anforderungen für die praktische Umsetzung unerlässlich sind, egal wie ausgereift das Protokoll hinsichtlich der anderen Kriterien auch ist.

3.2.1 Verifizierbarkeit

Durch die Verifizierbarkeit soll die Eigenschaft der Korrektheit des Ergebnisses sowie die Integrität der abgegebenen Stimmen feststellbar sein. In Abbildung 3 wird schematisch veranschaulicht, wie die Verifizierbarkeit im folgenden Abschnitt unterteilt wird und wie die einzelnen Eigenschaften zur Einhaltung der Korrektheit bzw. der Integrität der Wahl beitragen. Ein System ist Ende-zu-Ende-verifizierbar [23] (manchmal auch nur verifizierbar genannt), wenn es individuell sowie universell verifizierbar ist. In den folgenden Abschnitten werden die unterschiedlichen Level von individueller und universeller Verifizierbarkeit beschrieben.

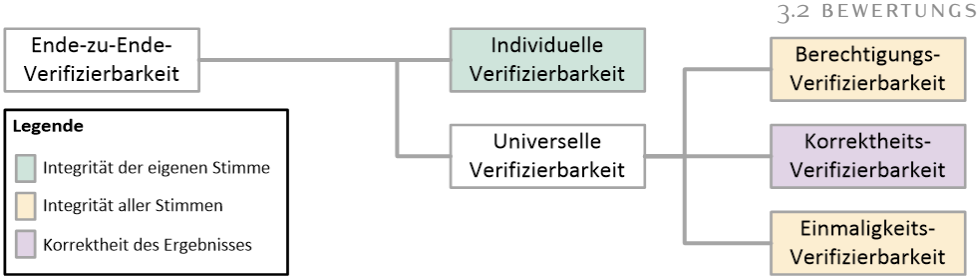


Abbildung 3: Schematische Kategorisierung der Verifizierbarkeit.

Individuelle Verifizierbarkeit

Als Basis dient die Definition, welche bereits in Kapitel 2 abgeleitet wurde. Sie wird zur Erinnerung an dieser Stelle noch einmal wiederholt:

Definition 1. *Ein Wahlsystem ist individuell verifizierbar, wenn der Wähler überprüfen kann, dass seine Stimme mit ins Ergebnis eingeflossen ist.*

Für individuelle Verifizierbarkeit gibt es in der Literatur keine einheitliche Definition. Jedes Wahlprotokoll bringt seine eigene Interpretation mit, je nachdem auf welche Weise die individuelle Verifizierbarkeit erfüllt wird. Langer et al. [3] haben daraus verschiedene Stufen entwickelt, für die jedoch leider noch keine Formalisierung existiert. Sie unterscheiden hierbei zwischen innerer[1] und äußerer[2] individueller Verifizierbarkeit, abhängig davon, welche Informationen der Wähler überprüfen kann.

IV.1 **Innere individuelle Verifizierbarkeit:** Der Wähler kann überprüfen, ob seine Stimme auf dem Bulletin Board veröffentlicht wurde und dass diese Stimme den korrekten Kandidaten enthält.

IV.2 **Äußere individuelle Verifizierbarkeit:** Der Wähler kann überprüfen, ob seine Stimme auf dem Bulletin Board veröffentlicht wurde, er kann allerdings nicht verifizieren, ob die Stimme den korrekten Kandidaten enthält.

Es sollte klar sein, dass die Eigenschaft IV.1 als höherwertig anzusehen ist, als die Eigenschaft IV.2. Erfüllt ein Wahlprotokoll die Eigenschaft IV.1, kann der Wähler sicherstellen, dass seine Stimme nicht gelöscht oder geändert wurde. Das bedeutet, dass die Integrität der Stimme sichergestellt werden kann. Wird von einem Protokoll nur die Eigenschaft IV.2 erfüllt, so kann der Wähler zwar überprüfen, ob seine Stimme auf dem Bulletin Board veröffentlicht wurde, er kann allerdings nicht verifizieren, ob die Stimme geändert wurde. Für beide Eigenschaften existieren weitere Differenzierungen. So kann man zwischen der individuellen Verifizierbarkeit **vor der Auszählung** oder **nach der Auszählung** unterscheiden. Das „x" stellt in diesem Kontext sowohl in diesem als auch in den folgenden Unterabschnitten des Bewertungsschemas einen Platzhalter dar, der beliebig durch eine der im selben Unterabschnitt weiter oben bzw. vorne aufgelisteten Ziffern ersetzt werden kann.

IV.x.1 **Individuelle Verifizierbarkeit nach der Auszählung:** Der Wähler kann überprüfen, ob seine Stimme korrekt in das Ergebnis eingeflossen ist.

IV.x.2 **Individuelle Verifizierbarkeit vor der Auszählung:** Der Wähler kann überprüfen, ob seine Stimme korrekt beim Bulletin Board bzw. Urnenserver abgegeben wurde.

Wie später allerdings noch deutlich werden wird, kann **individuelle Verifizierbarkeit vor der Auszählung** kombiniert mit **universeller Verifizierbarkeit** die Eigenschaft der **individuellen Verifizierbarkeit nach der Auszählung** erfüllen.

Ein weiteres Bewertungskriterium, welches von Riera [27] und Lambrinoudakis et al. [25] erwähnt wird, bewertet, wie ein Einspruch bei der fehlgeschlagenen individuellen Verifizierung erfolgen kann. Diese Einspruchsmöglichkeiten werden für diese Bewertungsmethodik zwar nicht in Betracht gezogen, nichtsdestotrotz werden sie hier kurz benannt. Zum einen gibt es einen mit dem **Wahlgeheimnis konformen beweisbaren Einspruch**, welcher genau dann gegeben ist, wenn der Wähler seinen Einspruch belegen kann, ohne das Wahlgeheimnis zu brechen. Des Weiteren gibt es die Möglichkeit des mit dem **Wahlgeheimnis nicht konformen beweisbaren Einspruchs**. Hierbei muss der Wähler, um den Betrug aufzudecken, beweisen was er gewählt hat, wodurch das Wahlgeheimnis gebrochen wird. Die letzte Möglichkeit ist, dass **keine Möglichkeit zum beweisbaren Einspruch** existiert [3].

1 Entspricht den Definitionen von Delaune, Kremer und Ryan [24] sowie Sampigethaya und Poovendran [10].
2 Entspricht den Definitionen von Lambrinoudakis et al. [25] sowie Smith [26].

Universelle Verifizierbarkeit

Als Basis dient die Definition, welche bereits in Kapitel 2 abgeleitet wurde:

Definition 2. *Jeder kann die Korrektheit des Wahlergebnisses prüfen.*

Für eine formale Überprüfung der universellen Verifizierbarkeit wird auf die Definition von Delaune, Kremer und Ryan [24] verwiesen. Durch die universelle Verifizierbarkeit soll also die Korrektheit der Wahl durch jeden überprüfbar gewährleistet werden. Das heißt unter anderem, jeder muss überprüfen können, dass nur wahlberechtigte Personen gewählt haben. Diese Eigenschaft wird von Sampigethaya und Poovendran [10] als „Berechtigung" und von Langer et al. [3] als „Wahlberechtigungs-Verifizierbarkeit" bezeichnet. Außerdem muss gewährleistet werden, dass jeder berechtigte Wähler tatsächlich nur eine Stimme abgegeben hat, was von Langer [22] als „Einmaligkeits-Verifizierbarkeit" bezeichnet wird. Die korrekte Aufsummierung der abgegebenen Stimmen wird von Langer [22] als „Korrektheits-Verifizierbarkeit" bezeichnet. Alle drei genannten Eigenschaften sind unabhängig voneinander. Somit ergeben sich folgende Abstufungen:

WV.1 **Bedingungslose Wahlberechtigungs-Verifizierbarkeit:** Jeder kann verifizieren (ohne einer in den Prozess involvierten Partei zu vertrauen), dass ausschließlich berechtigte Wähler ihre Stimme abgegeben haben.

WV.2 **Bedingte Wahlberechtigungs-Verifizierbarkeit:** Jeder kann verifizieren, dass ausschließlich berechtigte Wähler Stimmen abgegeben haben. Dazu ist es nötig, dass der Verifizierer bestimmten Parteien vertraut, welche in den Authentifizierungsprozess eingebunden sind.

EV.1 **Bedingungslose Einmaligkeits-Verifizierbarkeit:** Jeder kann verifizieren (ohne einer in den Prozess involvierten Partei zu vertrauen), dass alle Wähler ausschließlich eine Stimme abgegeben haben.

EV.2 **Bedingte Einmaligkeits-Verifizierbarkeit:** Jeder kann verifizieren, dass alle Wähler ausschließlich eine Stimme abgegeben haben. Dazu ist es nötig, dass der Verifizierer bestimmten Parteien vertraut, welche in den Authentifizierungs- und Wahlprozess eingebunden sind.

KV.1 **Kontinuierliche Korrektheits-Verifizierbarkeit:** Jeder kann verifizieren, dass während des kompletten Auszählungsprozesses keine Fehler gemacht wurden.[3]

KV.2 **Diskrete Korrektheits-Verifizierbarkeit:** Jeder kann verifizieren, dass während eines Teils des Auszählungsprozesses keine Fehler gemacht wurden.[4]

Es gilt, dass WV.1, EV.1 und KV.1 die jeweils dazugehörigen Eigenschaften WV.2, EV.2 und KV.2 implizieren. Folglich ist WV.1 höher anzusehen als WV.2, EV.1 höher anzusehen als EV.2 und KV.1 höher anzusehen als KV.2. Es ist zu erwähnen, dass innere individuelle Verifizierbarkeit nach der Auszählung (IV.1.1), durch innere individuelle Verifizierbarkeit vor der Auszählung (IV.1.2) zusammen mit kontinuierlicher Korrektheits-Verifizierbarkeit (KV.1) erreicht werden kann. Dies gilt, da der Wähler in diesem Fall weiß, dass seine Stimme vor der Auszählung enthalten ist und er zusätzlich nachvollziehen kann, dass die Auszählung korrekt abgelaufen ist. So kann der Wähler nachvollziehen, dass seine Stimme in das Ergebnis eingeflossen ist. Das Selbe gilt für äußere individuelle Verifizierbarkeit nach der Auszählung (IV.2.1) [3].

3.2.2 Wahlgeheimnis und Quittungsfreiheit

Betrachtet man ein allgemeines Wahlverfahren, dann ist klar, dass der Stimmzettel die Wähleridentität und die Wahlentscheidung des Wählers verknüpft (siehe dazu Abbildung 4). Folglich gibt es drei Möglichkeiten, um die Verbindung zwischen Wähleridentität und Wahlentscheidung zu trennen [10, 22]:

- **Verbergen der Wähleridentität:** Der Wähler wählt anonym. Folglich wird die Verbindung zwischen Wähleridentität und Stimmzettel zerstört, z. B. durch Einwurf in eine Urne oder durch die Benutzung eines Mixed-Nets.
- **Verbergen der Wahlentscheidung:** Der Wähler gibt eine verschlüsselte Stimme ab. Folglich wird die Verbindung zwischen Wahlentscheidung und Stimmzettel zerstört, z. B. durch ein homomorphes Auszählungsverfahren.
- **Verbergen von Wähleridentität und Wahlentscheidung:** Der Wähler gibt anonym und verschlüsselt seine Stimme ab.

3 Es wird sichergestellt, dass während der gesamten Auszählungsphase keine Stimme geändert, gelöscht oder vervielfältigt wird.
4 Es wird sichergestellt, dass während eines Teils der Auszählungsphase keine Stimme geändert, gelöscht oder vervielfältigt wird.

Abbildung 4: Zusammenhang Wähleridentität und Wahlentscheidung.

Wahlgeheimnis

Das Wahlgeheimnis wurde in Kapitel 2 wie folgt beschrieben:

Definition 3. *Es darf während sowie nach der Wahl keine Zuordnung zwischen der Identität des Wählers und der Wahlentscheidung möglich sein.*

Langer [22] verwendet dazu den Begriff der Unverknüpfbarkeit (Unlinkability). Die Idee hierbei ist es, die Bewertung des Wahlgeheimnisses davon abhängig zu machen, wo die Verbindung getrennt wird. Allerdings hat diese Unterscheidung keine Auswirkung auf die Stärke des Wahlgeheimnisses. Das Wahlgeheimnis wird demnach wie folgt definiert:

UL.1 **Unverknüpfbarkeit zwischen Wähleridentität und Wahlentscheidung:** Es ist (dem Angreifer) nicht möglich, eine Verbindung zwischen Wähleridentität und Wahlentscheidung herzustellen.

UL.2 **Unbeweisbarkeit der Verknüpfung zwischen Wähleridentität und Wahlentscheidung:** Es ist (dem Angreifer) möglich, eine Verbindung zwischen Wähleridentität und Wahlentscheidung herzustellen. Diese Verbindung kann (der Angreifer) gegenüber Dritten jedoch nicht beweisen.

Ein weiteres Bewertungskriterium ist es, wie lange das Wahlgeheimnis eingehalten werden kann. Dies wird immer dann interessant, wenn kryptografische Methoden (z. B. Verschlüsselungsverfahren) zum Einsatz kommen. Der Grund dafür ist, dass manche kryptografischen Verfahren auf Problemen beruhen, die schwer zu berechnen sind. Die tatsächliche Sicherheit dieser kryptografischen Verfahren kann durch schnellere Algorithmen und schnellere bzw. andersartige Hardware (z. B. Quanten-Computer) verringert werden [3, 22]. Hierbei unterscheiden Langer et al. [3] zwischen Langzeit-Unverknüpfbarkeit und Kurzzeit-Unverknüpfbarkeit. Wobei die damit in Verbindung stehenden Zeiträume nicht genauer definiert sind.

In dieser Arbeit wird dieses Problem deshalb in **informationstheoretische Unverknüpfbarkeit (Information-Theoretic Unlinkability)**, **rechnerische Langzeit-Unverknüpfbarkeit** (Computational Long-Term Unlinkability) und **rechnerische Kurzzeit-Unverknüpfbarkeit** (Computational Long-Term Unlinkability) unterteilt. Rechnerische Unverknüpfbarkeit im Allgemeinen bedeutet, dass das Brechen des Wahlgeheimnisses auf einem schwer berechenbaren Problem beruht, weshalb die variable Sicherheit (Langzeit oder Kurzzeit) des Verfahrens im jeweiligen Einzelfall von der Schlüssellänge des jeweils verwendeten Algorithmus abhängig ist und individuell abgewogen werden muss. Informationstheoretische Unverknüpfbarkeit bedeutet, dass selbst ein Angreifer mit unendlichen Ressourcen das Wahlgeheimnis nicht brechen kann, da ein solches Verfahren beweisbar sicher und deshalb erstrebenswert ist [28]. Bei der Bewertung in Kapitel 4 wird dieses Kriterium nicht berücksichtigt, da die gewählte Schlüssellänge letztendlich eine Eigenschaft der tatsächlichen Implementierung ist und die Bewertung dadurch übersichtlicher gehalten werden kann. Allerdings ist es ratsam, sich vor der Einführung eines entsprechenden Wahlverfahrens über diese Aspekte Gedanken zu machen.

Quittungsfreiheit

Die Quittungsfreiheit ist eine stärkere Definition des Wahlgeheimnisses. In Kapitel 2 wurde diese wie folgt beschrieben:

Definition 4. *Ein Wahlverfahren ist unter der Annahme, dass es das Wahlgeheimnis einhält, quittungsfrei wenn es dem Wähler nicht möglich ist, seine Wahlentscheidung gegenüber Dritten zu beweisen.*

Langer [22] beschreibt die Quittungsfreiheit also unter Zuhilfenahme das Wahlgeheimnisses verbunden mit der Fähigkeit des Angreifers, beliebige Informationen vom Wähler zu erhalten. Die letztgenannte Eigenschaft wird im Angreifermodell festgelegt. Diese Arbeit wird die Quittungsfreiheit selbst jedoch unabhängig vom Angreifermodell betrachten, weshalb sie wie folgt kategorisiert wird:

QF.1 **Quittungsfreiheit:** Der Wähler kann gegenüber dem Angreifer (Erpresser) eine falsche Wahlentscheidung vorgeben, ohne dass der Angreifer feststellen kann, ob der Wähler gelogen oder die Wahrheit gesagt.

QF.2 **Unmöglichkeit von Stimmenkauf:** Es ist (dem Angreifer) selbst mit der Hilfe des Wählers nicht möglich, eine verifizierbare Verbindung zwischen Wähleridentität und Wahlentscheidung herzustellen.

Mit verifizierbarer Verbindung ist gemeint, dass der Angreifer die Informationen, welche er vom Wähler erhält, überprüfen kann, sodass er sicher sein kann, dass der Wähler nicht gelogen hat.

Reihenfolge

Offensichtlich gilt QF.1 impliziert QF.2 und UL.1 impliziert UL.2. Für die Bewertung wird die Quittungsfreiheit als stärkere Definition des Wahlgeheimnisses betrachtet. Der Grund dafür ist, dass die Eigenschaft der Quittungsfreiheit das Wahlgeheimniss impliziert, da der Angreifer bei der Quittungsfreiheit die Informationen des Wählers in seinen Angriff einbeziehen kann. Zusammengefasst ergibt sich also folgende Reihenfolge: QF.1 impliziert QF.2, QF.2 impliziert UL.1, und UL.1 impliziert UL.2.

3.2.3 Nicht-Erpressbarkeit

Die Nicht-Erpressbarkeit wurde früher mit der Quittungsfreiheit gleichgesetzt. Juels, Catalano und Jacobsen beschreiben in [29] zum ersten Mal eine Definition von Nicht-Erpressbarkeit, welche sich von der Quittungsfreiheit unterscheidet. Die Idee dabei ist, dass der Angreifer den Wähler erpressen kann, ohne seine Wahl zu lernen. Juels et al. beschreiben dazu die folgenden drei Angriffe, welche von der Quittungsfreiheit nicht erfasst werden:

- **Randomisierungsangriff (Randomisation Attack):** Der Wähler wird dazu gezwungen, seine Wahlentscheidung zufällig zu treffen. So kann der Angreifer dem Wähler zwar nicht vorgeben, was er wählen soll, aber er kann die freie Entscheidung des Wählers zu einer gewissen Wahrscheinlichkeit verhindern.
- **Abwesenheitsangriff (Forced Abstention Attack):** Bei diesem Angriff verbietet der Angreifer dem Wähler die Wahl. Dies ist insbesondere dann möglich, wenn der Angreifer unterscheiden kann, wer gewählt hat und wer nicht. Es wird davon ausgegangen, dass der Wähler während der Wahl und während der Registrierung zumindest für eine ausreichende Dauer unbeobachtet ist, ansonsten könnte der Angreifer die Teilnahme des Wählers an der Wahl verhindern, indem er ihn für die Dauer der Wahlphase einsperrt.
- **Simulationsangriff (Simulation Attack):** Bei diesem Angriff wird der Wähler zur Herausgabe der Authentifizierungsmerkmale gezwungen. Dadurch kann der Angreifer nun anstelle des Wählers eine Stimme abgeben. Ein solcher Angriff ist immer dann problematisch, wenn der Angreifer zwischen echten und unechten Authentifizierungsmerkmalen unterscheiden kann.

Juels et al. [29] formulieren ein formales Sicherheitsmodell, welches allerdings sehr unflexibel ist. Ein flexibleres Sicherheitsmodell wird von Küsters, Truderung und Vogt [30] vorgestellt. Auf Basis dieses Angreifermodells wird im Folgenden die Nicht-Erpressbarkeit definiert. Dazu wird angenommen, dass es für jeden möglichen Angriff eine Gegenstrategie geben muss, sodass der Angreifer nicht unterscheiden kann, ob der Wähler seine Anweisungen befolgt hat oder ob er eine Gegenstrategie ausgeführt hat.

AA.1 Es existiert eine Strategie zur **Abwehr von Abwesenheitsangriffen**, sodass der Angreifer nicht unterscheiden kann, ob der Wähler seinen Anweisungen gefolgt ist.

AA.2 Der **Angreifer kann** anhand der Informationen, die das System bereitstellt, **nicht entscheiden, ob der Wähler gewählt hat**.

RA.1 Es existiert eine Strategie zur **Abwehr von Randomisierungsangriffen**, sodass der Angreifer nicht unterscheiden kann, ob der Wähler seinen Anweisungen gefolgt ist.

SA.1 Es existiert eine Strategie zur **Abwehr von Simulationsangriffen**, sodass der Angreifer nicht unterscheiden kann, ob der Wähler seinen Anweisungen gefolgt ist.

AA.1 impliziert offensichtlich AA.2, da hier unterschieden wird, ob der Angreifer Informationen des Nutzers benötigt, um einen solchen Angriff abwehren zu können. Die restlichen Eigenschaften sind voneinander unabhängig. Ein Wahlsystem erfüllt also nur dann die Nicht-Erpressbarkeit, wenn es die Eigenschaften AA.1, RA.1 und SA.1 erfüllt. Ansonsten erfüllt das System nur die Eigenschaft der „teilweisen Nicht-Erpressbarkeit“.

3.2.4 Robustheit

Bei der Analyse vieler theoretischer Internetwahlprotokolle wurde festgestellt, dass wenig Wert auf die Robustheit der Wahl gelegt wird. Da bei der Analyse der Anforderungen festgestellt wurde, dass die Robustheit ein entscheidendes Kriterium darstellt, wird das Bewertungsframework von Langer et al. [3] um dieses Kriterium erweitert. In Kapitel 2 wurde die Robustheit wie folgt definiert:

Definition 5. *Robustheit bedeutet, dass niemand einen Abbruch der Wahl oder eine Wahlwiederholung provozieren kann.*

Es werden dazu zwei Möglichkeiten betrachtet. Eine Möglichkeit, eine Wahlwiderholung zu provozieren, ist es, die Verfügbarkeit des Internetwahlsystems anzugreifen. Somit kann kein Wähler seine Stimme abgeben, wodurch die Wahl verhindert wird. Gemeint sind hierbei vor allem Denial of Service (DoS) Angriffe. Diese Eigenschaft wird **äußere Robustheit** genannt, womit die Robustheit gegen Angriffe auf die Verfügbarkeit des Wahlsystems mit dem Ziel, die Durchführbarkeit der Wahl zu beeinträchtigen, gemeint ist. Da diese Eigenschaft sehr stark von der tatsächlichen Implementierung und entsprechenden Sicherheitsvorkehrungen der Infrastruktur abhängt, wird sie nicht in die Bewertung miteinbezogen. Ein weiterer Angriffsvektor ist es, ohne Angriff auf die Verfügbarkeit zu versuchen, die Auszählung des Ergebnisses zu verhindern, z. B. wenn ein Wahloffizieller sich weigert, seinen Schlüssel zur Entschlüsselung des Wahlergebnisses bereit zu stellen. Die Fähigkeit, solche Angriffe auf das System zu verhindern, wird **innere Robustheit** genannt.

RI.1 **Innere Robustheit:** Die Robustheit gegen Angriffe, welche die Auszählung verhindern, ohne die Verfügbarkeit des Wahlsystems anzugreifen.

Eine weitere Unterscheidung wäre es, die Robustheit gegenüber des Vortäuschens fehlgeschlagener Verifizierung abzugrenzen. Dieser Aspekt wird allerdings bereits durch die Eigenschaft des **beweisbaren Einspruchs**, welche in Unterabschnitt 3.2.1 erläutert wurde, abgedeckt.

3.2.5 Benutzbarkeit

Für die praktische Umsetzung von Internetwahlverfahren ist auch die Benutzbarkeit ein sehr wichtiges Bewertungskriterium. Deshalb wird das Framework von Langer et al. [3] auch um diesen Punkt erweitert. Die Benutzbarkeit ist zwar ein Implementierungsaspekt, allerdings sind die theoretischen Protokolle aufgrund der Sicherheitsanforderungen teilweise sehr komplex zu benutzen. Somit kann diese Anforderung oft nicht durch die Implementierung ausgeglichen werden.

Die Benutzbarkeit lässt sich nach Neumann und Volkamer [2] in zwei Bereiche unterteilen: **Benutzbarkeit bei der Wahl** und **Benutzbarkeit bei der Verifizierung**. Ein weiterer Aspekt ist die **Benutzbarkeit bei der Authentifizierung**. Dieser wird hier nicht weiter betrachtet, da er in den meisten Fällen nicht direkt mit dem Wahl-, sondern dem eingesetzten Authentifizierungsverfahren zusammenhängt. Unabhängig davon ist zu klären, was „Benutzbarkeit“ überhaupt bedeutet. Olembo und Volkamer [31] benutzen hierzu die Definition des Standards ISO 9241-11 [32]. Diese unterscheidet zwischen Effektivität, Effizienz und Zufriedenheit der Benutzer. Effektivität bezieht sich auf die Korrektheit, mit welcher der Benutzer die ihm gestellten Aufgaben erfüllt. Sie ist also eine Maßzahl für die Fehleranfälligkeit. Die Effizienz bezieht sich auf die Zeit und die Hilfsmittel, welche der Benutzer dafür benötigt. Die Zufriedenheit der Benutzer ist eine subjektive Maßzahl für das „Benutzererlebnis“ (User Experience). Da diese Arbeit Wahlverfahren ausschließlich aus einer literaturbasierten Perspektive analysiert, fließen ausschließlich Effektivität und Effizienz in das Bewertungsmodell ein.

BENUTZBARKEIT BEI DER WAHL Die Effektivität wird von der Eingabekomplexität des Wahlvorgangs abhängig gemacht. Dazu wird die Benutzbarkeit in die folgenden drei Kategorien unterteilt:

BW.1 **Klick-Voting:** Der Wahlvorgang gestaltet sich derart, dass der Wähler sich ausschließlich durch einen Wahlassistenten klicken muss.

BW.2 **Einfache Eingabe:** Der Wähler muss eine Zeichenfolge (z. B. einen Code) in den Wahlclient eingeben.

BW.3 **Selbst zusammengesetzte Eingabe:** Der Wähler muss seine Eingabe selbst zusammensetzen und diese eingeben.

Offensichtlich gilt BW.1 ist am effektivsten und BW.3 ist am wenigsten effektiv. Für die Effizienz wird zwischen einer Wahl mit und ohne Hilfsmittel sowie der einmaligen und mehrfachen Teilnahme am Wahlprotokoll unterschieden:

BW.x.1 **Einmalige Teilnahme ohne Hilfsmittel:** Der Wähler muss nur einmal aktiv am Wahlvorgang teilnehmen und benötigt dazu keine Hilfsmittel.

BW.x.2 **Einmalige Teilnahme mit Hilfsmittel:** Der Wähler muss nur einmal aktiv am Wahlvorgang teilnehmen, benötigt zur Wahl allerdings Hilfsmittel.

BW.x.3 **Mehrfache Teilnahme ohne Hilfsmittel:** Der Wähler muss mehrmals aktiv am Wahlvorgang teilnehmen, benötigt dazu jedoch keine Hilfsmittel.

BW.x.4 **Mehrfache Teilnahme mit Hilfsmittel:** Der Wähler muss mehrmals aktiv am Wahlvorgang teilnehmen und benötigt zur Wahl außerdem Hilfsmittel.

Auch hier sind die Eigenschaften absteigend nach der Effizienz geordnet. Mit der mehrfachen Teilnahme ist gemeint, dass der Wähler oder sein Wahlclient an unterschiedlichen Wahlphasen aktiv teilnehmen muss. Ein Beispiel für ein solches Wahlprotokoll ist das Protokoll von Fujioka, Okamoto und Ohta [33]. Ein Hilfsmittel ist alles, was für die erfolgreiche Durchführung der Wahl benötigt wird und nicht der Wahlclient selbst oder Software auf dem Wahlclient ist. Beispiele für Hilfsmittel sind Code Sheets und Hardware-Tokens, wie sie in einigen Wahlprotokollen vorgesehen sind. Die fehlerfreie Authentifizierung ist für viele Internetwahlverfahren zwar kritisch, wird im Kontext der Benutzbarkeit jedoch explizit nicht betrachtet, da sie bei vielen Verfahren je nach Implementierung modular austauschbar ist. Ausnahmen hiervon bilden lediglich Wahlverfahren mit nahtlos in den Wahlprozess integrierter Authentifizierung, wie das z. B. beim Code Voting der Fall sein kann.

BENUTZBARKEIT BEI DER VERIFIZIERUNG Wie bereits an anderer Stelle erwähnt, lässt sich die Verifizierung in universelle und individuelle Verifizierung unterteilen. Hier wird nur die Benutzbarkeit des individuellen Verifizierungsprozesses behandelt. Das hat den Grund, dass die universelle Verifizierung bei den betrachteten Wahlverfahren ohnehin nur mit Expertenwissen durchgeführt werden kann. Folglich muss dieser Prozess keine speziellen Anforderungen an die Benutzbarkeit erfüllen.

Für die individuelle Verifizierbarkeit (egal ob innere oder äußere) ist mindestens die Überprüfung einer Zeichenkette erforderlich. Das heißt der Wähler muss mindestens zwei Zeichenketten miteinander vergleichen. Aufgrund der ähnlichen Vorgehensweise bei beiden Arten der individuellen Verifizierung, werden Effektivität und Effizienz mit nur einem Kriterium bewertet.

BV.1 **Vergleiche auf Basis von Zeichenketten:** Der Wähler muss zur individuellen Verifizierung vergleichen, ob die Zeichenkette, welche er beim Wahlvorgang erhalten hat, die gleiche ist, wie auf dem Bulletin Board bzw. in den postalisch übersandten Wahlunterlagen.

BV.2 **Überprüfen von Zero-Knowledge-Beweisen:** Der Wähler muss zur individuellen Verifizierung mindestens einen Zero-Knowledge-Beweis überprüfen.

Hierbei ist BV.1 die Eigenschaft, welche gegenüber BV.2 zu bevorzugen ist, da das Vergleichen zweier Strings offensichtlich einfacher durchzuführen ist, als die Überprüfung eines Zero-Knowledge-Beweises.

3.3 REFERENZ-ANGREIFERMODELL

In diesem Kapitel sollen die Möglichkeiten des Angreifers aufgezeigt werden. Dies dient als Basis für das Angreifermodell, das später wiederum als Grundlage zur Bewertung der Wahlverfahren herangezogen wird.

3.3.1 Allgemein angenommener Systemaufbau

Zur näheren Definition des Angreifermodells wird grundsätzlich von dem in Abbildung 5 skizzierten Systemaufbau ausgegangen. Das Angreifermodell dient dazu, die Fähigkeiten des Angreifers zu beschreiben. Hierfür müssen nicht unbedingt alle Systemkomponenten vorhanden sein, z. B. werden für viele Wahlverfahren keine Hilfsmittel benötigt. Allerdings ist davon auszugehen, dass jedes Internetwahlverfahren mindestens einen Wahlcomputer, einen Server sowie ein Bulletin Board umfasst. Im Folgenden werden die einzelnen Komponenten erklärt:

- **Wahlcomputer:** Komponente mit Verbindung zum Server sowie Schnittstelle zu etwaigen weiteren elektronisch angeschlossenen Geräten.
- **Server:** Alle serverseitigen, in die Authentifizierungs- oder Wahlphase involvierten Komponenten des Wahlsystems.
- **Bulletin Board:** Zuständig für die Bereitstellung jeglicher durch das Wahlsystem öffentlich zugänglich gemachten Informationen. Es wird davon ausgegangen, dass auf dem Bulletin Board ausschließlich Informationen veröffentlicht werden können.

- **Hilfsmittel:** Für die Wahl benötigte Gegenstände, die keine elektronische Verbindung zum Wahlcomputer besitzen, z. B. Code Sheets oder Hardwaretoken.
- **Produktion:** Produktions- und Bereitstellungsprozess (inkl. Postweg) der Hilfsmittel und aller involvierten Systeme.
- **Wahloffizielle:** Personen, die Schlüsselmaterial für die Wahl bereitstellen. Können zusätzlich in die Wahlprozesse miteinbezogen sein.

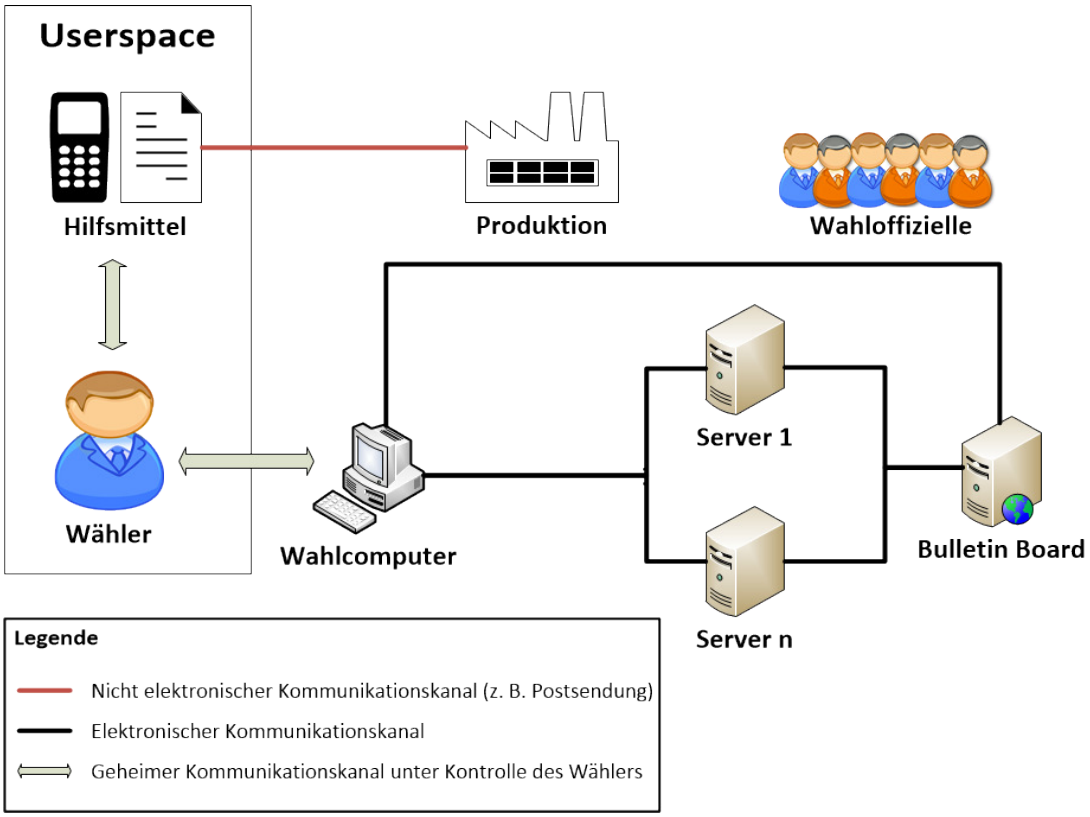


Abbildung 5: Angenommener Systemaufbau als Grundlage des Angreifermodells.

Der Benutzer ist im oben definierten Modell ausdrücklich keine Komponente des Systems, da Angriffe direkt auf den Benutzer durch die Eigenschaften Quittungsfreiheit und nicht-Erpressbarkeit direkt in der Definition des Wahlgeheimnisses enthalten sind. Angriffe auf den Benutzer sind also nicht im Angreifermodell, sondern in der Bewertungsmethodik selbst spezifiziert. Wie bereits in anderen Arbeiten [3, 22] wird auch in der vorliegenden Arbeit angenommen, dass sich der modellierte Angreifer nicht physisch im Userspace befindet und somit zum Zeitpunkt der Wahl keinen direkten Zugriff auf den Wähler hat. Weiterhin können die Kommunikationswege vom Wähler zum Hilfsmittel sowie vom Wähler zum Wahlcomputer nicht vom Angreifer kontrolliert werden.

3.3.2 Angreifer-Fähigkeiten

Folgende Szenarien, aus welchen sich ein individuelles Angreifermodell zusammenstellen lässt, sind denkbar. Es wird hierbei zwischen Angriffen auf die Kommunikation, die Hilfsmittel, die Systemkomponenten und die Wahloffiziellen unterschieden.

Kommunikation

Angriffe auf die Kommunikation sind für den Angreifer teilweise relativ einfach durchzuführen. Er muss dazu nicht Teil des Wahlsystems sein, sondern kann die Nachrichten an einem zentralen Knotenpunkt abfangen. Umso wichtiger ist es deshalb, die Kommunikation durch geeignete Maßnahmen zu schützen.

K.1 Der Angreifer kann die Kommunikationskanäle passiv abhören.[5]

K.2 Der Angreifer kann die Kommunikationskanäle aktiv manipulieren.[6]

K.3 Der Angreifer kann dem Wähler ein bestimmtes Hilfsmittel zuordnen.

Hilfsmittel

Diese Kategorie ist nicht bei allen Wahlsystemen vertreten, weshalb diese Angriffe für viele Wahlsysteme keine Anwendung finden.

H.1 Der Angreifer kann Hilfsmittel außerhalb des Userspaces [7] unverändert kopieren.

H.2 Der Angreifer kann existierende Hilfsmittel außerhalb des Userspaces ohne Vervielfältigung manipulieren.

H.3 Der Angreifer kann eigene Hilfsmittel erstellen / fälschen und dem System zuführen.

Wahlkomponenten

Diese Fähigkeiten sind äußerst kritisch, da sie sich auf das Innere des Wahlverfahrens beziehen. S.2 und S.3 implizieren deshalb die Fähigkeiten K.1 und K.2 von allen angrenzenden Kanälen, wobei die Transportschichtsicherheit in diesem Fall keine Rolle mehr spielt. Für S.1 trifft dies nicht in jedem Fall zu, da das Einfügen von Nachrichten auf dem Bulletin Board abhängig von der eingesetzten Platform z. B. auch per Cross-Site-Scripting möglich sein könnte. Aufgrund des öffentlichen Charakters der auf dem Bulletin Board veröffentlichten Informationen wäre eine Kompromittierung der Vertraulichkeit ohnehin kein Problem. Eine Kompromittierung von Integrität und Authentizität, was das unberechtigte Einfügen von Nachrichten darstellt, allerdings durchaus.

S.1 Der Angreifer kann Nachrichten auf dem Bulletin Board einfügen.

S.2 Der Angreifer kontrolliert den Wahlcomputer.

S.3 Der Angreifer kontrolliert einige, jedoch nicht alle Wahlserver.

S.4 Der Angreifer kontrolliert alle Wahlserver.

Wahloffizielle

Diese Fähigkeit ist ebenfalls sehr kritisch, da der Angreifer dadurch an Schlüsselmaterial gelangen kann.

O.1 Der Angreifer kontrolliert einige, jedoch nicht alle Wahloffizielle.

5 Die Fähigkeit umfasst: Die Benutzung eines Kanals zu erkennen, zu entscheiden wer der Sender einer bestimmten Nachricht ist und die Nachricht abzuhören.

6 Die Fähigkeit umfasst: Blockieren, Einfügen sowie das Modifizieren von Nachrichten.

7 Zwei Fälle sind dabei denkbar: Der Angreifer erlangt Zugriff auf Hilfsmittel während der Produktion oder während des Transports. Aufgrund des gleichen Resultats werden diese beiden Fälle zusammengefasst.

4 AKTUELLER STAND DER FORSCHUNG

Dieses Kapitel gibt eine breite Übersicht aktueller Forschungsprojekte- und Ergebnisse, die sich mit Internetwahlverfahren befassen. Es soll einerseits den Leser in den aktuellen Stand der Forschung einführen und andererseits als Grundlage für die Auswahl der näher zu betrachtenden Lösungen für Internet-Wahlen in den darauf folgenden Kapiteln dienen. Dieses Kapitel ist nach Techniken beziehungsweise kryptografischen Primitiven unterteilt, auf denen die Sicherheit der vorgestellten Verfahren jeweils beruht.

4.1 FUNKTIONSTRENNUNG UND SYSTEMSICHERHEIT

Im Folgenden werden Verfahren beschrieben, deren Sicherheit hauptsächlich auf der Funktionstrennung zwischen verschiedenen Komponenten sowie deren individueller Systemsicherheit beruht.

4.1.1 Estnisches Wahlsystem

Das estnische Wahlsystem wird seit 2001 eingesetzt und war damit das erste Verfahren, welches zur Durchführung von politischen Wahlen eingesetzt wurde. Das Protokoll besteht aus einer Wahlapplikation, welche auf dem Client ausgeführt wird, und dem Internet Voting System (IVS). Die Wahlapplikation verschlüsselt die Stimme mit Hilfe des RSA-Verfahrens und signiert die Stimme mit dem RSA-Schlüssel des estnischen Personalausweises. Anschließend wird die Stimme über eine TLS-Verbindung an das IVS gesendet.

Das IVS besteht aus dem Vote Forwarding Server (VFS), dem Vote Storing Server (VSS) und dem Vote Counting Server (VCS). Zwischen den verschiedenen Komponenten des IVS besteht eine strikte Aufgabentrennung. Der VFS dient als Zugangspunkt zum System, nimmt die Stimme entgegen und leitet diese an den VSS weiter, wo die Stimme bis zum Ende der Wahlphase gespeichert wird. Nach der Wahlphase wird auf dem VSS die Signatur von der verschlüsselten Stimme getrennt und auf den VCS übertragen, wo die Stimme entschlüsselt und ausgezählt wird. Allerdings ist dieses Verfahren aufgrund verschiedener Angriffe bei der estnischen Parlamentswahl 2011 inzwischen so erweitert worden, dass Wählerinnen ihre Stimme zur Verifikation über das Smartphone vom VSS abrufen können.

Die Sicherheit des Wahlsystems basiert auf der Systemsicherheit der einzelnen Komponenten. Dadurch kann das System das Wahlgeheimnis, die Quittungsfreiheit, die Nicht-Erpressbarkeit, Unmöglichkeit von Stimmenkauf sowie das gleiche Stimmgewicht nicht gewährleisten. Die Verifizierbarkeit wird selbst bei der Version ab 2011 nicht eingehalten, da nicht verifiziert werden kann, ob die Stimme tatsächlich ins Wahlergebnisse eingeflossen ist. Das Wahlsystem ist allerdings sehr leicht benutzbar und auch robust gegen Ausfälle, da die Onlinewahl nur eine Möglichkeit ist, seine Stimme abzugeben. Für eine nähere Erklärung und Analyse siehe Abschnitt 5.1.

4.1.2 Polyas

Das Internetwahlsystem *Polyas* wird unter anderem für die Gremienwahlen der Deutschen Forschungsgemeinschaft (DFG), der deutschen Gesellschaft für Informatik (GI) sowie deren US-amerikanischem bzw. internationalem Pendant, der Association for Computing Machinery (ACM), eingesetzt [34]. Polyas ist das erste vom Bundesamt für Sicherheit in der Informationstechnik (BSI) nach Common Criteria Schutzprofil BSI-CC-PP-0037-2008 [35] zertifizierte Internetwahlsystem. Das Wahlsystem basiert aus Sicht des Wahlberechtigten rein auf einer Web-Applikation und ist daher von jedem internetfähigen Betriebssystem mit Browser aus nutzbar. Die Authentifizierung der Wahlberechtigten erfolgt dabei durch eine dem Benutzer auf einem sicheren Kanal (zumeist postalisch) übermittelte Transaktionsnummer (TAN). Das elektronische Wählerverzeichnis, welches auch den für die Wahlberechtigten zugänglichen Webserver bereitstellt, nimmt die Credentials bzw. die TAN des Wahlberechtigten entgegen, verifiziert diese und leitet sie zur erneuten Überprüfung an den Validierungsserver weiter. Bei erfolgreicher Prüfung erstellt der Validierungsserver einen Token, den er an den Benutzer, das elektronische Wählerverzeichnis und den Urnenserver weiterleitet. Der Benutzer authentifiziert sich nun mit dem Token gegenüber

dem Urnenserver und erhält nach erfolgreicher Authentifizierung den elektronischen Stimmzettel angezeigt. Sobald der Stimmzettel ausgefüllt und übermittelt wurde, wird der Token sofort vom Urnenserver gelöscht, sodass retrospektiv keine Möglichkeit besteht, die individuelle Wahl mit dem Benutzer in Verbindung zu bringen [36, 37]. Die Gewährleistung der verschiedenen Sicherheitsanforderungen basiert auf dem Prinzip der Funktionstrennung (Separation of Duties) in Verbindung mit asymmetrischer Kryptografie und der Sicherheit der Systeme, auf denen die verschiedenen Komponenten installiert sind.

4.2 BLINDE SIGNATUREN

Die Idee hinter Protokollen, die blinde Signaturen einsetzen, ist, dass sich die Wählerin beim sogenannten Administrator autorisiert und sich daraufhin geheim auf eine Stimme festlegt. Anschließend überprüft der Administrator, ob die Wählerin berechtigt ist und stellt ihr ein sogenanntes Token aus (wobei dieser Prozess oft durch blinde Signaturen realisiert wird). Nun sendet die Wählerin ihre Klartext-Stimme zusammen mit dem Token über einen anonymen Kanal an den Counter. Der Counter überprüft, ob das Token korrekt ist. Ist dies der Fall, wird die Stimme gespeichert und später ausgezählt. In Abbildung 6 ist der prinzipielle Aufbau eines Wahlsystems auf Basis von blinden Signaturen beschrieben.

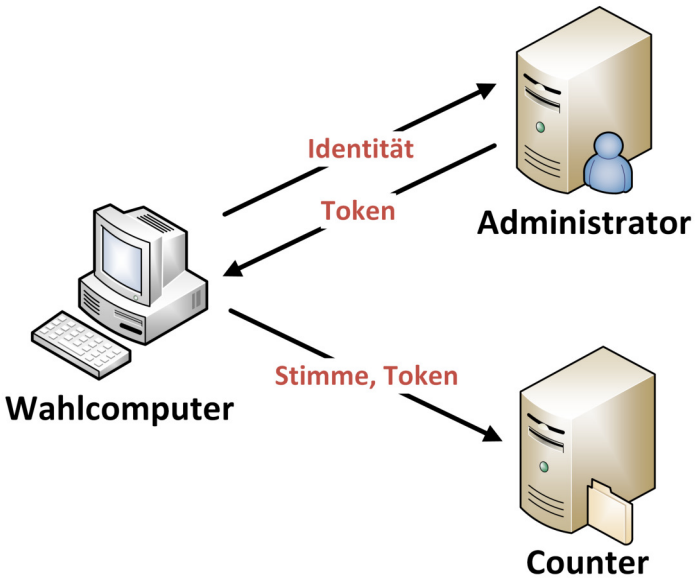


Abbildung 6: Schematischer Aufbau eines Wahlsystems auf Basis blinder Signaturen.

Das erste Wahlprotokoll auf Basis von blinden Signaturen wurde von Chaum [38] 1988 entwickelt. Allerdings ist die Kommunikationskomplexität dieses Protokolls sehr hoch [12]. Das Verfahren wurde ein Jahr später von Boyd [39] sowie Okamoto und Ohta [40], im Bezug auf die Effizienz der Kommunikation in der Vorbereitungsphase verbessert. Allerdings haben alle drei Protokolle den Nachteil, dass nicht garantiert werden kann, dass keine berechtigten Wählerinnen ausgeschlossen werden. Ein solcher Fall tritt dann auf, wenn es mehrere Wählerinnen mit dem selben Token gibt. Des Weiteren sind die Protokolle nicht fair, da der Server Teilergebnisse berechnen kann. Außerdem legen sie keinen Wert auf die universelle Verifizierbarkeit. So muss den Server in Fragen der Integrität komplett vertraut werden.

4.2.1 Protokoll von Fujioka, Okamoto und Ohta

Das Protokoll von Fujioka, Okamoto und Ohta (FOO) [33], ist dass bekannteste Internetwahlprotokoll auf Basis von blinden Signaturen. Die Idee hinter diesem Protokoll ist es, die Authentifizierung der Wählerin und die

Speicherung der Stimmen zu trennen. Dazu nehmen am Protokoll ein Administrator A, ein Counter C und die Wählerin V_i[1] teil. Jede Wählerin V_i besitzt eine ID_i, welche sie eindeutig identifiziert. Der Ablauf des Protokolls stellt sich wie folgt dar:

1. **Vorbereitungsphase:** Die Wählerin füllt auf V_i den Stimmzettel v_i aus, verschlüselt v_i mit dem zufälligen Schlüssel k_i zu x_i und blinded x_i, sodass e_i entsteht. Für die Verschlüsselung wird ein Bit-Commitment BC verwendet. Anschließend signiert die Wählerin e_i. Die Signatur wird mit s_i bezeichnet. V_i sendet das Tupel (ID_i, e_i, s_i) an A.
2. **Administrationsphase:** Der Administrator kontrolliert, ob die Wählerin wahlberechtigt ist und ob sie bereits gewählt hat, indem er die Korrektheit von s_i überprüft. Verlaufen diese Tests positiv, signiert der Administrator e_i blind zu d_i und sendet d_i an V_i. Am Ende der Administrationsphase veröffentlicht A eine Liste, welche (ID_i, e_i, s_i) für alle Wählerinnen enthält.
3. **Wahlphase:** V_i entfernt die Blendung von d_i, sodass V_i eine Signatur y_i von A für x_i erhält. V_i überprüft, ob die Signatur korrekt ist und sendet das Tupel (x_i, y_i) über einen anonymen Kanal (z. B. das Tor-Netzwerk[2]) an C. Anschließend überprüft C, ob die Signatur y_i von A kommt und korrekt ist. Ist dies der Fall, speichert C das Tupel (l, x_i, y_i) in einer Liste, wobei l die Zeilennummer ist. Nachdem alle Stimmen eingegangen sind, veröffentlicht C die Liste.
4. **Auszählungsphase:** Die Wählerin V_i kontrolliert, ob ihre Stimme in der Liste enthalten ist und sendet (l, k_i) an C. C berechnet v_i aus x_i und prüft, ob v_i valide ist. Zum Schluss ermittelt c das Ergebnis der Wahl und veröffentlicht (l, k_i), sodass jeder das Ergebnis prüfen kann.

Die Grundidee von Fujioka, Okamoto und Ohta dient als Ausgangslage vieler weiterer Wahlprotokolle und verschiedener Implementierungen. Diese werden im nachfolgenden Abschnitt beschrieben.

4.2.2 Protokolle auf Basis von FOO

Das Protokoll „Sensus“ von Cranor und Cytron [41] ist eine praktische Implementierung von FOO. Sensus setzt dabei auf das RSA-Signatur- und Verschlüsselungsverfahren. Ein Problem von Sensus ist, dass der anonyme Kommunikationskanal zwischen Counter und Wahlcomputer nicht implementiert wird, sondern auf andere Weise sichergestellt werden muss.

Okamoto [42, 43] erweitert den Ansatz mit dem Ziel, die Quittungsfreiheit zu erfüllen. Die Idee dabei ist, das Bit-Commitment aus dem FOO-Protokoll so zu modifizieren, dass die Wählerin sich (noch) nicht auf eine Wahlmöglichkeit festlegen muss. Dies geschieht durch die Commitment-Funktion $x_i = BC(v_i, k_i) = g^{v_i}(g^{\alpha+k_i})$, wobei $\alpha \in_R \mathbb{Z}_q$. Da der Angreifer α nicht kennt, kann die Wählerin ein Tupel (v_i', k_i') finden, sodass $v_i + \alpha = v_i' + \alpha r_i' \mod q$ ist. Um zu verhindern, dass der Angreifer mit den Informationen der Wählerin die Wahlentscheidung nachvollziehen kann, muss die Verbindung zwischen x_i und v_i gekappt werden. Dies ist der Fall, da der Angreifer mit den öffentlichen Informationen überprüfen kann, ob $m_i' = BC(v_i, k_i)$, wobei $m_i' = BC(v_i', k_i')$. Um die Verifizierbarkeit dennoch zu gewährleisten, setzt Okamoto einen nicht-interaktiven Zero-Knowledge-Proof ein.

Das Protokoll [42] lässt die Wählerin (zufällig) das α bestimmen. Halten sich alle Teilnehmenden an den vorgeschriebenen Ablauf, ist das Protokoll quittungsfrei. Allerdings kann ein Angreifer $G_i = g^{\alpha}$ berechnen und der Wählerin zukommen lassen. Ist das der Fall, so hat die Wählerin kein Tupel (v_i', k_i') da sie α nicht kennt. In seinem anderen Protokoll [43] korrigiert Okamoto seinen Fehler, indem er das α in n Teile aufteilt und diese an ein sogenanntes Parameter Register Committee, welches aus n Personen besteht, sendet. Die Idee dabei ist, dass nur die Wählerin eine Aufteilung von α an das Parameter Register Committee senden kann. Somit kann ein Angreifer (von außerhalb) das α zwar vorgeben, die Aufteilung des αs an die verschiedenen Parameter Register Committee Mitglieder jedoch nicht überprüfen. Allerdings ist diese Annahme unrealistisch, da ein Angreifer, welcher das α vorgeben kann, auch in der Lage ist, die Verbindungen zu den Parameter Register Committee Mitgliedern zu übernehmen.

Außerdem schlägt Okamoto [42] als erster vor, den anonymen Kanal zwischen Wählerin und Counter durch ein Mixed-Net zu realisieren. Des Weiteren sorgt er dafür, dass die Wählerin nur während der Administrationsphase mit dem Wahlsystem kommunizieren muss. Diese Eigenschaft wird im Kontext von Wahlverfahren auf Basis von blinden Signaturen als Walk-Away-Eigenschaft bezeichnet [44]. Allerdings wird durch die Einführung der Walk-Away-Eigenschaft die Fairness beeinträchtigt, da Stimmen bei einem Stimmspeicher S zwischengespeichert werden. Unter der Annahme, dass der Counter und S zusammenarbeiten, geht offensichtlich die Fairness verloren.

1 V_i repräsentiert in diesem Fall die Wählerin inklusive Wahlcomputer.
2 https://www.torproject.org

E-Vox von Herschberg [45] ist wie Sensus eine praktische Implementierungen des FOO Protokolls. Außerdem erfüllt E-Vox die Walk-Away-Eigenschaft. Dazu werden, wie bei den Protokollen von Okamoto [42, 43], die Stimmen auf einem Stimmspeicher, dem im Protokoll sogenannten „Anonymizer“, zwischengespeichert. Allerdings treten bei diesem Verfahren die gleichen Probleme, wie bei den Protokollen von Okamoto auf. Der Anonymizer ist außerdem für die Anonymisierung des Kanals zuständig, indem er verschlüsselte Stimmen entgegennimmt und entschlüsselte Stimmen in zufälliger Reihenfolge zurückliefert. Der Anonymizer kann als ein Ein-Server-Mixed-Netz angesehen werden. Allerdings führt dies dazu, dass bei einer Zusammenarbeit von Counter und Anonymizer das Wahlgeheimnis gebrochen werden kann, da der Anonymizer die Möglichkeit besitzt, die Stimmen zusammen mit den kanalspezifischen Informationen weiterzuleiten. Dadurch ist der Counter in der Lage, die Identität der Wählerin mit ihrer Wahlentscheidung zu verbinden.

Ohkubo et al. [44] verbessert das FOO Protokoll insoweit, dass die Walk-Away-Eigenschaft erfüllt wird, wobei jedoch die Fairness-Eigenschaft vollständig erhalten bleibt. Dies wird dadurch erreicht, dass das in FOO verwendete Verschlüsselungsverfahren durch ein Protokoll mit t aus n Threshold-Encryption ersetzt wird. Dazu werden sogenannte Talliers (Wahloffizielle) eingeführt, welche alle einen Teilschlüssel für die Entschlüsselung der Stimmen bekommen. In der Auszählungsphase müssen t Teilschlüssel für eine Entschlüsselung der Stimmen zur Verfügung stehen. Es wird dabei davon ausgegangen, dass mindestens $t-1$ der Talliers vertrauenswürdig sind, und ihren Teilschlüssel erst in der Auszählungsphase zur Verfügung stellen. Somit ist die Fairness des Protokolls gewährleistet, wenn mindestens $t-1$ der Talliers vertrauenswürdig sind. Des Weiteren beschreiben Ohkubo et al. wie der anonyme Kanal durch Mixed-Nets effizient realisiert werden kann.

DuRette [46] verbessert den Autorisierungsprozess von E-Vox, indem er mehrere Administratoren einführt, welche ein (t, n)-Threshold-Signatur-Protokoll benutzen, um s_i zu erzeugen. Das Problem dabei ist, dass $t \geqslant \frac{n}{2}+1$ sein muss, da sonst eine Wählerin mehrmals wählen kann, weil sie sich von den Administratoren $\{a_1 \dots a_{\frac{n}{2}}\}$ eine Signatur und von $\{a_{\frac{n}{2}+1}, \dots a_n\}$ eine weitere Signatur ausstellen lassen kann. Somit ist die Wahl des richtigen t eine Abwägung zwischen Anzahl der zu vertrauenden Administratoren und der Robustheit. DuRette schlägt deshalb zusätzlich einen Manager vor, von welchem alle Stimmen eine Signatur erhalten, falls sie mindestens t Signaturen von Administratoren bekommen haben und der Manager selbst feststellt, dass die Wählerin noch keine Stimme abgegeben hat. Das von DuRette vorgeschlagene Vorgehen löst somit das Problem, dass ein Administrator das Wahlergebnis verfälschen kann, indem er Wählerinnen mehrere oder nicht-berechtigten Wählerinnen überhaupt Tokens ausstellt. Allerdings wird weder die vollständige Wahlberechtigungs- noch die vollständige Einmaligkeits-Verifizierbarkeit eingehalten, da die Administratoren im Kollektiv den Autorisierungsprozess beeinflussen können, ohne dass dies festgestellt werden kann.

Joaquim [47] verbessert die Skalierbarkeit und die Resistenz gegen Denial-of-Service-Angriffe des von DuRette beschriebenen Protokolls. Dies geschieht, indem von jeder Komponente mehrere Instanzen eingeführt werden. Allerdings führt diese Veränderung auch dazu, dass mehrere der Administratoren einen Denial-of-Service-Angriff durchführen können. Dies wird von Lebre et al. [48] behoben. Liaw [20] erweitert die Idee von Fujioka, Okamoto und Ohta, indem er Smartcards einführt, auf welchen die kryptografischen Operationen ausgeführt werden. Die Sicherheit des Protokolls von Liaw verschlechtert sich im Vergleich zum FOO Protokoll allerdings, was auf einige Fehler im Design zurückzuführen ist. Weitere Protokolle auf Basis von blinden Signaturen sind die Protokolle von Juang und Lei [49], Karro und Wang [50], Dini [51] und Chen et al. [52]. Alle diese Protokolle sind zumeist praktische Implementierungen, welche nur Kleinigkeiten am grundlegenden Konzept verbessern.

Tabelle 2: Bewertung der Wahlverfahren basierend auf blinden Signaturen unter Annahme des Angreifermodells für Wahlen erster Ordnung.

	Verifizierbarkeit		**Wahl-geheim-nis**	**Nicht-Erpress-barkeit**	**Robust-heit**	**Benutz-barkeit**	
	ind.	**uni.**				**Wahl**	**Verif.**
Fujioka et al., Sensus 1992-99	IV.1.1	KV.1	x	x	RI.1	BW.1.3	BV.1
Okamoto 1997	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.2
Ohkubo 1999	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Herschberg, Durette, Joaquim, Lebre 1997-2004	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Liaw 2004	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1

4.2.3 Probleme

Das große Problem bei allen Protokollen, welche in Unterabschnitt 4.2.1 erörtert wurden, ist, dass das Wahlgeheimnis von der Vertrauenswürdigkeit des Wahlclients abhängt und die Integrität von einem oder mehreren Administratoren. Dies hat zur Folge, dass das System ohne einen Beweis der Vertrauenswürdigkeit der Administratoren nicht universell verifizierbar ist.

Wie man sieht, gab es in diesem Bereich in den letzten Jahren nur wenige Forschungsresultate. Das liegt nach Ansicht der Autoren dieser Arbeit daran, dass der Ansatz, blinde Signaturen als Basis für Internetwahlen zu benutzen, ausgereizt ist und die grundlegenden Probleme der universellen Verifizierbarkeit sowie des Wahlgeheimnisses nicht ohne weiteres gelöst werden können.

Allerdings ist der grundlegende Ansatz immer dann interessant, wenn die Wählerinnenregistrierung ebenfalls über das Internet erfolgen soll. So nutzt Civitas [53] eine ähnliche Architektur der Komponenten, setzt für die Generierung der privaten Credentials allerdings andere kryptografische Verfahren ein.

Wahlen auf Basis von blinden Signaturen sind sehr stark mit den Wahlverfahren auf Basis des verifizierbaren Mischens verwandt, da der anonyme Kanal oft mittels sogenannten Mixed-Nets realisiert wird. Diese Verfahren werden im nächsten Abschnitt beschrieben.

4.3 VERIFIZIERBARES MISCHEN

Verifizierbares Mischen kann man sich als kryptografisches Äquivalent zur Wahlurne vorstellen. Eine verifizierbare Menge an Stimmen geht hinein, wird gemischt und kommt in einer anderen Reihenfolge wieder heraus. Man spricht von einem Mixed-Server, wenn das Mischen von einer Instanz vorgenommen wird. Wird das Mischen hingegen von mehreren Instanzen vorgenommen, spricht man von sogenannten Mixed-Nets. Für eine allgemeine Beschreibung von Mixed-Nets und verifizierbarem Mischen, siehe Unterabschnitt A.1.4. In Internetwahlverfahren kommt verifizierbares Mischen zum einen für die Umsetzung eines anonymen Kanals zwischen Wahlclient und Wahlserver zum Einsatz, wie dies z. B. in Abschnitt 4.2 über blinde Signaturen beschrieben ist. Bei solchen Verfahren sind die Mixed-Nets nur für die Anonymisierung der Wählerinnen zuständig. Die Verifizierbarkeits-Eigenschaften werden auf anderem Wege festgestellt. Das andere Einsatzszenario ist die Anonymisierung der Stimmzettel bei der Stimmauszählung. Offensichtlich kann hierbei keine spätere Verifizierung (z. B. individuell) erfolgen, weshalb die Mixed-Nets, welche hierfür eingesetzt werden, verifizierbar sein müssen.

Verifizierbares Mischen wurde zuerst von Chaum [54] 1981 eingeführt, welcher dieses Verfahren auch gleich als Teil eines Wahlverfahrens vorschlägt. Dabei kommen Mixed-Nets bei der Anonymisierung des Kanals zum Einsatz. Seien $M_1 \dots M_k$ die Mixed-Server des Mixed-Nets, wobei jeder Server M_j den öffentlichen Schlüssel p_j und den geheimen Schlüssel s_j besitzt. Des Weiteren hat jede Wählerin V_i die Möglichkeit, mit der Funktion sign_i eine Signatur zu erstellen, welche öffentlich verifizierbar ist. Das Wahlprotokoll gestaltet sich folgendermaßen:

1. **Registrierungsphase:** Jede Wählerin V_i erstellt ein Schlüsselpaar (pk_i, sk_i) und berechnet

$$m_{i,1} = \text{enc}(p_1, \text{enc}(p_2, \text{enc}(\dots \text{enc}(p_k, pk_i) \dots))).$$

   Anschließend sendet sie $\text{sign}_i(m_i) \| m_i$ an das Bulletin Board BB. Durch die Signatur kann nun jeder prüfen, ob V_i zur Wahl berechtigt ist. Danach wird die Registrierung geschlossen. Das Mixed-Net berechnet aus allen $m_{i,1}$ eine Liste P, welche alle pk_i's enthält. Dazu entschlüsselt M_k die Nachricht und schickt diese in lexikografischer Reihenfolge weiter an M_{k-1} solange bis M_1 eine Liste mit den pk_i's erhält, welche M_1 auf dem BB veröffentlicht. Anschließend überprüft V_i, ob ihr pk_i in P enthalten ist. Falls nicht, reicht die Wählerin Beschwerde ein und die Wahl wird gestoppt.

2. **Wahlphase:** V_i trifft seine Wahl v_i und berechnet $s_i = pk_i \| enc(v_i \| 0^l, sk_i)$. Anschließend bereitet er m_i wie in der Registrierungsphase für das Mixed-Netz vor, wobei er die Nachricht $m_{i,2}$ erhält. Anschließend veröffentlicht V_i die Nachricht $sign_i(m_{i,2}) \| m_{i,2}$ auf dem BB. Sind alle Stimmen abgegeben, wird $m_{i,2}$ durch das Mixed-Net anonymisiert und m_i auf dem Bulletin Board veröffentlicht.

3. **Auszählungsphase:** Nun kann jeder prüfen, ob für die abgegebenen Stimmen $pk_i \in P$ und $dec(s_i, pk_i) = v_i | = 0^l$ ist. Außerdem kann jeder das Ergebnis ermitteln.

Ein Problem dieses Wahlverfahrens ist, dass die Wählerin mehrmals am Protokoll teilnehmen muss. Ein weiteres Problem ist die Länge der Nachricht, welche durch das Mixed-Net linear zur Anzahl der Mixed-Server wächst. Das liegt daran, dass das Mix-Verfahren von Chaum [54] die Nachricht für jeden Mix-Server einzeln verschlüsselt.

Park, Itoh und Kurosawa [55] lösen 1993 das Problem der langen Nachricht, indem sie auf Re-Randomisierung des Chiffrats setzen. Dazu ersetzen sie das von Chaum gewählte RSA-Kryptosystem durch das ElGamal-Kryptosystem.

Sako und Kilian [56] kombinieren 1995 die Idee von Park et al. [55] mit der Idee von Benaloh und Tuinstra [57] und erreichen dadurch ein quittungsfreies Wahlprotokoll. Außerdem kann jeder verifizieren, dass die Stimme genau so auf dem Bulletin Board eingegangen ist, wie von der Wählerin erwünscht, vorausgesetzt der Wahlcomputer ist vertrauenswürdig. Dies wird durch verifizierbares Mischen erreicht. Ein Nachteil des Protokolls ist, dass ausschließlich zwischen „Ja“ und „Nein“ entschieden werden kann. Um die Quittungsfreiheit einzuhalten, kommt ein sogenanntes Camelion-Bit-Commitment [58] zum Einsatz.

Weitere Verbesserungen in diesem Bereich betreffen vor allem die Mixed-Nets. So verbessern Ogata et al. [59], Abe [60, 61], Jakobsson [62], Neff [63] sowie Furukawa und Sako [64] allesamt die Eigenschaften von Mixed-Nets im Bezug auf Verifizierbarkeit, Effizienz oder Praktikabilität. Auf diese Details wird hier allerdings nicht eingegangen, da sich dadurch die grundlegenden Ideen nicht ändern.

4.3.1 Protokoll von Jules, Catalano und Jakobsson

JCJ von Jules, Catalano und Jakobsson [29, 65] ist das erste Protokoll, welches unter der Annahme, dass der Wahlcomputer vertrauenswürdig ist, die Eigenschaft der Nicht-Erpressbarkeit erfüllt. JCJ ist deshalb die Basis für viele weitere Protokolle. Die Idee von JCJ ist es, dass die Identität der Wählerin während des gesamten Wahlprozesses geheim bleibt. Bei der Stimmabgabe wird die Stimme verschlüsselt und zusammen mit einem geheimen Berechtigungsnachweis abgegeben. Dieser Berechtigungsnachweis kann als eine Art Anonymous Credential [66] betrachtet werden. Um sicherzustellen, dass nur berechtigte Wählerinnen ihre Stimme abgeben, werden die eingereichten Berechtigungsnachweise bei der Auszählung mit einer Liste, welche bei der Registrierung erstellt wurde, blind[3] verglichen. Durch verifizierbares Mischen und den erwähnten blinden Vergleich kann sichergestellt werden, dass bei der Verifikation keine Verbindung zwischen Wählerin und dem Berechtigungsnachweis hergestellt werden kann. Um die Nicht-Erpressbarkeit zu gewährleisten, hat die Wählerin die Möglichkeit, Fake-Berechtigungsnachweise zu erstellen, welche der Angreifer nicht von richtigen Berechtigungsnachweisen unterscheiden kann.

Im folgenden Abschnitt wird das Protokoll erklärt. Um dies zu vereinfachen, werden als erstes die Protokoll-Teilnehmer erklärt.

- **Registrar:** Der Registrar ist für die Registrierung der Wählerinnen zuständig. Er besitzt ein Schlüsselpaar (sk_R, pk_R).
- **Tailies:** Die Tailies $T = \{T_1 \dots T_n\}$ sind für die Auszählung zuständig. Sie teilen sich ein Schlüsselpaar (pk_T, sk_T) für ein Threshold-Encryption-Protokoll.
- **Mixed Server:** $M_1 \dots M_d$ sie bilden das Re-Encryption Mixed-Net MN.

3 „Blind“ bedeutet in diesem Kontext, dass die auszählende Instanz den Berechtigungsnachweis nicht lernt.

- **Bulletin Board:** BB
- **Wahlcomputer:** P

Als kryptografisches Bauelement wird eine modifizierte Variante von ElGamal (M-ElGamal) eingesetzt, welche als vereinfachte Version des Cramer-Soup Kryptosystems [67] angesehen werden kann. Im Folgenden bezeichnet $\mathbb{Z}_p$ die zyklische Gruppe, auf welcher das M-ElGamal-Kryptosystem arbeitet. $\text{enc}(k, m)$ bezeichnet die Verschlüsselung der Nachricht m durch M-ElGamal und den Schlüssel k. Das Schlüsselpaar (pk_T, sk_T) besteht aus den Schlüsseln für das M-ElGamal-Kryptosystem. Wie bereits erwähnt, ist sk_T zwischen T verteilt. Die genaue Funktionsweise des M-ElGamal-Kryptosystems mit Threshold-Encryption kann in [65] nachgelesen werden. Neben M-ElGamal kommt ein Plaintext-Equivalence-Test (PET) zum Einsatz, welcher zwei M-ElGamal Chiffrate erhält und eine Ja/Nein Entscheidung trifft, ob beide Chiffrate den selben Klartext enthalten. Eine effiziente Methode, diesen PET durchzuführen wird in [68] beschrieben. Der Protokoll-Ablauf ist wie folgt:

1. **Setup-Phase:** R und T generieren ihre Schlüssel und veröffentlichen pk_T und pk_R.
2. **Registrationsphase:** Die Wählerin V_i autorisiert sich gegenüber R. Anschließend sendet V_i ein $\sigma_i \in_R \mathbb{Z}_p$ an R. R berechnet $s_i = \text{enc}(pk_T, \sigma_i)$, fügt diesen zu einer Liste L hinzu und sendet s_i an V_i. Zu einem gewissen Zeitpunkt wird die Wahl geschlossen und L zusammen mit einer Signatur von R auf dem BB veröffentlicht.
3. **Wahlphase:** Nun wird die Kandidatenliste C sowie ein zufälliges ϵ (z. B. von R) veröffentlicht. Die Kandidatenliste ordnet jedem Kandidatennamen n_i ein eindeutiges $c_i \in \mathbb{Z}_p$ zu. C ist dabei signiert, sodass es nicht unerlaubt geändert werden kann. V_i gibt seine Stimme für Kandidat $c_j \in C$ ab, indem er das Tupel $(E_{1,i}, E_{2,i})$ berechnet, wobei $E_{1,i} = \text{enc}(pk_T, c_j)$ die Stimme und $E_{1,i} = \text{enc}(pk_T, \sigma_i)$ das Credential von V_i ist. Außerdem wird ein nicht-interaktiver ZKP $P_{c,i}$ beigefügt, welcher beweist, dass $c_j \in C$[4] sowie ein nicht-interaktiver ZKP $P_{\sigma,i}$ welcher beweist, dass V_i σ kennt. Als Challenge für die ZKP's wird $E_{1,i}, E_{2,i}, \epsilon$ verwendet. V_i sendet den Stimmzettel $B_i = (E_{1,i}, E_{2,i}, P_{c,i}, P_{\sigma,i})$ durch einen anonymen Kanal[5] an BB, wo B_i veröffentlicht wird. Sei BL eine Liste aller abgegebenen Stimmen. Die Stimme B_i ist keine Quittung, da der Angreifer nicht weiß, wem B_i gehört und V_i für sein B_i keine Schlüssel bereitstellen kann.
4. **Auszählungsphase:** Als erstes verifizieren die T's für jedes $B_i \in BL$ auf dem BB die Beweise $P_{c,i}, P_{\sigma,i}$. Sei BL_1 die Liste, welche die Überprüfung übersteht. Anschließend wird mittels PET jedes $E_{2,i} \in BL_1$ mit jedem $E_{2,j} \in BL_1$ mit $j \neq i$ verglichen. So werden die Duplikate nach vorgegebener Richtlinie entfernt. Die übrigen Stimmen sowie die Liste L werden durch MN gemischt. Sei L' und BL_2' das Resultat, dann wird mittels PET jedes $E_{2,i} \in BL_2'$ mit jedem $s_j \in L'$ verglichen und alle Stimmen in BL_2', für die kein Äquivalent in L' existiert, werden verworfen. Die übrigen Stimmen können nun von den T's entschlüsselt und ausgezählt werden.

Die Idee ist nun, dass die Wählerin mehrfach ihre Stimme abgeben kann und bei einer Erpressung anstatt des Credentials σ ein Fake-Credential σ' herausgeben kann. Durch die Anonymisierung zusammen mit der Verwendung von M-ElGamal kann ein Angreifer nicht feststellen, ob eine Wählerin gewählt hat oder nicht. Dieses Vorgehen verhindert Abwesenheits-und Randomisierungsangriffe.

Offensichtlich wird bei JCJ die Verifizierbarkeit nicht betrachtet, was einen entscheidenden Nachteil darstellt. Außerdem wird die Registrierung von nur einem Registrar durchgeführt, was eine Manipulation des Registrierungsprozesses ermöglicht.

4.3.2 Protokolle auf Basis von JCJ

Das Protokoll Civitas [53] von Clarkson, Chong und Myers basiert auf JCJ und ist die erste praktische Implementierung, welche Nicht-Erpressbarkeit gewährleistet. Civitas führt einen verteilten Registrierungsprozess ein, sodass die korrekte Autorisierung der Wählerin nicht alleine von einem Server

4 Dies ist nötig, damit die Wählerin eine ungültige Wahl nicht als Quittung benutzen kann.
5 Dieser kann durch ein Mixed-Net realisiert werden.

abhängig ist. Dies wird erreicht, indem sich die Wählerin V_i bei allen Registraren $R_1 \dots R_k$ authentifiziert und jeder R_j jeweils ein zufälliges Teilgeheimnis $\text{enc}(pk_T, \sigma_{i,j})$ erstellt und an die Wählerin sendet (zusammen mit einem ZKP für die Korrektheit). Die Wählerin kann nun durch homomorphe Multiplikation das Credential σ_i erstellen. Eine weitere Verbesserung ist, dass die Korrektheits-Verifizierbarkeit gegeben ist, was bei JCJ ausschließlich angedeutet wird. Außerdem erlaubt Civitas auch Rang-Wahlen. Civitas ist Ausgangslage für viele weitere Verbesserungen.

Ein Problem von JCJ und Civitas ist die quadratische Laufzeit bei der Auszählung. Smith [26] sowie Weber, Araújo und Buchmann [69] verbessern JCJ, sodass die Auszählung mit linearer Laufzeit ausgeführt werden kann. Dabei verletzt das Protokoll allerdings die Quittungsfreiheit sowie folglich auch die Nicht-Erpressbarkeit [69]. Araujo [70, 71] gelingt es mittels Gruppensignaturen, die Auszählung von JCJ in linearer Laufzeit durchzuführen. Allerdings kann dabei nicht ganz ausgeschlossen werden, dass auch nicht vom Registrar autorisierte Wählerinnen eine Stimme abgeben können. Spycher et al. [72] benutzt das Vorgehen von Smith [26] und Weber et al. [69] für das Entfernen von Duplikaten. Es gelingt, die Entfernung von nicht-berechtigten Stimmen (d. h. der Vergleich $E_{2,i} \in BL'_2$ mit jedem $s_j \in L'$) in linearer Zeit durchzuführen [73]. Schlaepfer et al. [73] beschreiben ein Protokoll, welches auf Civitas basiert und die Laufzeit auf Client-Seite verbessert, sodass das Protokoll auf Wahlclients mit wenig Rechenleistung, wie z. B. Smartcards, ausgeführt werden kann.

Shirazi et al. [74] erweitert Civitas insofern, dass die Robustheit verbessert wird. Dabei weisen Shirazi et al. darauf hin, dass Civitas für die Ermitlung des Ergebnisses keine Robustheit gewährleistet, da keine Threshold-Encryption zum Einsatz kommt. Dies lässt sich jedoch einfach beheben. Des Weiteren wird die Robustheit bei der Autorisierung bemängelt. Shirazi et al. beschreiben einen Angriff, bei dem ein Registration-Teller andere Werte $\sigma_{i,j}$ an die Wählerin sendet, als den Wert, welchen er in die Liste L hinzugefügt. Dadurch werden bei der Auszählung alle diese Stimmen aussortiert. Eine Lösung ist es, Threshold-Encryption einzusetzen, was allerdings dazu führt, dass es für eine korrekte Authentifizierung nicht ausreicht, einem Registration-Teller zu vertrauen (vgl. Protokoll von DuRette in Unterabschnitt 4.2.2). Die Idee dabei ist, dass sich die Wählerin die Registrare selbst aussuchen kann.

Bursuc, Grewal und Ryan entwickeln Trivitas [75], welches auf JCJ und Civitas basiert. Trivitas vereinfacht und verbessert die Verifizierbarkeit. Dabei kommen zwei grundlegende Neuerungen ins Spiel. Zum einen ist das die Idee, dass die Wählerin eine „Versuchsstimme“ abgeben kann, wie dies von Helios [76] bekannt ist. Der Unterschied ist allerdings, dass die Stimme den kompletten Wahlprozess durchläuft. Dazu kann die Wählerin „Versuchscredentials“ anfordern und mit diesen eine Versuchsstimme abgeben. Die Versuchsstimmen werden zusammen mit den Versuchscredentials nach der Wahl veröffentlicht, sodass jeder anhand der Versuchsstimmen jede einzelne Phase der Wahl überprüfen kann. Dies kann nur funktionieren, wenn mögliche korrupte Teilnehmende die Versuchsstimmen nicht von den „echten“ Stimmen unterscheiden können. Um dies zu gewährleisten, werden die Versuchsstimmen mit einem zwischen den Trusties verteilten Schlüssel verschlüsselt. Die Entschlüsselung der Versuchsstimmen erfolgt durch ein Decryption-Mixed-Net. Eine weitere Neuerung von Trivitas basiert auf der Feststellung, dass niemand außer der Wählerin das echte Credential von einem Fake-Credential unterscheiden kann. Folglich kann nach der Anonymisierung die Stimme zusammen mit dem Credential veröffentlicht werden. Dadurch kann jede Wählerin überprüfen, ob ihre Stimme auf dem Bulletin Board enthalten ist. Dies führt zu keiner Beeinflussung der Nicht-Erpressbarkeit, da der Angreifer nicht weiß, zu wem das Credential gehört. Zum anderen kann jedes Fake-Credential genauso überprüft werden wie ein echtes Credential, da die Fake-Credentials (mit den jeweiligen Stimmen) erst im nächsten Schritt universell verifizierbar entfernt werden. Durch diesen Ansatz kann die innere individuelle Verifizierbarkeit nach der Wahl sichergestellt werden.

Selections [77] von Clark und Hengartner auf Basis von JCJ erfüllt die Nicht-Erpressbarkeit selbst bei Shoulder-Surfing[6]. Allerdings ist dazu eine In-Person-Registrierung[7] notwendig. Weitere Verbesserungen wurden hinsichtlich der praktischen Implementierung erzielt. So konnten Neumann und Volkamer [78] sowie Neumann et al. [79] Civitas durch Smartcards erweitern.

6 Szenario, bei dem der Angreifer die Wählerin bei der Wahl beobachtet.
7 Die Wählerin muss sich persönlich bei der Wahlbehörde registrieren.

Tabelle 3: Bewertung der Wahlverfahren basierend auf verifizierbarem Mischen unter Annahme des Angreifermodells für Wahlen erster Ordnung.

	Verifizierbarkeit		Wahl-geheim-nis	Nicht-Erpress-barkeit	Robust-heit	Benutz-barkeit	
	ind.	uni.				Wahl	Verif.
Chaum 1981	IV.1.1	WV.1, KV.1	x	x	x	BW.1.3	BV.1
Park et al. 1993	IV.1.1	WV.1, KV.1	x	x	x	BW.1.1	BV.1
SK 1994	IV.1.1	KV.1	x	x	x	BW.1.1	BV.2
JCJ 2005	IV.2.2	x	x	x	RI.1	BW.1.1	BV.1
Civitas 2008	IV.2.1	EV.1, KV.1	x	x	x	BW.1.1	BV.2
Shirazi 2011	IV.2.1	EV.1, KV.1	x	x	RI.1	BW.1.1	BV.2
Trivitas 2012	IV.1.1	WV.2, EV.1, KV.1	x	x	x	BW.1.1	BV.2

4.4 HOMOMORPHE VERSCHLÜSSELUNG

Einige der Sicherheitsanforderungen werden bei den im Folgenden vorgestellten Verfahren mit Hilfe der Homomorphieeigenschaft der eingesetzten Verschlüsselungsverfahren umgesetzt. Homomorphie bedeutet in diesem Kontext, dass aus mathematischen Operationen resultierende Veränderungen des Chiffrats den dazugehörigen Klartext in der selben Art und Weise verändern. Dabei wird zwischen additiver und multiplikativer Homomorphie unterschieden. Verfahren, die sowohl additive als auch multiplikative Homomorphie aufweisen, werden „voll homomorph“ [80] genannt. Eine Verschlüsselungsfunktion $E()$ ist homomorph, wenn es zwei Operationen $\oplus$ und $\odot$ gibt, sodass gilt, dass $E(a) \odot E(b) = E(a \oplus b)$. Es besteht dabei durchaus die Möglichkeit, dass es sich bei $\oplus$ und $\odot$ um dieselbe Funktion handelt. Abbildung 7 veranschaulicht die grundlegende Funktionsweise von homomorpher Verschlüsselung beim Einsatz in Internetwahlverfahren. Zur weiteren, beispielhaften Vertiefung seien hier die homomorphen Eigenschaften des in Unterabschnitt A.1.3 erläuterten ElGamal-bzw. exponentiellen ElGamal-Verfahrens genannt.

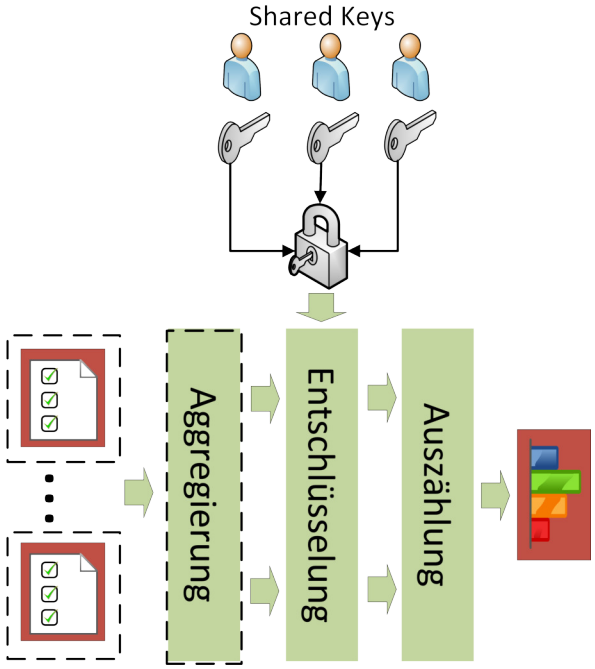


Abbildung 7: Schematische Darstellung der Funktionsweise von homomorpher Verschlüsselung beim Einsatz in Internetwahlverfahren.

4.4.1 Protokolle auf Basis des n-ten Restklassen-Problems (Cohen / Benaloh)

Vorreiter beim Einsatz homomorpher Verschlüsselungsalgorithmen zur Umsetzung sicherer Wahlprotokolle waren Josh Cohen Benaloh (aufgrund seiner variierenden Namensverwendung im Weiteren entweder Benaloh oder Cohen genannt, je nach Verwendung im jeweiligen Paper) und Michael Fischer mit ihrer Arbeit „A Robust and Verifiable Cryptographically Secure Election Scheme“ [81]. Darin beschreiben sie ein Protokoll, das zwei Arten von Akteuren kennt, die der Wählerin (*Voters*) und die der Regierung (*Government*). Weiterhin spezifiziert das Protokoll, dass es einen öffentlich zugänglichen, vertrauenswürdigen Zeitgeber sowie ein vertrauenswürdiges *Beacon* [82] gibt, welches eine öffentlich zugängliche Quelle zufälliger Bits darstellt, dass jede Nachricht mit einem verifizierbaren Zeitstempel sowie Informationen über den jeweiligen Urheber versehen ist und dass jegliche Kommunikation öffentlich über Bulletin Boards stattfindet. Diese Öffentlichkeit ist es, die die Verifizierbarkeit gewährleistet, indem der öffentlich zugängliche, lediglich aus zwei verschlüsselten[8] und zufällig angeordneten Werten (je einmal 0 und 1) bestehende Stimmzettel jeder Wählerin, der nachweislich vor Ende der Wahlperiode auf dem Bulletin Board gepostet wurde, sowohl von der Regierung zur Auszählung als auch von einer *check* genannten Funktion zur Überprüfung des Auszählungsergebnisses der Regierung genutzt werden kann. Die Funktion *check* kann von jeder beliebigen Person ausgeführt werden, das Wahlgeheimnis bleibt dabei erhalten. Im Allgemeinen beruht die Einhaltung des Wahlgeheimnisses bei diesem Protokoll auf dem n-ten Restklassenproblem. Allerdings machen Cohen und Fischer ausdrücklich deutlich, dass ihr Protokoll das Wahlgeheimnis der einzelnen Wählerinnen nicht gegenüber der Regierung schützt, womit die in Kapitel 2 definierte Anforderung des Wahlgeheimnisses nicht erfüllt wird. Viele zeitlich später formulierten Wahlprotokolle basieren auf dem von Cohen und Fischer vorgestellten Modell, so z. B. auch alle weiteren in Unterabschnitt 4.4.1 vorgestellten Arbeiten, die dieses Protokoll sukzessive um weitere Eigenschaften erweitern.

Benaloh und Yung [83] erweitern das von Cohen und Fischer vorgeschlagene Modell, indem sie die Funktionen, die die Regierung in Cohen und Fischers bisherigem Protokoll in sich vereint, auf mehrere Auszähl-Komponenten (sogenannte *Tellers*) verteilt, sodass das Wahlgeheimnis gegenüber der Regierung sogar erhalten bleibt, wenn alle bis auf eine Komponente konspirieren. Allerdings gewährleistet das Protokoll keine Robustheit, das heißt, sobald eine Komponente vom Protokoll abweicht oder sich weigert, es auszuführen, muss die Wahl wiederholt werden, da es keine Auszählung geben kann. Während das ursprüngliche Modell lediglich binäre Abstimmungen zulässt, zeigen Benaloh und Yung nun auch Wege auf, wie Wahlen mit beliebig vielen Kandidaten bzw. Optionen realisiert werden können.

Cohen stellt eine Erweiterung [84] vor, die die soeben beschriebenen Verfahren [81, 83] durch die Einbettung in ein *Secret Sharing Verfahren* robuster gegenüber korrupten Auszähl-Komponenten (*Tellers*) macht. Konkret verwendet er dafür das von Shamir entwickelte *Threshold Scheme* [85]. Außerdem erlaubt es seine Erweiterung der Regierung bzw. deren einzelnen Komponenten, die Siegerin einer Wahl beweisbar zu veröffentlichen, ohne dabei das eigentliche Wahlergebnis offenzulegen. Dies geschieht unter Anwendung eines umgekehrten Ausschlussverfahrens. So wird für jeden Kandidaten jeweils bewiesen (bzw. im Falle des Gewinners lediglich versucht zu beweisen), dass die Annahme, dass er die Mehrheit der Stimmen auf sich vereinen konnte, falsch ist.

Zusammen mit Tuinstra [57] veränderte Benaloh das bisherige Modell dahingehend, dass es der Wählerin unmöglich ist nachzuweisen, welche Wahl sie getroffen hat. Hierfür wird der Wählerin das bereits oben genannte Paar einer jeweils verschlüsselten und zufällig angeordneten 0 sowie einer 1 präsentiert und durch einen interaktiven Beweis nachgewiesen, welcher Teil des Stimmzettels welcher ist. Die Wählerin entscheidet daraufhin lediglich, für welchen der beiden Teile sie sich entscheidet. Da niemand außer den vereinten Komponenten der Regierung wissen kann, welcher Teil des per Secret Sharing [85] unter den Komponenten verteilten Stimmzettels welche Entscheidung beinhaltet, kann die Wählerin ihre Wahl auch nicht beweisen. Vorausgesetzt wird jedoch die Annahme, dass sich die Wählerin alleine in einer sicheren Wahlkabine befindet, in der ihr der Stimmzettel sowie der interakti-

8 Zum Schutz des Wahlgeheimnisses gegenüber anderen Wählerinnen.

ve Beweis angezeigt wird. Aufgrund dieser Annahme wird die behauptete Quittungsfreiheit[9] nach der Definition der vorliegenden Arbeit nicht eingehalten, was von Hirt und Sako später auch nachgewiesen wird [86]. Ellard und Alpert [87] modifizieren das von Benaloh und Tuinstra [57] vorgeschlagene Verfahren, indem sie eine vertrauenswürdige und unerpressbare dritte Instanz einführen, welche den Prozess der Stimmzettelabgabe beobachtet und auf dessen korrekte Ausführung achtet. Dadurch ist es nicht mehr nötig, dass sich die wählende Person zu Beginn des Wahlprozesses in einer sicheren Wahlkabine aufhält, was nach Ellards und Alpers Auffassung eine unrealistische Annahme ist, die das ursprüngliche Verfahren in der Realität dadurch schwer implementierbar macht.

Iversen [88] präsentiert ein Protokoll, welches laut eigener Bekundung sehr stark von den Werken Josh Cohen Benalohs inspiriert wurde. Es beruht auf dem ebenfalls von Iversen eingeführten *Novel Probabilistic Additive Privacy Homomorphism* [89] sowie RSA Blind Signatures. Es kennt insgesamt drei Teilnehmergruppen. Dabei handelt es sich um die Wählerinnen, die zu wählenden Kandidaten und die Instanz, die die Wahl abhält (üblicherweise *Regierung* genannt), die selbst aber auch zu der Gruppe der Kandidaten gehören kann. Die Wählerinnen beantragen hier vor der Wahl unter Vorlage eines digitalen Zertifikates ein Berechtigungs-Token, welches von den Kandidaten signiert wird. Außerdem signieren die Kandidaten den Hash eines gemeinsam generierten leeren Stimmzettels, der daraufhin an die Wählerinnen verteilt wird. Die Wählerin markiert daraufhin kryptografisch, welchen Kandidaten sie wählt und sendet ihren verschlüsselten Wahlzettel und das zuvor generierte Berechtigungs-Token an alle Kandidaten. Diese überprüfen die Rechtmäßigkeit des Stimmzettels, also ob die Wählerin lediglich für einen Kandidaten gestimmt hat, ohne dabei das Wahlgeheimnis zu kompromittieren. Ist die Stimmabgabe fehlerfrei und auch das Token valide und bisher noch nicht benutzt worden, wird der Stimmzettel angenommen. Anderenfalls wird die Wählerin von der Wahl ausgeschlossen. Aufgrund der Homomorphie-Eigenschaft muss nicht jeder Kandidat jeden Stimmzettel speichern, sondern kann diese zusammenführen. Am Ende des Wahlzeitraums entschlüsselt jeder Kandidat einen Teil des Stimmzettels, der für sich allein jedoch nichts über die Wahl aussagt. Erst das Zusammenführen aller Teilergebnisse gibt Aufschluss über das Ergebnis der Wahl, ohne dabei irgendwelche Rückschlüsse auf die einzelnen Stimmen der jeweiligen Wählerinnen schließen zu können. Das Protokoll gewährleistet die Einhaltung des Wahlgeheimnisses, wenn mindestens ein Kandidat ehrlich ist. Iversen beschreibt weiterhin, dass die Kandidaten kontrollieren können, ob die Regierung das korrekte Wahlergebnis veröffentlicht hat. Er nennt diese Eigenschaft lediglich Verifizierbarkeit, nach der in Kapitel 3 zu findenden Definition handelt es sich hierbei jedoch lediglich um eine Untermenge von universeller Verifizierbarkeit. Indiviudell ist sie nicht, da die Wählerinnen selbst nicht verifizieren können und universell ist sie nicht, da zwar die Kandidaten, nicht jedoch andere Akteure, wie z. B. Wahlbeobachter die Korrektheit der Wahl überprüfen können.

4.4.2 Verfahren auf Basis des ElGamal-Verschlüsselungsverfahrens

Kiayias und Yung [90] präsentieren ein Protokoll, welches es ermöglicht, vorbestimmte Wahloptionen und eine sogenannte „Write-In Option“[10] im selben Protokoll abzubilden. Ihr Ansatz, den sie „The Vector-Ballot E-Voting Approach“ nennen, basiert auf homomorpher Kryptografie (entweder *ElGamal* oder *Pailliers Probabilistic Public-Key System* - beide Varianten werden behandelt) und Mix-Nets. Jeder Stimmzettel enthält dabei drei Chiffrate, die die Autoren „Vektoren“ nennen. Dabei handelt es sich zuerst um ein Chiffrat, das die Auswahl einer der vorgegebenen Wahloptionen beinhalten könnte. Außerdem ein Chiffrat, das eine Flag beinhaltet, die darüber Auskunft gibt, ob sich eine Wählerin für die Write-In Option entschieden hat. Das dritte Chiffrat kann eben diese Write-In Option enthalten, falls sich die Wählerin dafür entschieden hat. Bei der Auszählung werden zuerst die ersten, also die nicht-Write-In-Vektoren, mit Hilfe der Homomorphie-Eigenschaft addiert. Dann werden die Flags im

9 Eigentlich wird von Benaloh und Tuinstra sogar behauptet, dass Nicht-Erpressbarkeit vorliegt, da die beiden Begriffe Quittungsfreiheit und Nicht-Erpressbarkeit entgegen der von Juels, Catalano und Jacobsen vorgestellten Terminologie [29] gleichgesetzt werden. Allerdings wurde diese Terminologie erst ca. eine Dekade später publiziert.

10 „Write-In Option“ bedeutet, dass die Wählerin sich nicht für eine der vorgegebenen Wahloptionen, sondern für eine weitere, von ihr festgelegte, entscheidet. In der Praxis könnte das z. B. der Name eines Kandidaten sein, den sie für besonders befähigt hält, der jedoch bisher nicht auf der Wahlliste steht.

zweiten Vektor und je nach dessen Wert auch der Vektor ausgewertet, welcher die Write-In-Option enthält.

Kiayias, Korman und Walluck [91] stellen ihr „Adder“ genanntes, praktisch implementiertes Open-Source-System vor, welches unter der GNU General Public License veröffentlicht wurde. Adder besteht aus einem SQL-basierten Bulletin Board und einem Kerberos-ähnlichen „Gatekeeper-Server“. Die Benutzerinteraktion kann wahlweise browserbasiert über ein Java-Applet oder über eine eigenständige Client-Software stattfinden. Mit Ablauf des Wahlzeitraums addiert das Bulletin Board automatisch alle verschlüsselten Stimmen und veröffentlicht die verschlüsselte Summe. Der Schlüssel wurde vor der Wahl auf eine bestimmte Anzahl von Wahlentitäten verteilt. Diese Wahlentitäten nutzen ihren Schlüssel, um ihr Teilergebnis zu entschlüsseln und laden dies wieder auf das Bulletin Board hoch, welches daraus dann das finale Ergebnis berechnet und veröffentlicht. In einer früheren Publikation beschreiben Kiayias und Yung [92] die Eigenschaft des „Selbstauszählens“, welche es jedweder teilnehmenden oder beobachtenden Instanz ermöglicht, unabhängig und transparent das Ergebnis der Wahl zu berechnen. Das dort vorgestellte Protokoll wird in ihrer praktischen Implementierung [91] jedoch nicht berücksichtigt.

Hirt und Sako [86] stellen ein Protokoll vor, welches auf dem *ElGamal-Verschlüsselungsverfahren* und *Designated Verifier Proofs* basiert sowie unter der Voraussetzung korrekt ist, dass eine bestimmte Anzahl von Auszähl-Komponenten t ehrlich ist und das Wahlgeheimnis gewahrt bleibt, solange nicht mindestens t oder mehr Auszähl-Komponenten unerlaubterweise Informationen austauschen. Das Protokoll sieht vor, dass vor Beginn der Wahl ein verifizierbar gültiger Standard-Stimmzettel mit allen möglichen verschlüsselten Wahl-Optionen erstellt und veröffentlicht wird. Nun wird der Inhalt des Stimmzettels von der ersten Komponente gemischt, was bedeutet, dass die einzelnen Wahl-Optionen erneut verschlüsselt sowie ihre Anordnung innerhalb des Stimmzettels zufällig verändert wird. Nun sendet die entsprechende Komponente die Information darüber, wie die Reihenfolge der Optionen geändert wurde sowie einen Beweis, dass es sich tatsächlich noch um einen gültigen, jedoch neu verschlüsselten sowie permutierten Wahlzettel handelt, auf einem sicheren Kanal an die wählende Person. Dieser Vorgang wird wiederholt, bis ihn alle Komponenten durchlaufen haben und die wählende Person alle nötigen Informationen von den Komponenten erhalten hat. Die wählende Person muss nun aus den erhaltenden Informationen lediglich die tatsächliche Reihenfolge der Wahl-Optionen rekonstruieren und die entsprechende verschlüsselte Option veröffentlichen. Ohne die Informationen anderer Komponenten kann keine Komponente feststellen, welche Wahl getroffen wurde. Durch den Einsatz von *Designated Verifier Proofs* beweist die jeweilige Komponente der wählenden Person, dass die übermittelten Information über das Mischen auch tatsächlich stimmen, allerdings wird dieser Beweis so durchgeführt, dass ihn auch die wählende Person selbst erstellen und fälschen könnte, weshalb der Beweis keinerlei Aussagekraft gegenüber Dritten hat und damit Quittungsfreiheit erreicht ist. Um die Stimmen auszuzählen, werden sie nun lediglich addiert und können dank ihrer Homomorphie-Eigenschaft von den beteiligten Komponenten entschlüsselt und ausgezählt werden.

Einen ähnlichen Ansatz verfolgt Schoenmakers [93]. Er stellt ein *Publicly Verifiable Secret Sharing* (PVSS) Protokoll vor und erläutert dessen beispielhaften Einsatz in Wahlverfahren. Hierfür beruft er sich zuerst auf die von Benaloh et al. vorgestellten Protokolle [57, 81, 83, 84], wobei die Wählerinnen jedoch die PVSS-Rolle der Dealer und die Auszähl-Komponenten die PVSS-Rolle der Participants einnehmen. Um zu wählen, veröffentlichen die Wählerinnen das Ergebnis der Secret Sharing Verteilung sowie einen Beweis, dass es sich bei dem per Secret Sharing verschlüsselten Wert tatsächlich um eine korrekte Stimmabgabe handelt. Für die Auszählung können die verschlüsselten Shares dank ihrer Homomorphie-Eigenschaft zuerst aufsummiert werden, woraufhin das Gesamtergebnis durch Ausführung der Rekonstruktions-Funktion aller Auszählungs-Komponenten berechnet werden kann.

4.4.3 Verfahren auf Basis von Pailliers Probabilistic Public-Key System

Ähnlich wie Schoenmakers beschreiben Damgard und Jurik [94] zuerst ein kryptografisches Primitiv, in ihrem Fall eine spezielle Form von *Pailliers Probabilistic Public-Key System* [95] und zeigen anschließend, wie dieses in Wahlverfahren Anwendung finden kann. Hierfür erweitern sie Pailliers System

um einen Gültigkeitsbeweis für von der Wählerin abgegebene bzw. veröffentlichte Stimmzettel sowie eine Threshold-Funktion, die es ermöglicht, dass mindestens eine bestimmte Anzahl von Auszählungs-Komponenten notwendig ist, um die Stimmauszählung der bereits homomorph addierten Stimmen vorzunehmen, sodass verhindert werden kann, dass einzelne oder wenige unehrlich Komponenten in der Lage sind, die jeweiligen Wahlentscheidungen der wählenden Personen herauszufinden. Durch die von Hirt und Sako vorgestellte Art und Weise [86], die verschlüsselten Wahl-Optionen innerhalb des Stimmzettels zu mischen, kann Damgards und Juriks Protokoll sogar Quittungsfreiheit erreichen.

Auch Baudron et al. [96] nutzen *Paillier's Probabilistic Public-Key System*, um inspiriert von Cramers et al. [97, 98] und Damgard [94] anstatt theoretischer Multi-Candidate Crypto-Primitives ein praktisches Multi-Candidate System zu konstruieren, das von den Autoren für Wahlen auf dem nationalen Level vorgesehen ist. Entgegen bisher vorgestellter Systeme, bildet dieses System durch den Einsatz von lokalen, regionalen und nationalen Wahlkomponenten quasi ein föderales System nach. Die Komponenten der jeweils niedrigeren Ebene berichten dabei ihre Teilergebnisse an die jeweils höhere Ebene, wo sie aggregiert und weitergeleitet werden, wie man es z. B. auch von regulären Papierwahlen aus Deutschland kennt.

Acquisti [99] stellt ebenfalls einen Ansatz vor, der auf dem Einsatz von *Paillier's Probabilistic Public-Key System* (zusammen mit Mix-Nets [54]) basiert und der sehr stark dem Verfahren von Juels und Jakobsson [65] ähnelt. Die Durchführung der Wahl obliegt hierbei mehreren unabhängigen Komponenten, welche jeweils für jede wahlberechtigte Person ein Share der Credentials, mit denen sich die jeweilige Person später als wahlberechtigt ausweist, und ein Share für jede mögliche Wahloption erstellen. Diese Shares werden dann verschlüsselt und auf dem Bulletin Board veröffentlicht sowie zusammen mit einem Designated Verifier Proof, der die Äquivalenz des veröffentlichten und des an die Wählerin gesendeten Credential-Shares beweist, sowie einem Zero-Knowledge-Proof, der beweist, dass die Shares der Wahloptionen auf einem gemeinsamen Standard-Wahlzettel beruhen, an die Wählerinnen gesendet. Um zu wählen, multipliziert die Wählerin nun die Credential-Shares der verschiedenen Komponenten und die Shares der Wahloption, für die sie sich entscheidet, und sendet das verschlüsselte Ergebnis an das Bulletin Board. Zur Auszählung mixen die Komponenten alle eingegangenen Chiffrate sowie alle vorher veröffentlichten Credential-Shares. Daraus ergeben sich zwei Listen, eine mit verschlüsselten und zufällig angeordneten Credentials und eine mit verschlüsselten und zufällig angeordneten Stimmzetteln. Mit Hilfe einer Funktion wird herausgefunden, zu welchen Stimmzetteln valide Credentials existieren, ohne dabei eine tatsächliche Zuordnung zu ermöglichen, wodurch das Wahlgeheimnis geschützt werden soll. Alle Stimmzettel mit validen Credentials werden daraufhin addiert und das Ergebnis am Ende von den Wahlkomponenten gemeinsam entschlüsselt.

4.4.4 Andere Protokolle auf Basis des Diskreten Logarithmus-Problems

Das Protokoll von Sako und Kilian [100] reduziert die Rundenkomplexität, also den mit dem Protokoll verbundenen Kommunikationsaufwand, von Benaloh und Yungs Protokoll [83] durch optimierte Zero-Knowledge-Proofs um mindestens das Zwanzigfache, indem es erlaubt, den Großteil des benötigten Informationsaustausches bereits im Vorfeld der eigentlichen Wahl zu berechnen. Während des Wahlzeitraums reicht es dann aus, lediglich ein Bit sowie zusätzliche Informationen zur Authentifizierung zu senden. Anstelle des n-ten Restklassenproblems basiert das Protokoll nun auf dem Diskreten Logarithmus-Problem. Das Helios-Projekt[11] [76] bietet eine praktische Open-Source-Implementierung des Protokolls von Sako und Kilian [100].

Der Kommunikationsaufwand, der von Sako und Kilians Protokoll vor der eigentlichen Wahl stattfindet, ist laut Cramer et al. [97] dennoch enorm hoch. Sie präsentieren deshalb ein ähnliches Protokoll, das eine abgeschwächte und dadurch wesentlich weniger aufwendige Form von Zero-Knowledge-Proofs, die sogenannten *Witness-Indistinguishable-Proofs* [101], anstelle klassischer Zero-Knowledge-Proofs nutzt, um die Gültigkeit der von der Wählerin erstellten Stimmzettel zu beweisen. Dies führt, verglichen mit früheren Protokollen, zu einer linearen anstelle einer quadratischen Komplexität. Trotz-

11 http://heliosvoting.org

dem korreliert die Komplexität des Protokolls weiterhin mit der Anzahl der eingesetzten Auszähl-Komponenten. Ein Jahr später optimieren Cramer et al. [98] ihr Protokoll deshalb noch weiter, indem sie einen Ansatz vorschlagen, dessen Komplexität unabhängig von der Anzahl der eingesetzten Auszähl-Komponenten ist.

Tabelle 4: Bewertung der Wahlverfahren basierend auf homomorpher Kryptografie unter Annahme des Angreifermodells für Wahlen erster Ordnung.

	Verifizierbarkeit		Wahl-geheim-nis	Nicht-Erpress-barkeit	Robust-heit	Benutz-barkeit	
	ind.	uni.				Wahl	Verif.
Cohen 1985	IV.1.1	KV.1	x	x	x	BW.1.3	BV.1
Benaloh, Yung 1986	IV.1.1	KV.1	x	x	x	BW.1.3	BV.1
Cohen 1994	IV.1.1	KV.1	x	x	RI.1	BW.1.3	BV.1
Benaloh, Tuinstra 1994	IV.1.1	KV.1	x	x	RI.1	BW.1.3	BV.1
Ellard, Alpert 2003	IV.1.1	KV.1	x	x	RI.1	BW.1.3	BV.1
Iversen 1991	x	WV.2, EV.2	x	x	RI.1	BW.1.1	x
Kiayias, Yung 2004	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.2
Kiayias et al. 2006	x	KV.1	x	x	x	BW.1.1	x
Hirt 2000	IV.2.1	x	x	x	x	BW.1.1	BV.2
Schoenmakers 1999	IV.1.1	KV.1	x	x	x	BW.1.1	BV.2
Damgard, Jurik 2001	IV.2.1	x	x	x	RI.1	BW.1.1	BV.2
Baudron et al. 2001	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Acquisti 2004	IV.1.1	EV.1, KV.1	x	x	RI.1	BW.1.3	BV.1
Sako, Kilian 1994	IV.2.1	KV.1	x	x	x	BW.1.1	BV.2
Cramer et al. 1996/97	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.2

4.5 CODE VOTING

Das Prinzip von Code Voting wurde zum ersten Mal von Chaum [102] im Jahr 2001 unter dem Namen „SureVote“ vorgestellt. Code Voting stellt einen Ansatz dar, das Secure Platform Problem[12] zu lösen. Die Idee hinter Code Voting ist, dass die Wählerin ein Code Sheet über einen sicheren Kanal (z. B. per Post) zugesandt bekommt. Auf diesem Code Sheet ist für jeden Kandidaten-Namen ein Vote-Code sowie ein Audit-Code abgedruckt (siehe Abbildung 8).

Kandidat	**Vote-Code**	**ACK-Code**
Darth Vader	9197	7069
Palpatine	7353	9132
Ben Kenobi	4056	6290
R2-D2	5212	2222
C-3PO	3333	9517
Stimmen ID: 21951749166		

Abbildung 8: Aufbau Code Sheet.

Zur Wahl gibt die Wählerin einfach den Vote-Code ein, welcher neben dem von ihr präferierten Kandidaten steht. Anschließend antwortet der Wahlserver mit dem Audit-Code, welchen die Wählerin mit dem Audit-Code auf ihrem Code Sheet vergleicht. Nachdem die Wahlphase beendet wurde, wird der Code wieder in einen Kandidaten zurückübersetzt. Bei SureVote ist dieser Schritt jedoch nicht verifizierbar. Ein weiteres Problem von SureVote ist, dass die Wählerinnen nicht die Möglichkeit besitzen,

12 Das Secure Platform Problem besagt, dass dem Wahlcomputer, über welchen die Stimme abgegeben wird, nicht vertraut werden kann, da dieser z. B. mit Malware infiziert sein könnte [103].

die Erstellung der Code Sheets zu überprüfen (d. h. Zuordnung zwischen Kandidat und Code). Beides führt dazu, dass die Korrektheits-Verifizierbarkeit nicht gewährleistet werden kann [104]. Außerdem erfüllt SureVote nur die äußere individuelle Verifizierbarkeit vor der Auszählung. Joaquim und Ribeiro [104, 105] sowie Joaquim, Ribeiro und Ferreira [106] beschreiben Ansätze, wie Code Voting zusammen mit einer vertrauenswürdigen Smartcard eingesetzt wird. Das Problem ist, dass das Secure Platform Problem bei diesem Protokoll nur auf die Smartcard verschoben wird. Nach der Bewertungsmethodik, welche in dieser Arbeit angewandt wird, bietet diese Vorgehensweise keine Vorteile. Joaquim, Ribeiro und Ferreira [106] beschreiben in diesem Zusammenhang Matrix Code Voting. Ein Vorteil von Matrix Code Voting gegenüber normalem Code Voting ist, dass bei vielen Kandidaten die Code Sheets kurz gehalten werden können, was allerdings zu Kosten der Benutzbarkeit geht.

Helbach und Schwenk [107] führen 2007 ein Protokoll mit sogenannten „Finalization Codes“ ein. Diese Finalization Codes dienen dazu, die Robustheit und die individuelle Verifizierbarkeit zu erhöhen, denn es kann passieren, dass kein (bzw. ein falscher) Audit-Code vom Wahlserver zurückgegeben wird, wenn z. B. ein Angreifer die Kommunikation zufällig manipuliert oder löscht. Die Wählerin kann sich in diesem Fall nicht sicher sein, was genau passiert ist. Mit den Finalization Codes gilt die Wahl als auf dem Server eingegangen, wenn der jeweilige Finalization Code beim Wahlserver eingeht. Dadurch wird ein 3-Way-Handshake mit Vote Code, Audit Code und Finalization Code erreicht. Dies ist vor allem dann interessant, wenn im Wahlsystem nicht bereits während der Wahlphase verifiziert werden kann, dass die Stimme auf dem Bulletin Board eingegangen ist und das Wahlsystem Re-Voting erlaubt.

Helbach, Schwenk und Schäge [108] stellen 2008 ein Protokoll vor, welches den Stimmenkauf (den Verkauf der Code Sheets) durch den Einsatz von Re-Voting verhindert. Die Idee dabei ist, dass die Wählerin ihren Code mit einer Linkable Ring Signatur signiert. In der Auszählungsphase werden alle Stimmen der gleichen Wählerin über die Signatur verknüpft. Anschließend wird nur der neuste Wahlvorgang jeder Wählerin berücksichtigt. Somit ist ein Verkauf der Code Sheets zwar ausgeschlossen, allerdings wird weiterhin der Verkauf von Code Sheets zusammen mit dem jeweiligen Signaturschlüssel ermöglicht.

Joaquim, Ribeiro und Ferreira führen mit VeryVote [109] ein Protokoll ein, welches Ende-zu-Ende-verifizierbar ist. Die Idee dabei ist, dass der Audit-Code mit dem Kandidaten verbunden wird. Dabei kommt das MarkPledge-Verfahren von Neff [110] zum Einsatz, welches die Stimme verschlüsselt. Dieses Verfahren erlaubt es mittels verschiedener Zero-Knowledge-Beweise, allgemein zu beweisen, dass der abgegebene Code richtig interpretiert wurde. Dieser wird zusammen mit der Identität der Wählerin auf einem Bulletin Board veröffentlicht. Zur Auszählung kommt dann ein Mixed-Net zum Einsatz, um die geheime Entschlüsselung der Stimmen zu gewährleisten. Ein großes Problem des Verfahrens ist, dass der Server die Stimme der Wählerin lernt.

Joaquim, Ribeiro und Ferreira [111] kombinieren VeryVote mit der Idee, eine vertrauenswürdige Smartcard zu verwenden [104, 105]. Wie bereits erwähnt, führt dies nur dazu, dass das Secure Platform Problem auf die Smartcard verschoben wird. Unter Annahme des Angreifermodells für Wahlen erster Ordnung mit der Fähigkeit Hilfsmittel zu manipulieren, welche in dieser Arbeit angewandt wird, bietet diese Vorgehensweise keine Vorteile.

Kutylowski und Zagórski [112] beschreiben 2010 mit Scratch, Click & Vote ein Protokoll, das die Idee von den Papier auf basierenden Wahlverfahren ThreeBallot [16] und Punchscan [16, 113] für Internetwahlen adaptiert.

Eine interessante Ergänzungsmöglichkeit zur inneren individuellen Verifizierbarkeit vor der Auszählung liefern Zagorski et al. [114] mit ihrer *Remotegrity* genannten Erweiterung von *Scantegrity* [115–117]. Diese kombiniert Code Voting mit einem sogenannten *Lock-In Code*, der zusammen mit den Codes auf einer separaten Karte unter einem Rubbelfeld an die Wählerinnen versendet wird. Der *Lock-In Code* dient dabei zur Bestätigung, dass der Wahl-Code der jeweiligen Wählerin korrekt erfasst wurde. Sollte von der ausführenden Wahlbehörde ein falscher Code erfasst worden sein, kann die Wählerin dies nicht nur passiv erkennen, sondern durch das unbeschadete Rubbelfeld beweisen und entsprechend aktiv Widerspruch einlegen.

4.5.1 Pretty Good Democracy (PGD)

Pretty Good Democracy (PGD) von Ryan und Teague [118] ist das bekannteste Protokoll auf Basis von Code Voting. Eine Verbesserung dieses Protokolls gegenüber den vorher beschriebenen Protokollen ist, dass sogenannte „Trusties“ in den Prozess der Registrierung vor der Wahl miteinbezogen werden. Des Weiteren wird das Problem der Quittungsfreiheit adressiert, indem jedes Code Sheet nur einen Audit-Code bekommt. Dadurch wird zwar die innere individuelle Verifizierbarkeit eingeschränkt, aber dafür die Quittungsfreiheit eingehalten. Das Problem dieses Protokolls ist, dass die Einhaltung der Integrität und der Korrektheits-Verifizierbarkeit auf der Geheimhaltung des Code Sheets basiert. Um die Code Sheets während der Produktion geheimzuhalten, wird die Erstellung der Codes verteilt durch mehrere Instanzen vorgenommen. Allerdings ist vor allem das Drucken der Code Sheets ein Problem, denn hierbei müssen alle Informationen zentral an einer Stelle zur Verfügung stehen. Dazu werden als erstes $\lambda\nu(c+1)$ Codes generiert, wobei $\lambda > 1$ ein Sicherheits-Parameter, ν die Anzahl der Wählerinnen und c die Anzahl der Kandidaten ist. Anschließend werden alle Codes mit pk_T, dem Public-Key der Trusties, mittels Threshold-Verschlüsselung verschlüsselt. Die verschlüsselten Codes werden durch ein Re-Encryption-Mix-Verfahren gemischt und in einer Tabelle mit $\lambda\nu$ Zeilen und $c+1$ Spalten der P-Tabelle angeordnet. Anschließend wird aus jeder Spalte ein Code Sheet generiert, wobei die Einträge 1 bis c die Codes für die Kandidaten repräsentieren und der $c+1$te Code der Audit-Code ist. Der sogenannte Registrar ist zusammen mit den Trusties für das Drucken der Code Sheets zuständig. Dies ist der kritische Teil des Protokolls. Bei einer praktischen Implementierung müssen hier Maßnahmen zur Geheimhaltung der Code Sheets getroffen werden. Genaue Maßnahmen für die geheime Erstellung der Code Sheets sind von Ryan und Teague [118] beschrieben. Die Code Sheets werden an die Wählerinnen versendet und aus der P-Tabelle wird durch ein Decryption-Mix-Verfahren [119] oder ein Re-Encryption-Mix-Verfahren auf Basis homomorpher Verschlüsselung eine sogenannte Q-Tabelle erstellt und auf dem Bulletin Board veröffentlicht. Während der Wahlphase übermittelt die Wählerin ihre Code Sheet Nummer[13] zusammen mit dem Code durch den Wahlclient an den Server. Anschließend verschlüsselt der Server die Stimme mit pk_T und veröffentlicht diese zusammen mit der Code Sheet Nummer und einem Zero-Knowledge-Proof, der beweist, dass der Server den Klartext kennt. Die Trusties führen dann einen Plaintext-Equivalence-Test[14], finden sie einen Treffer in der Q-Tabelle, dann markieren sie den Treffer, beweisen dies mittels Zero-Knowledge-Proof und entschlüsseln den Audit-Code. Der Audit-Code wird dann von den Trusties über den Server an die Wählerin gesendet. Nach der Wahlphase existiert für jede Stimme eine Liste mit Kandidaten und ein Index, welcher die Wahl markiert. Aus Basis dieser Informationen kann die Stimme errechnet werden [119, 122, 123].

4.5.2 Protokolle auf Basis von PGD

Pretty Understandable Democracy (PUD) von Budurushi et al. [124] erweitert PGD, sodass das Verfahren für die Wählerin besser verständlich ist, wodurch das Vertrauen in das Wahlprotokoll erhöht werden soll. Dazu vereinfachen Budurushi et al. den Erstellungsprozess der Code Sheets, indem anstatt verschiedener Mix-Verfahren homomorphe Stimmauszählung eingesetzt wird. Außerdem wird das Code Sheet aus drei Teilen zusammengesetzt, wobei die Registration Authority (RA) die Namen, die Voting Authority 1 (VA_1) die Code Spalte 1 und die Voting Authority 2 (VA_2) die zweite Code Spalte druckt. Alle drei Code Sheets werden in Umschläge verpackt, welche mit der Code Sheet ID i beschriftet sind. VA_1 und VA_2 wählen dabei die Codes zufällig und einmalig. Die Codes von VA_1 und VA_2 sowie die Kandidatennamen von RA werden mit einem Threshold-Encryption-Verfahren mittels des öffentlichen Schlüssels der Trusties verschlüsselt und auf einem Bulletin Board veröffentlicht. Von einer weiteren Instanz werden die Umschläge an Hand ihrer Code Sheet ID in einen weiteren Umschlag gepackt, gemischt und an die Wählerin gesendet. Die Trusties testen den Prozess der Erstellung der Code Sheets, indem sie zufällig gewählte Code Sheets öffnen und mit denjenigen auf dem Bulletin

13 Entspricht der Zeilennummer der P-Tabelle, aus welcher das Code Sheet erstellt wurde.
14 Details dazu werden von Jakobsson und Jules [120] sowie Ting und Huang [121] beschrieben.

Board vergleichen. Dazu werden mehr Code Sheets erstellt, als Wählerinnen vorhanden sind, da die geöffneten Code Sheets nicht mehr verwendet werden können.

Während der Wahlphase setzt die Wählerin die drei Code Sheets dann wie in Abbildung 9 schematisch beschrieben zusammen.[15] Anschließend authentifiziert sich die Wählerin über eine verschlüsselte Verbindung bei RA und sendet beide Codes aus der Zeile mit dem Kandidaten, welchen sie wählen möchte, an RA. RA leitet die entsprechenden Codes an VA_1 oder VA_2 weiter, welche den Kandidatennamen erneut verschlüsseln und dann auf dem Bulletin Board veröffentlichen. Zum Abschluss senden VA_1 und VA_2 den Audit-Code an RA, welcher diesen an die Wählerin ausliefert.

In der Auszählungs-Phase veröffentlicht RA die Anzahl der authentifizierten Wählerinnen. Dann werden die Stimmen, welche von RA_1 auf dem Bulletin Board veröffentlicht wurden, homomorph addiert und das Ergebnis veröffentlicht. Anschließend passiert das Gleiche mit den Stimmen von RA_2. Beide Summen werden von den Trusties entschlüsselt und die Ergebnisse zusammen mit einem Zero-Knowledge-Proof für die korrekte Entschlüsselung auf dem Bulletin Board veröffentlicht. Das Prinzip, auf welchem PUD beruht, ist die bereits in Abschnitt 4.1 beschriebene Funktionstrennung, wodurch in diesem Beispiel immer zwei Komponenten des Wahlsystems kooperieren müssen, um die Sicherheitsziele zu untergraben.

i
$Code_{A,i,1}$
$Code_{A,i,2}$
⋮
$Code_{A,i,n}$
A: $Ack_{A,i}$

i
$Code_{B,i,1}$
$Code_{B,i,}$
⋮
$Code_{B,i,n}$
B: $Ack_{B,i}$

i
$\phi(Kandidat_1)$
$\phi(Kandidat_2)$
⋮
$\phi(Kandidat_n)$

Abbildung 9: Aufbau Code Sheets in PUD.

Neumann et al. [125] implementieren PUD als Proof-of-Concept[16], womit sie anschließend eine Testwahl durchführen. Dabei wurde das Stimmen-Encoding für die homomorphe Verschlüsselung der Stimme insofern verbessert, dass die Entschlüsselung des Ergebnisses schneller durchgeführt werden kann. Dazu kommt das Verfahren von Kiayias und Yung [90], das eine Verbesserung durch die Cross-Validierung zwischen VA_1 und VA_2 erreicht. Dabei überprüfen VA_1 und VA_2, ob sie die gleiche Position des Codes gesendet bekommen. Dadurch wird ausgeschlossen, dass eine Wählerin die Auszählung des Ergebnisses verhindern kann, indem sie eine ungültige Stimme für zwei unterschiedliche Kandidaten generiert.

Tabelle 5: Bewertung der Wahlverfahren basierend auf Code Voting unter Annahme des Angreifermodells für Wahlen erster Ordnung.

	Verifizierbarkeit		Wahl-geheim-nis	Nicht-Erpress-barkeit	Robust-heit	Benutz-barkeit	
	ind.	uni.				Wahl	Verif.
Helbach 2007	IV.2.2	x	x	x	x	BW.2.2	BV.1
VeryVote 2009	IV.1.1	WV.1, EV.1, KV.1	x	x	RI.1	BW.2.2	BV.2
EVIV 2013	IV.1.1	WV.1, EV.1, KV.1	x	x	RI.1	BW.2.2	BV.2
PGD 2009	IV.2.1	EV.1, KV.1	x	x	RI.1	BW.2.2	BV.2
PUD 2013	IV.2.1	KV.1	x	x	x	BW.2.2	BV.2
PUD 2014	IV.2.1	KV.1	x	x	RI.1	BW.2.2	BV.2

15 Die Reihenfolge wird vorab festgelegt und in den Wahlunterlagen beschrieben.
16 Dabei werden nicht alle nötigen Sicherheitsvorkehrungen implementiert. So wird z. B. die Authentifizierung nur durch Eingabe des Namens realisiert.

4.6 ATTRIBUTE-BASED CREDENTIALS

Abendroth et al. [126] beschreiben in ihrem Beitrag zwar unter anderem einen Anwendungsfall von Attribute-Based Credentials für Online-Abstimmungen. Allerdings bezieht sich dieser weniger auf Wahlen, sondern viel mehr auf Petitionen und ähnliche Partizipationsmöglichkeiten. Abendroth et al. warnen weiterhin explizit davor, Attribute-Based Credentials für politische Internetwahlen einzusetzen, die im Einklang mit zum Zeitpunkt der Arbeit geltendem Recht stehen sollen. Nichtsdestotrotz stellen Attribute-Based Credentials eine mögliche Lösung für den Zielkonflikt zwischen der Anonymität der Wählerinnen auf der einen Seite und den Anforderungen, dass nur berechtigte Personen an der Wahl teilnehmen sowie jeweils nur genau ein Mal wählen dürfen, auf der anderen Seite dar. Das Konzept wird deshalb im Folgenden kurz vorgestellt.

Attribute-Based Credentials sind kryptografische Mechanismen, die es einer Partei (in der Regel einem Benutzer) erlauben, einer anderen Partei (in der Regel ein Anbieter eines Dienstes oder eben der Betreiber einer Internetwahlplattform) zu beweisen, über bestimmte Charakteristika zu verfügen, ohne diese Charakteristika oder gar seine Identität im Detail preiszugeben. Außerdem soll es nicht möglich sein, mehrere solcher Preisgaben des selben Benutzers miteinander in Verbindung zu bringen, es sei denn, der Benutzer stimmt dem unter Benutzung eines Pseudonyms, das seine wahre Identität verbirgt, ausdrücklich zu. Zusätzlich zu den beiden bereits beschriebenen Parteien, also dem Benutzer und dem Anbieter (der aufgrund der Tatsache, dass er den Beweis des Benutzers überprüft, auch *Verifier* genannt wird), sieht der Einsatz von Attribute-Based Credentials in der Regel auch noch eine dritte, vom Benutzer und *Verifier* als vertrauenswürdig erachtete Partei vor, die dafür verantwortlich ist, dem Benutzer nach erfolgreicher Demonstration bestimmter Attribute (z. B. Bestätigung seiner Identität, seines Alters oder dass er im Besitz bestimmter Informationen ist) seine Credentials zur Verfügung zu stellen und ggf. deren Gültigkeit gegenüber dem *Verifier* zu bestätigen (als Echtzeit-Alternative zu den bereits bei Ausstellung „beglaubigten“ Self-Claimed Attributes). Diese Partei wird in der Literatur in der Regel *Issuer* genannt. Auf Basis bestimmter, vom *Verifier* formulierter *Presentation Policies*, die definieren, welche Informationen überprüft werden sollen, kann der Benutzer nun Zugriff auf Dienste beantragen. Hierfür muss er dem *Verifer* der *Presentation Policy* entsprechende Credentials zur Verfügung stellen, sodass dieser die erforderlichen Attribute verifizieren kann. Nach erfolgreicher Verifizierung wird dem Benutzer sodann der Dienst zur Verfügung gestellt. Abbildung 10 stellt den beschriebenen Prozess zur Veranschaulichung noch einmal vereinfacht dar.

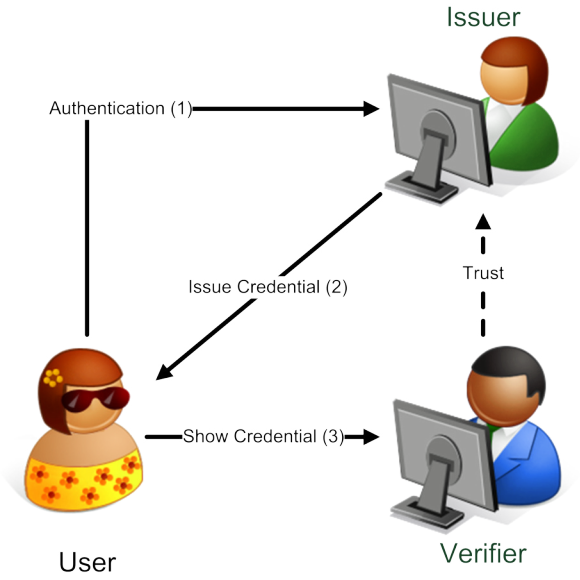


Abbildung 10: Schematische Darstellung der Funktionsweise von Attribute-based Credentials.

Praktische Implementierungen von Attribute-Based Credentials stellen die Systeme *U-Prove* [127] von Microsoft sowie *idemix* [128] von IBM dar [129]. Letzteres basiert unter anderem auf nach ihren Erfindern benannten *Camenisch-Lysyanskaya* (oder auch kurz CL) Signatures [130].

Put et al. [131] stellen ihr System *avisPoll* vor, welches unter anderem auf *idemix* aufbaut und speziell für Wahlen und Abstimmungen entworfen wurde, bei denen die Wählerinnen kein Vertrauen in die Institution haben, die die Wahl organisiert. *avisPoll* ist dabei als Online-Service konzipiert, der von einem unabhängigen Anbieter (*Poll Service Provider*) angeboten wird. Weiterhin sieht das System die Rollen der Wählerinnen, der optionalen Auditoren (*Poll Auditors*), des Issuers (*Credential Issuer*) sowie des Wahlorganisators (*Poll Organizer*) vor, welcher gleichzeitig auch eine Art Kunde des Anbieters ist. Die Aufgabe des des Auditors ist die Überwachung des gesamten Wahlprozesses. Der Wahlorganisator bestimmt den Gegenstand und den Ablauf der Wahl sowie die Anforderungen an wahlberechtigte Personen in einer *Policy Description*. Diese Policy Description enthält unter anderem auch einen öffentlichen Schlüssel. Der dazugehörige private Schlüssel ist lediglich dem Wahlorganisator zugänglich, kann bei einer Beteiligung von Auditoren aber auch zwischen diesen und dem Wahlorganisator aufgeteilt werden. Der Anbieter wiederum führt die Wahl dann auf Basis der vom Wahlorganisator in der Policy Description definierten Rahmenbedingungen durch. Der Issuer stellt nach der Registrierung der Wählerin die für die Authentifizierung der Wählerinnen notwendigen *idemix*-Credentials aus, mit denen sich die Wählerin fortan anonym beim Anbieter authentifizieren kann. Die Wählerin verschlüsselt ihre Wahlentscheidung nun mit Hilfe eines zufällig generierten AES-Schlüssels und verschlüsselt eben diesen AES-Schlüssel mit dem öffentlichen Schlüssel aus der Policy Description. Außerdem erstellt sie ein Domain-Pseudonym sowie einen *idemix*-Beweis über die Kenntnis eben dieses Domain-Pseudonyms, über gültige *idemix*-Credentials sowie ggf. über zusätzliche Informationen. Dieser *idemix*-Beweis wird dann zusammen mit der verschlüsselten Wahlentscheidung sowie einem Zeitstempel an den Anbieter gesendet. Dieser prüft, ob der *idemix*-Beweis konform zur Policy Description ist und speichert die empfangenen Informationen bei einer positiven Prüfung in seiner Datenbank und sendet dem Nutzer eine eindeutige, verifizierbare Bestätigung, sodass die Wählerin sich sicher sein kann, dass ihre Daten tatsächlich beim Anbieter angekommen sind. Sollte der Verifizierungsprozess nicht erfolgreich verlaufen, kann sich die Wählerin beim Wahlorganisator oder einem Auditor beschweren. Sobald der in der Policy Description festgelegte Wahlzeitraum abgelaufen ist, erstellt der Server einen Merkle-Baum, der aus allen an die Nutzer gesendeten, einmaligen Bestätigungen besteht sowie Start und Endzeitpunkt der Wahl besteht und signiert eben diesen Merkle-Baum. Dann sendet er diesen sowie den Inhalt der Datenbank an den Wahlorganisator sowie die Auditoren. Diese prüfen und signieren nach erfolgreicher Prüfung beides, woraufhin Merkle-Baum und Datenbank veröffentlicht werden. Nun kann jede Wählerin an Hand ihrer eindeutigen Bestätigung überprüfen, ob ihre Stimme tatsächlich in das Wahlergebnis einfließen wird. Durch die Veröffentlichung der Datenbank können nun Wahlorganisator sowie Auditoren mit Hilfe ihres privaten Schlüssel die verschlüsselten AES-Schlüssel und mit deren Hilfe die eigentlichen Wahlentscheidungen entschlüsseln und somit die Wahl auszählen. Rückschlüsse darauf, welche Wählerin welche Stimme abgegeben hat, sind aufgrund der vom Issuer anonym ausgestellten Credentials nicht möglich. Die Auszählung der Wahl wird standardmäßig vom Wahlorganisator bzw. den Auditoren vorgenommen, kann auf Verlangen des Wahlorganisators jedoch ebenfalls vom Anbieter durchgeführt werden.

Tabelle 6: Bewertung des Wahlverfahrens basierend auf Attribute-based Credentials unter Annahme des Angreifermodells für Wahlen erster Ordnung.

	Verifizierbarkeit		**Wahl-geheim-nis**	**Nicht-Erpress-barkeit**	**Robust-heit**	**Benutz-barkeit**	
	ind.	**uni.**				**Wahl**	**Verif.**
Put et al. 2014	IV1.1	x	x	x	x	BW.1	BV.1

4.7 BLOCKCHAIN

Längst bestimmt das Thema *Blockchain* nicht mehr nur digitale Zahlungsmittel bzw. Krypto-Währungen wie die allseits bekannten *Bitcoins* [132]. Andere Anwendungen, wie z. B. Smart Contracts [133] sind bereits auf dem Vormarsch. Auch Vorschläge, Blockchain-Technologien für digitale Wahlen zu nutzen, konnte man in jüngster Zeit vermehrt vernehmen [134]. Es gibt zwar bereits einige kommerzielle Produkte[17], Prototypen[18] und Whitepaper[19], die sich auf den Einsatz der Blockchain-Technologie berufen, allerdings sind deren Dokumentationen und anderweitige öffentlich zugängliche Informationen eher vage, sodass sie für eine Auswertung im Rahmen dieser Arbeit nicht in Frage kommen [135, 136]. Leider gibt es bisher nur wenige Ansätze auf wissenschaftlichem Niveau. Nichtsdestotrotz soll der folgende Abschnitt einen kurzen Überblick über die zum Zeitpunkt der Arbeit veröffentlichten Verfahren in diesem Segment bieten.

Zunächst soll jedoch kurz die allgemeine Funktionsweise der Blockchain umrissen werden. Obwohl der Einsatz kryptografisch verketteter Daten bereits früher diskutiert wurde [137], kam deren Nutzung in der hier besprochenen Form zum ersten Mal 2008 auf [132], wenn auch noch nicht direkt unter dem Begriff „Blockchain“. Die Blockchain wird nicht zentral auf einem System gespeichert, stattdessen kann jeder eine eigene Kopie auf seinem eigenen System vorhalten. Änderungen werden dann entsprechend auf allen Kopien repliziert. Um die Integrität der Blockchain zu gewährleisten sind die einzelnen Blöcke so miteinander verkettet, dass es sehr schwierig ist, einzelne Blöcke nachträglich, also nachdem sie zur Blockchain hinzugefügt wurden, zu manipulieren. Wie in Abbildung 11 zu sehen ist, bestehen die Blöcke dabei aus Merkle-Bäumen, die in einem Root-Hash münden. Die Merkle-Bäume wiederum bestehen aus Hash-Werten, die z. B. Transaktionen eindeutig beschreiben. Außerdem hat jeder Block einen Block-Header, der aus eben genanntem Root-Hash des Merkle-Baumes, dem Hash des vorherigen Block-Headers sowie einer Nonce besteht. Die Blockchain selbst besteht im Grunde nun aus einer Reihe dieser Blöcke, die darüber verkettet sind, dass jeder Block-Header den Hash des Block-Headers des jeweils vorhergehenden Blocks beinhaltet.

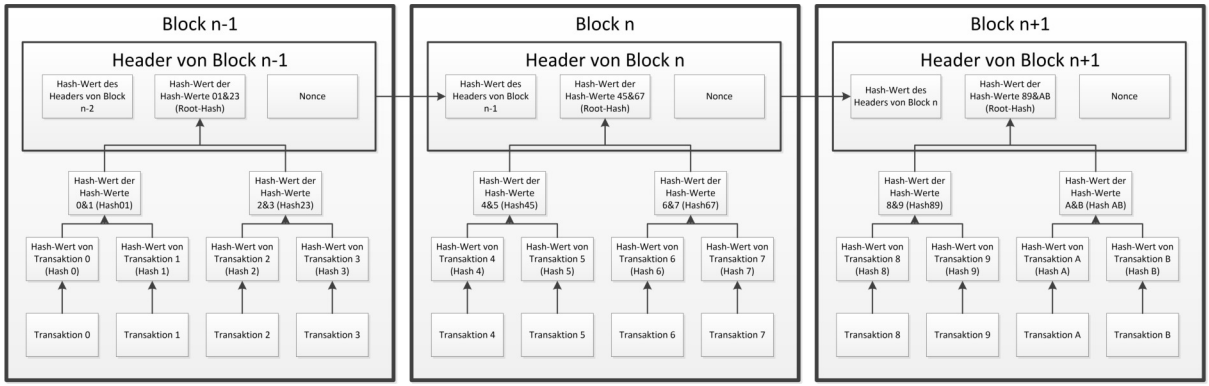


Abbildung 11: Grundlegende Funktionsweise der Blockchain in Anlehnung an Bitcoin [132].

Die Vorteile davon, die Durchführung einer Wahl auf einer Blockchain statt auf einer oder mehreren Datenbanken abzubilden, sind zum Einen die erhöhte Verfügbarkeit durch den verteilten und partizipativen Charakter, der es jeder interessierten Person oder Institution ermöglicht, die Organisation der Wahl durch das Bereitstellen eines eigenen Nodes zu unterstützen. Zum Anderen erhöhte Verifizierbarkeit und Integrität durch den verbindlichen Charakter der öffentlichen und frei zugänglichen Blockchain und den damit verbundenen sehr hohen Aufwand, eine Manipulation durchzuführen [138].

17 Z. B. *Follow My Vote* (https://followmyvote.com) oder *Vote Watcher* (http://votewatcher.com/#voting).
18 Z. B. *PublicVotes* (https://github.com/domschiener/publicvotes) oder *VotoSocial* (https://github.com/votosocial).
19 Z. B. *Boulé* (https://www.boule.one/boule-whitepaper.pdf) oder *BitCongress* (https://bravenewcoin.com/assets/Whitepapers/BitCongressWhitepaper.pdf).

Ayed [139] schlägt vor, für jeden Kandidaten eine eigene Blockchain zu generieren, deren erster Block den entsprechenden Kandidaten eindeutig identifiziert. Außerdem gibt es einen leeren Block für Protestwählerinnen, die es vorziehen, ungültig zu wählen. Ayed bietet keine eigene Authentifizierungs-Methodik und nimmt für sein Verfahren explizit an, dass teilnehmende Wählerinnen z. B. bereits anhand ihrer Sozialversicherungsnummer und einer zusätzlichen, vor der Wahl bereitgestellter Wahlnummer authentifiziert sind. Jede abgegebene Stimme wird der Blockchain des entsprechenden Kandidaten hinzugefügt und enthält einen Hashwert, der aus der Wahlnummer und dem Namen der Wählerin sowie der jeweils vorherigen Stimme generiert wird. Falls mehrere Stimmen gleichzeitig an unterschiedlichen Instanzen der selben Blockchain abgegeben werden und deshalb alle den selben Vorgängerblock bzw. die selbe Blocknummer haben, gilt die *Longest Chain Rule*. Das heißt, es wird in diesem Falle mit dem Block fortgefahren, der als erstes einen validen Nachfolger hat. Andere, zeitgleich abgegebene Stimmen werden als verwaiste Blöcke gespeichert. Zwar stellt dies hier laut Ayed kein Problem dar, da jeder Kandidat seine eigene Blockchain hat und auch verwaiste Blöcke bei der Auszählung berücksichtigt werden können. Allerdings sehen die Autoren in dieser Vorgehensweise eine Gefahr für die Integrität der Blockchain, da dadurch nachträglich gefälschte Blöcke als angeblich verwaiste Blöcke in die Blockchain eingeschleust werden könnten.

Barnes, Brake und Perry [140] schlagen einen Ansatz vor, der die Blockchain-Technologie in bereits bestehende Online- sowie Präsenzwahlverfahren integriert. Allerdings bleiben sie die Antwort auf die Frage schuldig, welches Problem genau sie durch die Integration der Blockchain lösen möchten. Die Wahl ist dabei sowohl in einem Wahllokal als auch online möglich. Das Netzwerk besteht aus den drei Ebenen *Local*, *Constituency* und *National*, die jeweils mit den Ebenen des deutschen Wahlsystems vergleichbar sind. Die Wahllokale sind dabei auf der lokalen Ebene verortet und stellen jeweils einen Node dar. Diese lokalen Nodes sind einer Constituency-Node zugeordnet, von der die lokalen Nodes einen Public Key zur Verschlüsselung der entgegengenommenen Stimmen bereitgestellt bekommen. Sie können nur mit dieser und mit anderen lokalen Nodes kommunizieren, die ebenfalls dieser Constituency-Node zugeordnet sind. Die Constituency-Nodes wiederum sind einer nationalen Node zugeordnet. Die nationalen Nodes sind dafür zuständig, Transaktionen zu minen und Blöcke der Blockchain hinzuzufügen. Es sind die nationalen Nodes, die zu Kontrollzwecken auch von unabhängigen Personen und Institutionen betrieben werden können. Weiterhin sind zwei voneinander getrennte Blockchains vorgesehen, eine Wahl-Blockchain sowie eine Wählerinnen-Blockchain. Zur Vorbereitung der Wahl wird jeder wahlberechtigten Person postalisch ein Passwort zugesandt. Als zugehörige Benutzernamen bzw. IDs fungieren z. B. Sozialversicherungs- oder Personalausweisnummern. Unter Benutzung dieser Informationen können sich Wählerinnen fortan an einer Constituency-Node für die Wahl registrieren. Sobald sich eine Wählerin zur Wahl registriert hat, wird von der zugehörigen National-Node eine Transaktion auf der Wählerinnen-Blockchain erzeugt. Daraufhin entscheidet ein automatisierter Miner auf Basis der von der Wählerin zur Verfügung gestellten Informationen, ob die Wählerin verifiziert wird und fügt seine Verifizierungsentscheidung ebenfalls als Transaktion der Wählerinnen-Blockchain hinzu. Fällt die Verifizierung der Wählerin positiv aus, wird ihr postalisch ein Stimmzettel zugesandt, der einen (QR-)Wahl-Code zur einmaligen Authentifizierung seiner Stimmabgabe enthält. Um letztendlich ihre Stimme abgeben zu können, muss sich die Wählerin unter Zuhilfenahme aller drei ihr in diesem Kontext zur Verfügung stehenden Informationen (ID, Wahl-Passwort, Wahl-Code) authentifizieren. Ist dies geschehen, überprüft das System anhand der Wählerinnen-Blockchain, ob die Wählerin bereits gewählt hat. Falls nicht, wird sie zur Wahlmaske weitergeleitet und kann ihre Wahl treffen. Nach Übermittlung der bestätigten Wahl wird diese vom lokalen Node unter Zuhilfenahme des Public Keys der Constituency-Node verschlüsselt und an eben diese weitergeleitet, wo sie zu einem Block hinzugefügt wird. Sobald alle lokalen Nodes im Besitz des aktuellen Stands des Blocks sind, generiert die Constituency-Node eine Transaktion, um die Wahlberechtigung der Wählerin in der Wählerinnen-Blockchain ungültig zu machen. Sobald der Block, in dem die Stimme verschlüsselt gespeichert wurde, genügend Transaktionen enthält, wird er der Wahl-Blockchain hinzugefügt. Der von den Autoren vorgeschlagene Ansatz hat einige Schwachstellen. So bietet er z. B. Angriffsfläche für Erpressungen, da die Transaktionen der Wählerinnen-Blockchain persönliche Daten enthalten, welche einem Angreifer Hinweise über die Teilnahme einzelner Personen an der Wahl gewähren kön-

nen. Außerdem besteht ein Risiko, dass durch zeitliche Korrelation der Transaktionen auf Wahl- und Wählerinnen-Blockchain das Wahlgeheimnis gebrochen werden kann.

Lee et al. [141] setzen in ihrem auf Bitcoin basierenden Verfahren für die Authentifizierung der Wählerinnen auf eine *Trusted Third Party* (TTP) und damit auf Funktionstrennung von der Behörde, die die Wahl durchführt. Hierfür generiert die Wählerin zuerst ein Schlüsselpaar bestehend aus privatem und öffentlichem Schlüssel. Letzteren benutzt sie, um sich eine zugehörige Bitcoin-Adresse zu generieren. Dann übermittelt die Wählerin den Hashwert eines von der Behörde im Vorfeld zur Verfügung gestellten Geheimnisses an die TTP. Diese leitet sie an die Behörde weiter und erfragt, ob dieser Hashwert im Wählerverzeichnis existiert. Falls nicht, bricht die TTP ab. Falls doch, erfragt diese bei der Wählerin auch die geheime Nachricht und prüft, ob sich aus dieser tatsächlich der Hashwert berechnen lässt. Falls nicht, bricht die TTP ab. Falls doch, wird die Wählerin als verifiziert markiert und ihre Bitcoin-Adresse in der TTP gespeichert. Auf Basis dieser Adresse kann die TTP bei der Auszählung nun entscheiden, welche Transaktionen bzw. Stimmen auf der Blockchain von wahlberechtigten und welche von nicht wahlberechtigten bzw. nicht authentifizierten Personen stammen. Dabei hat sie keinerlei Kenntnis über die wahre Identität der wählenden Person. Allerdings setzt dies voraus, dass die TTP nicht über die für die Authentifizierung vorgesehene Kommunikation hinaus mit der Behörde kollaboriert.

Ein ebenfalls auf Funktionstrennung basierendes Verfahren stellen Bistarelli et al. vor [142]. Anstatt eines selbst entwickelten Authentifizierungsverfahrens, wie von Lee et al. vorgeschlagen, setzt es auf den Einsatz eines Anonymous Kerberos Protokolls [143] zur Authentifizierung der wahlberechtigten Personen sowie zur Trennung zwischen Authentication Server (AS) und Token Distribution Server (TDS), welches dazu dient, die Wahlentscheidung vor der Behörde geheim zu halten, die die Wahl organisiert. Auch hier wird wieder davon ausgegangen, dass diese beiden Komponenten nicht miteinander kollaborieren. Damit muss auch hier, wie bereits beim Ansatz von Lee et al. [141], einem Teil der beteiligten Instanzen vertraut werden.

Liu und Wang [144] stellen ein Konzept vor, welches aus einer Kombination aus Blinden Signaturen und der Blockchain besteht. Das Protokoll sieht drei Arten von Akteuren vor: Wählerinnen, Organisatoren und Inspekteure, welche jeweils alle ein Schlüsselpaar bestehend aus privatem und öffentlichem Schlüssel besitzen. Wählerinnen besitzen zusätzlich noch ein zweites Schlüsselpaar, welches in keiner Weise mit ihrer Identität in Verbindung gebracht werden kann. Für die digitalen Stimmzettel beschreiben Liu und Wang ein spezielles Format, welches jeweils Felder zur eindeutigen Identifizierung, zur Sicherstellung der Korrektheit sowie zur eigentlich Wahl beinhaltet. Um zu wählen, erstellt die Wählerin eben solch einen Stimmzettel, berechnet einen Hashwert davon und verschlüsselt diesen unter Zuhilfenahme eines Blendfaktors. Den verschlüsselten Hashwert des Stimmzettels signiert sie daraufhin mit dem privaten Schlüssel aus dem Schlüsselpaar, das ihrer Identität zugeordnet ist, und sendet den verschlüsselten Hashwert zusammen mit der Signatur dem Organisator. Der Organisator überprüft nun, ob die Wählerin überhaupt wahlberechtigt ist. Falls ja, signiert er den verschlüsselten Hashwert mit Hilfe eines blinden Signaturverfahrens und sendet diese Signatur zurück an die Wählerin. Nun übermittelt die Wählerin den mit Hilfe des Blendfaktors verschlüsselten, blind signierten Hashwert per Blockchain-Transaktion an die Inspekteure. Diese überprüfen ebenfalls die Berechtigung zur Wahl und außerdem, ob der zu signierende Wert der selbe ist, der vorher dem Organisator vorgelegt wurde. Falls beides der Fall ist, signieren auch die Inspekteure den Wert mit Hilfe eines blinden Signaturverfahrens senden diese Signatur zurück an die Wählerin. Diese nutzt nun den nur ihr bekannten Blendfaktor, um die vom Organisator und von den Inspekteuren erhaltenen blinden Signaturen zu entblenden. Um ihre Stimme final abzugeben nutzt die Wählerin nun ihr nicht mit ihrer Identität in Verbindung zu bringendes Schlüsselpaar, um den Klartext seines digitalen Stimmzettels zusammen mit den Signaturen des Organisators und der Inspekteure als Transaktion auf der Blockchain zu veröffentlichen. Zwar basiert das Protokoll laut Lius und Wangs eigenen Angaben nicht auf dem Vertrauen in eine oder mehrere Instanzen, allerdings gilt auch hier, dass es auf einer Funktionstrennung zwischen sich gegenseitig kontrollierenden Instanzen aufbaut, wofür letztendlich ein gewisses Maß an Vertrauen in die Voraussetzung nötig ist, dass die Inspekteure nicht mit dem Organisator kollaborieren.

Cruz und Kaji [145] stellen einen ähnlichen Ansatz wie Liu und Wang vor, der ebenfalls auf dem Einsatz von blinden Signaturen und der Blockchain-Technologie beruht. Allerdings wird hier nicht ein generisches Modell der Blockchain eingesetzt, sondern konkret auf Basis von Bitcoin gearbeitet. Cruz und Kaji schlagen vor, die Wahlberechtigung auf Basis zufällig verteilter Prepaid Bitcoin Cards[20] (PBC) zu überprüfen. Hierfür wird eine Liste aller berechtigten öffentlichen Schlüssel erstellt, die Zuordnung der jeweiligen PBCs zu den Wählerinnen jedoch nicht festgehalten. Die Wahlberechtigten können die auf der PBC enthaltenen Informationen dann nutzen, um ihre vom Wahlorganisator blind signierte und nach wie vor geblendete Stimme per Bitcoin-Transaktion an den Wahl-Auszähler zu senden bzw. zu veröffentlichen, sodass alle blinden Signaturen auf ihre Gültigkeit überprüft werden können. Nach Ablauf des Wahlzeitraums senden dann alle Wählerinnen ihren Blendfaktor an den Auszähler, der somit die Stimmen entschlüsseln und auszählen kann. Durch die nach wie vor nicht bekannte Assoziation von Wählerinnen und PBC soll das Wahlgeheimnis weiterhin bestehen bleiben.

Auch Zhao und Chan [146] greifen für ihr vorgeschlagenes Internetwahlverfahren auf das bestehende Bitcoin-System zurück. Allerdings benutzen sie nur Bit-Commitment Schemes anstatt blinder Signaturen, um sich auf eine binäre Wahl (0 oder 1) festzulegen, die mit Hilfe eines im Vorfeld von den Organisatoren bereit gestellten Zufallswertes unkenntlich gemacht wurde. Das Verfahren erlaubt deshalb ausschließlich Wahlen mit zwei Abstimmungsoptionen bzw. Kandidatinnen durchzuführen. Hierfür erhält jede Wählerin eine andere Zufallszahl. Alle Zufallszahlen zusammen addiert ergeben eine ebenfalls vor der Wahl festgelegte Summe n. Nach Ablauf des Wahlzeitraums veröffentlicht jede Wählerin ihre immernoch unkenntlich gemachte Entscheidung per Bitcoin-Transaktion. Das Ergebnis der Wahl wird dann ermittelt, indem alle von den Wählerinnen veröffentlichten Werte $mod\ n$ aufaddiert werden. Wählerinnen beweisen dabei mit Hilfe von Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK) [147], dass sie das Protokoll richtig ausgeführt bzw. die richtige Zufallszahl benutzt haben, ohne diese tatsächlich offenzulegen.

Abschließend lässt sich sagen, dass die vorgestellten, auf der Blockchain-Technologie beruhenden Ansätze größtenteils von einer bereits auf irgendeine Art und Weise registrierten bzw. authentifizierten Wählerin ausgehen. Um diese Lücke in der Wahl-Prozesskette zu schließen, regen die Autoren an, die vorgestellten Blockchain-Ansätze um eine Registrierung bzw. Authentifizierung auf Basis des in Abschnitt 4.6 vorgestellten Ansatzes der Attribute-Based Credentials oder auf Basis der in Abschnitt 4.2 vorgestellten blinden Signaturen zu ergänzen.

Tabelle 7: Bewertung der Wahlverfahren basierend auf Blockchain unter Annahme des Angreifermodells für Wahlen erster Ordnung.

	Verifizierbarkeit		Wahl-geheim-nis	Nicht-Erpress-barkeit	Robust-heit	Benutz-barkeit	
	ind.	uni.				Wahl	Verif.
Lee et al. 2016	IV.1.1	KV.1	x	x	x	BW.1.1	BV.1
Barnes et al. 2016	IV.1.1	KV.1	x	x	x	BW.2.3	BV.1
Ayed 2017	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Bistarelli et al. 2017	IV.1.1	KV.1	x	x	x	BW.1.1	BV.1
Liu, Wang 2017	IV.1.1	KV.1	x	x	x	BW.1.1	BV.1
Cruz, Kaji 2017	IV.1.1	KV.1	x	x	x	BW.2.2	BV.1
Zhao, Chan 2016	IV.1.1	KV.1	x	x	x	BW.1.1	BV.1

20 Prepaid Bitcoin Cards sind virtuelle oder (was in diesem Anwendungsfall mehr Sinn ergibt) physikalische Prepaid-Karten, auf die der öffentliche und - unter einem Rubbelfeld versteckt - private Schlüssel eines Bitcoin-Wallets zusammen mit eben dessen Wert in Bitcoins gedruckt ist.

4.8 HYBRIDE VERFAHREN AM BEISPIEL DU-VOTE

Du-Vote („**D**evices that are **u**ntrusted used to **Vote**“) ist ein neues Verfahren, welches 2015 von Grewal et al. [148] zum ersten Mal publiziert wurde. Dieses Verfahren ist bislang nur als theoretisches Konzept zu verstehen. Zum Zeitpunkt dieser Arbeit gab es keine öffentlich verfügbare Implementierung, weshalb es folglich bei Wahlen auch noch nicht zum Einsatz kam. Die Idee hinter Du-Vote ist, dass der Wahlvorgang zwischen Wahlcomputer, Server und einem Hardware-Token aufgeteilt wird. Dazu generiert der Client eine Code Page, welche auf dem Bildschirm ausgegeben wird. Die Wählerin tippt ihre Wahl dann in das Hardware-Token ein, welches die Wahl mittels eines homomorphen Commitments codiert und den entsprechenden Wert c^* auf dem Display ausgibt. Die Wählerin überträgt diesen Wert durch den Wahlcomputer auf den Server. Zusammen mit weiteren Informationen des Wahlcomputers überprüft der Server die Berechnungen des Wahlcomputers und berechnet die verschlüsselte Stimme aus dem Code c^*. Sie wird erneut verschlüsselt und auf einem Bulletin Board veröffentlicht. Um die Verifizierbarkeit des Servers zu gewährleisten, beweist der Server die von ihm erstellten Berechnungen mittels mehrerer Signature-based Proofs-of-Knowledge über Ringsignaturen. Die Auszählung kann wahlweise über homomorphe Addition der einzelnen Stimmen oder über verifizierbares Mischen erfolgen.

Du-Vote ist dafür entwickelt, das Wahlgeheimnis und die Verifizierbarkeit zu gewährleisten. Die Verifizierbarkeit kann mit hoher Wahrscheinlichkeit sogar eingehalten werden, wenn alle Komponenten des Systems kompromittiert sind. Das Wahlgeheimnis kann hingegen ausschließlich unter der Annahme eingehalten werden, dass der Wahlcomputer und der Server oder der Wahlcomputer und das Hardware-Token nicht gleichzeitig vom Angreifer kontrolliert werden. Bei näherer Betrachtung fällt allerdings auf, dass Du-Vote auch die Quittungsfreiheit erfüllt sowie Maßnahmen gegen Erpressung und Stimmenkauf implementiert. Das Verfahren ist in Abschnitt 5.3 genauer beschrieben.

Tabelle 8: Bewertung hybrider Wahlverfahren unter Annahme des Angreifermodells für Wahlen erster Ordnung.

	Verifizierbarkeit		Wahl-geheim-nis	Nicht-Erpress-barkeit	Robust-heit	Benutz-barkeit	
	ind.	uni.				Wahl	Verif.
Du-Vote 2015	IV.1.1	WV.1, EV.1, KV.1	x	x	x	BW.3.2	BV.2

4.9 BEWERTUNG

In diesem Abschnitt werden die in diesem Kapitel beschriebenen Wahlverfahren bewertet. Dazu wird die in Kapitel 3 vorgestellte Methodik verwendet. Um die Unterschiede zwischen den einzelnen Verfahren herauszustellen, werden sie anhand von drei verschiedenen Angreifermodellen bewertet, die jeweils aus Basis der Kritikalität der drei in Abschnitt 2.1 beschriebenen Wahlkategorien (Wahlen erster, zweiter und dritter Ordnung) modelliert werden. Diese Angreifermodelle sind explizit nur zur Veranschaulichung der Unterschiede unterschiedlicher Angreifermodelle geeignet. Für die Bewertung im konkreten Anwendungsszenario sollte deshalb immer ein eigenes, an die jeweilige Situation angepasstes Angreifermodell erstellt werden.

4.9.1 Angreifermodell für Wahlen dritter Ordnung

Für Wahlen dritter Ordnung wird ein schwaches Angreifermodell gewählt. Es wird hierfür angenommen, dass der Angreifer nicht ins System eindringen kann. Folglich besitzt er lediglich die Fähigkeiten K.1 (passives Abhören von Kommunikationskanälen) und K.2 (aktives Manipulieren von Kommunikationskanälen).

Tabelle 9: Bewertung der Wahlverfahren unter Annahme des Angreifermodells für Wahlen dritter Ordnung.

	Verifizierbarkeit		Wahl-geheim-nis	Nicht-Erpress-barkeit	Robust-heit	Benutz-barkeit	
	ind.	uni.				Wahl	Verif.
Verfahren auf Basis von Funktionstrennung und Systemsicherheit							
Estland V1	x	x	QF.2	RA.1,AA.2,SA.1	RI.1	BW.1.1	-
Estland V2	IV.1.2	x	UL.1	AA.2,SA.1	RI.1	BW.1.1	BV.1
Polyas	IV.1.2	KV.1, WV.2, EV.2	UL.1	AA.2	x	BW.1.1	BV.1
Verfahren auf Basis von blinden Signaturen							
Fujioka et al., Sensus 1992-99	IV.1.1	WV.2, EV.2, KV.1	UL.1	x	RI.1	BW.1.3	BV.1
Okamoto 1997	IV.1.1	WV.2,KV.1	QF.2	x	RI.1	BW.1.1	BV.2
Ohkubo 1999	IV.1.1	WV.2, EV.2, KV.1	UL.1	x	RI.1	BW.1.1	BV.1
Herschberg, Durette, Joaquim, Lebre 1997-2004	IV.1.1	WV.2, EV.2, KV.1	UL.1	x	RI.1	BW.1.1	BV.1
Liaw 2004	IV.1.1	KV.1	UL.1	x	RI.1	BW.1.1	BV.1
Verfahren auf Basis verifizierbaren Mischens							
Chaum 1981	IV.1.1	WV.1,KV.1	UL.1	x	x	BW.1.3	BV.1
Park et al. 1993	IV.1.1	WV.1,KV.1	UL.1	x	x	BW.1.1	BV.1
SK 1994	IV.1.1	KV.1	QF.1	x	x	BW.1.1	BV.2
JCJ 2005	IV.2.2	WV.2, EV.2	QF.1	AA.2, RA.1, SA.1	RI.1	BW.1.1	BV.1
Civitas 2008	IV.2.1	WV.2, EV.1, KV.1	QF.1	AA.2, RA.1, SA.1	x	BW.1.1	BV.2
Shirazi 2011	IV.2.1	WV.2, EV.1, KV.1	QF.1	AA.2, RA.1, SA.1	RI.1	BW.1.1	BV.2
Trivitas 2012	IV.1.1	WV.2, EV.1, KV.1	QF.1	AA.2, RA.1, SA.1	x	BW.1.1	BV.2
Verfahren auf Basis von homomorpher Verschlüsselung							
Cohen 1985	IV.1.1	KV.1	X	x	x	BW.1.3	BV.1
Benaloh, Yung 1986	IV.1.1	KV.1	UL.1	x	x	BW.1.3	BV.1
Cohen 1994	IV.1.1	KV.1	UL.1	x	RI.1	BW.1.3	BV.1
Benaloh, Tuinstra 1994	IV.1.1	KV.1	UL.1	x	RI.1	BW.1.3	BV.1
Ellard, Alpert 2003	IV.1.1	KV.1	QF.1	x	RI.1	BW.1.3	BV.1
Iversen 1991	x	WV.2	UL.1	x	x	BW.1.1	x
Kiayias, Yung 2004	IV.1.1	KV.2	UL.1	x	RI.1	BW.1.1	BV.2
Kiayias et al. 2006	x	KV.1	UL.1	x	x	BW.1.1	x
Hirt 2000	IV.2.1	x	QF.1	x	x	BW.1.1	BV.2
Schoenmakers 1999	IV.1.1	KV.1	UL.1	x	x	BW.1.1	BV.2
Damgard, Jurik 2001	IV.2.1	x	QF.1	x	RI.1	BW.1.1	BV.2
Baudron et al. 2001	IV.1.1	KV.1	QF.1	x	RI.1	BW.1.1	BV.1
Acquisti 2004	IV.1.1	EV.1, KV.1	QF.1	AA1, RA.1, SA.1	RI.1	BW.1.3	BV.1
Sako, Kilian 1994	IV.2.1	KV.1	UL.1	x	x	BW.1.1	BV.2
Cramer et al. 1996/97	IV.1.1	KV.1	UL.1	x	RI.1	BW.1.1	BV.2
Verfahren auf Basis von Attribute-based Credentials							
Put et al. 2014	IV1.1	BV.2, EV.2, KV.1	UL.1	x	x	BW.1.1	BV.1
Verfahren auf Basis von Blockchain							
Lee et al. 2016	IV.1.1	KV.1, WV.2, EV.2	UL.1	x	x	BW.1.1	BV.1
Barnes et al. 2016	IV.1.1	KV.1, WV.2, EV.2	UL.2	x	x	BW.2.3	BV.1
Ayed 2017	IV.1.1	KV.1	UL.1	x	RI.1	BW.1.1	BV.1
Bistarelli et al. 2017	IV.1.1	KV.1, WV.2, EV.2	UL.1	x	x	BW.1.1	BV.1
Liu, Wang 2017	IV.1.1	KV.1, WV.2, EV.2	UL.1	x	x	BW.1.1	BV.1
Cruz, Kaji 2017	IV.1.1	KV.1, WV.2, EV.2	UL.1	x	x	BW.2.2	BV.1
Zhao, Chan 2016	IV.1.1	KV.1	UL.1	x	x	BW.1.1	BV.1

Verfahren auf Basis von Code Voting							
Helbach 2007	IV.2.2	x	UL.1	SA.1,RA.1	x	BW.2.2	BV.1
VeryVote 2009	IV.1.1	WV.1, EV.1, KV.1	UL.1	x	RI.1	BW.2.2	BV.2
EVIV 2013	IV.1.1	WV.1, EV.1, KV.1	UL.1	x	RI.1	BW.2.2	BV.2
PGD 2009	IV.2.1	WV.2, EV.1, KV.1	QF.2	x	RI.1	BW.2.2	BV.2
PUD 2013	IV.2.1	WV.2, EV.2, KV.1	QF.2	x	x	BW.2.2	BV.2
PUD 2014	IV.2.1	WV.2, EV.2, KV.1	QF.2	x	RI.1	BW.2.2	BV.2
Hybride Verfahren							
Du-Vote 2015	IV.1.1	WV.1, EV.1, KV.1	QF.1	x	x	BW.3.2	BV.2

4.9.2 Angreifermodell für Wahlen zweiter Ordnung

Für Wahlen zweiter Ordnung werden dem Angreifer ebenfalls die Fähigkeiten K.1 (passives Abhören von Kommunikationskanälen) und K.2 (aktives Manipulieren von Kommunikationskanälen) sowie zusätzlich S.1 (Einfügen von Nachrichten auf dem Bulletin Board) und S.3 (Kontrolle über einige, jedoch nicht alle Wahlserver) zugewiesen.

Tabelle 10: Bewertung der Wahlverfahren unter Annahme des Angreifermodells für Wahlen zweiter Ordnung.

	Verifizierbarkeit		**Wahl-geheim-nis**	**Nicht-Erpress-barkeit**	**Robust-heit**	**Benutz-barkeit**	
	ind.	**uni.**				**Wahl**	**Verif.**
Verfahren auf Basis von Funktionstrennung und Systemsicherheit							
Estland V1	x	x	x	x	RI.1	BW.1.1	-
Estland V2	IV.1.2	x	x	x	RI.1	BW.1.1	BV.1
Polyas	IV.1.2	KV.1	x	AA.2	x	BW.1.1	BV.1
Verfahren auf Basis von blinden Signaturen							
Fujioka et al., Sensus 1992-99	IV.1.1	KV.1	UL.1	x	RI.1	BW.1.3	BV.1
Okamoto 1997	IV.1.1	KV.1	UL.1	x	RI.1	BW.1.1	BV.2
Ohkubo 1999	IV.1.1	KV.1	UL.1	x	RI.1	BW.1.1	BV.1
Herschberg, Durette, Joaquim, Lebre 1997-2004	IV.1.1	KV.1	UL.2	x	RI.1	BW.1.1	BV.1
Liaw 2004	IV.1.1	KV.1	UL.2	x	RI.1	BW.1.1	BV.1
Verfahren auf Basis verifizierbaren Mischens							
Chaum 1981	IV.1.1	KV.1, WV.1	UL.1	x	x	BW.1.3	BV.1
Park et al. 1993	IV.1.1	KV.1, WV.1	UL.1	x	x	BW.1.1	BV.1
SK 1994	IV.1.1	KV.1	QF.1	x	x	BW.1.1	BV.2
JCJ 2005	IV.2.2	x	QF.1	AA.2, RA.1, SA.1	RI.1	BW.1.1	BV.1
Civitas 2008	IV.2.1	EV.1,KV.1	QF.1	AA.2, RA.1, SA.1	x	BW.1.1	BV.2
Shirazi 2011	IV.2.1	EV.1, KV.1	QF.1	AA.2, RA.1, SA.1	RI.1	BW.1.1	BV.2
Trivitas 2012	IV.1.1	EV.1, KV.1	QF.1	AA.2, RA.1, SA.1	x	BW.1.1	BV.2
Verfahren auf Basis von homomorpher Verschlüsselung							
Cohen 1985	IV.1.1	KV.1	x	x	x	BW.1.3	BV.1
Benaloh, Yung 1986	IV.1.1	KV.1	UL.1	x	x	BW.1.3	BV.1
Cohen 1994	IV.1.1	KV.1	UL.1	x	RI.1	BW.1.3	BV.1
Benaloh, Tuinstra 1994	IV.1.1	KV.1	UL.1	x	RI.1	BW.1.3	BV.1
Ellard, Alpert 2003	IV.1.1	KV.1	QF.1	x	RI.1	BW.1.3	BV.1
Iversen 1991	x	WV.2	UL.1	x	RI.1	BW.1.1	x
Kiayias, Yung 2004	IV.1.1	KV.2	UL.1	x	RI.1	BW.1.1	BV.2
Kiayias et al. 2006	x	KV.1	UL.1	x	x	BW.1.1	x
Hirt 2000	IV.2.1	x	QF.1	x	x	BW.1.1	BV.2

Schoenmakers 1999	IV.1.1	KV.1	UL.1	x	x	BW.1.1	BV.2
Damgard, Jurik 2001	IV.2.1	x	QF.1	x	RI.1	BW.1.1	BV.2
Baudron et al. 2001	IV.1.1	KV.1	QF.1	x	RI.1	BW.1.1	BV.1
Acquisti 2004	IV.1.1	EV.1, KV.1	QF.1	AA1, RA.1, SA.1	RI.1	BW.1.3	BV.1
Sako, Kilian 1994	IV.2.1	KV.1	UL.1	x	x	BW.1.1	BV.2
Cramer et al. 1996/97	IV.1.1	KV.1	UL.1	x	RI.1	BW.1.1	BV.2
Verfahren auf Basis von Attribute-based Credentials							
Put et al. 2014	IV1.1	BV.2, EV.2, KV.1	UL.1	x	x	BW.1.1	BV.1
Verfahren auf Basis von Blockchain							
Lee et al. 2016	IV.1.1	KV.1, WV.2, EV.2	UL.1	x	x	BW.1.1	BV.1
Barnes et al. 2016	IV.1.1	KV.1, WV.2, EV.2	UL.2	x	x	BW.2.3	BV.1
Ayed 2017	IV.1.1	KV.1	UL.1	x	RI.1	BW.1.1	BV.1
Bistarelli et al. 2017	IV.1.1	KV.1, WV.2, EV.2	UL.1	x	x	BW.1.1	BV.1
Liu, Wang 2017	IV.1.1	KV.1, WV.2, EV.2	UL.1	x	x	BW.1.1	BV.1
Cruz, Kaji 2017	IV.1.1	KV.1	UL.1	x	x	BW.2.2	BV.1
Zhao, Chan 2016	IV.1.1	KV.1	UL.1	x	x	BW.1.1	BV.1
Verfahren auf Basis von Code Voting							
Helbach 2007	IV.2.2	x	UL.1	x	x	BW.2.2	BV.1
VeryVote 2009	IV.1.1	WV.1, EV.1, KV.1	x	x	RI.1	BW.2.2	BV.2
EVIV 2013	IV.1.1	WV.1, EV.1, KV.1	UL.1	x	RI.1	BW.2.2	BV.2
PGD 2009	IV.2.1	EV.1, KV.1	QF.2	x	RI.1	BW.2.2	BV.2
PUD 2013	IV.2.1	KV.1	x	x	x	BW.2.2	BV.2
PUD 2014	IV.2.1	KV.1	x	x	RI.1	BW.2.2	BV.2
Hybride Verfahren							
Du-Vote 2015	IV.1.1	WV.1, EV.1, KV.1	QF.2	x	x	BW.3.2	BV.2

4.9.3 Angreifermodell für Wahlen erster Ordnung

Bei Wahlen erster Ordnung wird unterschieden, ob der Angreifer Zugriff auf die Produktion der Hilfsmittel besitzt oder nicht. In der Praxis wäre diese Unterscheidung z. B. relevant, wenn der die Wahl durchführende Staat die Produktion der Hilfsmittel nicht überwachen lassen bzw. die Hilfsmittel sogar im Ausland und damit komplett außerhalb seiner Kontrolle einkaufen würde.

Hat der Angreifer keinen Zugriff auf die Hilfsmittel, wird davon ausgegangen, dass der Angreifer die folgenden Fähigkeiten besitzt: K.1 (passives Abhören von Kommunikationskanälen), K.2 (aktives Manipulieren von Kommunikationskanälen), S.1 (Einfügen von Nachrichten auf dem Bulletin Board), S.2 (Kontrolle über den Wahlcomputer) , S.4 (Kontrolle über alle Server). Wobei jedoch weiterhin angenommen wird, dass Server und Wahlcomputer nicht zusammenarbeiten.

Tabelle 11: Bewertung sämtlicher Wahlverfahren unter Annahme des Angreifermodells für Wahlen erster Ordnung ohne die Fähigkeit Hilfsmittel zu manipulieren.

	Verifizierbarkeit		**Wahl-geheim-nis**	**Nicht-Erpress-barkeit**	**Robust-heit**	**Benutz-barkeit**	
	ind.	**uni.**				**Wahl**	**Verif.**
Verfahren auf Basis von Funktionstrennung und Systemsicherheit							
Estland V1	x	x	x	x	RI.1	BW.1.1	-
Estland V2	IV.1.2	x	x	x	RI.1	BW.1.1	BV.1
Polyas	IV.1.2	KV.1	x	x	x	BW.1.1	BV.1
Verfahren auf Basis von blinden Signaturen							
Fujioka et al., Sensus 1992-99	IV.1.1	KV.1	x	x	RI.1	BW.1.3	BV.1
Okamoto 1997	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.2
Ohkubo 1999	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1

Herschberg, Durette, Joaquim, Lebre 1997-2004	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Liaw 2004	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Verfahren auf Basis verifizierbaren Mischens							
Chaum 1981	IV.1.1	WV.1, KV.1	x	x	x	BW.1.3	BV.1
Park et al. 1993	IV.1.1	WV.1, KV.1	x	x	x	BW.1.1	BV.1
SK 1994	IV.1.1	KV.1	x	x	x	BW.1.1	BV.2
JCJ 2005	IV.2.2	x	x	x	RI.1	BW.1.1	BV.1
Civitas 2008	IV.2.1	EV.1,KV.1	x	x	x	BW.1.1	BV.2
Shirazi 2011	IV.2.1	EV.1, KV.1	x	x	RI.1	BW.1.1	BV.2
Trivitas 2012	IV.1.1	EV.1, KV.1	x	x	x	BW.1.1	BV.2
Verfahren auf Basis von homomorpher Verschlüsselung							
Cohen 1985	IV.1.1	KV.1	x	x	x	BW.1.3	BV.1
Benaloh, Yung 1986	IV.1.1	KV.1	x	x	x	BW.1.3	BV.1
Cohen 1994	IV.1.1	KV.1	x	x	RI.1	BW.1.3	BV.1
Benaloh, Tuinstra 1994	IV.1.1	KV.1	x	x	RI.1	BW.1.3	BV.1
Ellard, Alpert 2003	IV.1.1	KV.1	x	x	RI.1	BW.1.3	BV.1
Iversen 1991	x	WV.2, EV.2	x	x	RI.1	BW.1.1	x
Kiayias, Yung 2004	IV.1.1	KV.2	x	x	RI.1	BW.1.1	BV.2
Kiayias et al. 2006	x	KV.1	x	x	x	BW.1.1	x
Hirt 2000	IV.2.1	x	x	x	x	BW.1.1	BV.2
Schoenmakers 1999	IV.1.1	KV.1	x	x	x	BW.1.1	BV.2
Damgard, Jurik 2001	IV.2.1	x	x	x	RI.1	BW.1.1	BV.2
Baudron et al. 2001	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Acquisti 2004	IV.1.1	EV.1, KV.1	x	x	RI.1	BW.1.3	BV.1
Sako, Kilian 1994	IV.2.1	KV.1	x	x	x	BW.1.1	BV.2
Cramer et al. 1996/97	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.2
Verfahren auf Basis von Attribute-based Credentials							
Put et al. 2014	IV1.1	x	x	x	x	BW.1.1	BV.1
Verfahren auf Basis von Blockchain							
Lee et al. 2016	IV.1.1	KV.1	x	x	x	BW.1.1	BV.1
Barnes et al. 2016	IV.1.1	KV.1	x	x	x	BW.2.3	BV.1
Ayed 2017	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Bistarelli et al. 2017	IV.1.1	KV.1	x	x	x	BW.1.1	BV.1
Liu, Wang 2017	IV.1.1	KV.1	x	x	x	BW.1.1	BV.1
Cruz, Kaji 2017	IV.1.1	KV.1	x	x	x	BW.2.2	BV.1
Zhao, Chan 2016	IV.1.1	KV.1	x	x	x	BW.1.1	BV.1
Verfahren auf Basis von Code Voting							
Helbach 2007	IV.2.2	x	x	x	x	BW.2.2	BV.1
VeryVote 2009	IV.1.1	WV.1, EV.1, KV.1	x	x	RI.1	BW.2.2	BV.2
EVIV 2013	IV.1.1	WV.1, EV.1, KV.1	x	x	RI.1	BW.2.2	BV.2
PGD 2009	IV.2.1	EV.1, KV.1	QF.2	x	RI.1	BW.2.2	BV.2
PUD 2013	IV.2.1	KV.1	x	x	x	BW.2.2	BV.2
PUD 2014	IV.2.1	KV.1	x	x	RI.1	BW.2.2	BV.2
Hybride Verfahren							
Du-Vote 2015	IV.1.1	WV.1, EV.1, KV.1	QF.2	x	x	BW.3.2	BV.2

Besitzt der Angreifer die Fähigkeit, die Hilfsmittel zu manipulieren, so kommen zu den oben genannten Fähigkeiten noch H.1 (unverändertes Kopieren von Hilfsmitteln außerhalb des Userspace), H.2 (Manipulation von Hilfsmitteln ohne Vervielfältigung außerhalb des Userspace), H.3 (Herstellung / Fälschung eigener Hilfsmittel und Zuführung zum System).

Tabelle 12: Bewertung sämtlicher Wahlverfahren unter Annahme des Angreifermodells für Wahlen erster Ordnung mit der Fähigkeit Hilfsmittel zu manipulieren.

	Verifizierbarkeit		Wahl-geheim-nis	Nicht-Erpress-barkeit	Robust-heit	Benutz-barkeit	
	ind.	uni.				Wahl	Verif.
Verfahren auf Basis von Funktionstrennung und Systemsicherheit							
Estland V1	x	x	x	x	RI.1	BW.1.1	-
Estland V2	IV.1.2	x	x	x	RI.1	BW.1.1	BV.1
Polyas	IV.1.2	KV.1	x	x	x	BW.1.1	BV.1
Verfahren auf Basis von blinden Signaturen							
Fujioka et al., Sensus 1992-99	IV.1.1	KV.1	x	x	RI.1	BW.1.3	BV.1
Okamoto 1997	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.2
Ohkubo 1999	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Herschberg, Durette, Joaquim, Lebre 1997-2004	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Liaw 2004	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Verfahren auf Basis verifizierbaren Mischens							
Chaum 1981	IV.1.1	WV.1, KV.1	x	x	x	BW.1.3	BV.1
Park et al. 1993	IV.1.1	WV.1, KV.1	x	x	x	BW.1.1	BV.1
SK 1994	IV.1.1	KV.1	x	x	x	BW.1.1	BV.2
JCJ 2005	IV.2.2	x	x	x	RI.1	BW.1.1	BV.1
Civitas 2008	IV.2.1	EV.1, KV.1	x	x	x	BW.1.1	BV.2
Shirazi 2011	IV.2.1	EV.1, KV.1	x	x	RI.1	BW.1.1	BV.2
Trivitas 2012	IV.1.1	WV.2, EV.1, KV.1	x	x	x	BW.1.1	BV.2
Verfahren auf Basis von homomorpher Verschlüsselung							
Cohen 1985	IV.1.1	KV.1	x	x	x	BW.1.3	BV.1
Benaloh, Yung 1986	IV.1.1	KV.1	x	x	x	BW.1.3	BV.1
Cohen 1994	IV.1.1	KV.1	x	x	RI.1	BW.1.3	BV.1
Benaloh, Tuinstra 1994	IV.1.1	KV.1	x	x	RI.1	BW.1.3	BV.1
Ellard, Alpert 2003	IV.1.1	KV.1	x	x	RI.1	BW.1.3	BV.1
Iversen 1991	x	WV.2, EV.2	x	x	RI.1	BW.1.1	x
Kiayias, Yung 2004	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.2
Kiayias et al. 2006	x	KV.1	x	x	x	BW.1.1	x
Hirt 2000	IV.2.1	x	x	x	x	BW.1.1	BV.2
Schoenmakers 1999	IV.1.1	KV.1	x	x	x	BW.1.1	BV.2
Damgard, Jurik 2001	IV.2.1	x	x	x	RI.1	BW.1.1	BV.2
Baudron et al. 2001	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Acquisti 2004	IV.1.1	EV.1, KV.1	x	x	RI.1	BW.1.3	BV.1
Sako, Kilian 1994	IV.2.1	KV.1	x	x	x	BW.1.1	BV.2
Cramer et al. 1996/97	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.2
Verfahren auf Basis von Attribute-based Credentials							
Put et al. 2014	IV1.1	x	x	x	x	BW.1.1	BV.1
Verfahren auf Basis von Blockchain							
Lee et al. 2016	IV.1.1	KV.1	x	x	x	BW.1.1	BV.1
Barnes et al. 2016	IV.1.1	KV.1	x	x	x	BW.2.3	BV.1
Ayed 2017	IV.1.1	KV.1	x	x	RI.1	BW.1.1	BV.1
Bistarelli et al. 2017	IV.1.1	KV.1	x	x	x	BW.1.1	BV.1
Liu, Wang 2017	IV.1.1	KV.1	x	x	x	BW.1.1	BV.1
Cruz, Kaji 2017	IV.1.1	KV.1	x	x	x	BW.2.2	BV.1
Zhao, Chan 2016	IV.1.1	KV.1	x	x	x	BW.1.1	BV.1

Verfahren auf Basis von Code Voting							
Helbach 2007	IV.2.2	x	x	x	x	BW.2.2	BV.1
VeryVote 2009	IV.1.1	WV.1, EV.1, KV.1	x	x	RI.1	BW.2.2	BV.2
EVIV 2013	IV.1.1	WV.1, EV.1, KV.1	x	x	RI.1	BW.2.2	BV.2
PGD 2009	IV.2.1	EV.1, KV.1	x	x	RI.1	BW.2.2	BV.2
PUD 2013	IV.2.1	KV.1	x	x	x	BW.2.2	BV.2
PUD 2014	IV.2.1	KV.1	x	x	RI.1	BW.2.2	BV.2
Hybride Verfahren							
Du-Vote 2015	IV.1.1	WV.1, EV.1, KV.1	x	x	x	BW.3.2	BV.2

5 ANALYSE AUSGEWÄHLTER VERFAHREN

Dieses Kapitel bietet detailliertere Analysen zweier bereits praktisch eingesetzter Verfahren, über die bereits ausreichend öffentlich zugängliche Literatur verfügbar ist. Namentlich handelt es sich um das offizielle estnische Internetwahlsystem sowie Polyas, welches unter anderem bereits von verschiedenen Wissenschaftsorganisationen zur Wahl ihrer Gremien eingesetzt wurde. Außerdem wird zusätzlich das Internetwahlprotokoll Du-Vote detaillierter analysiert, da dieses sich während der bisherigen Sichtung der einschlägigen Literatur als besonders vielversprechend herausgestellt hat. Die jeweils am Ende des Abschnitts abschließende Sicherheitsbewertung der drei Verfahren basiert auf der in Kapitel 3 vorgestellten Bewertungsmethodik. Es wird bei der Analyse jeweils von Angreifer-Fähigkeiten ausgegangen, die dem Angreifermodell für Wahlen erster Ordnung entsprechen, da auf diese Art und Weise eine möglichst umfängliche Diskussion der Sicherheitsanforderungen möglich ist.

5.1 ESTNISCHES WAHLSYSTEM

Estland war das erste Land, welches ab 2005 seinen Bürgern bei einer politischen Wahl landesweit die Möglichkeit einräumte, über das Internet abzustimmen. Das eingesetzte System beruht zum großen Teil auf Estlands elektronischem Personalausweis sowie der staatseigenen Public-Key-Infrastruktur. Im August 2017 wurde bekannt, dass genau dieser Personalausweis, das Herzstück der estnischen Digitalisierungsstrategie, eine gravierende Sicherheitslücke aufweist [149]. Die *Return of Coppersmith's Attack (ROCA)* getaufte Schwachstelle beruht auf einem Fehler in der Generierung von Primzahlen im Zuge der Schlüsselgenerierung von RSA und kann durch einen Faktorisierungsangriff ausgenutzt werden, der es Angreifern ermöglicht, den privaten Teil des Schlüsselpaars aus dem öffentlichen zu berechnen [150, 151]. Betroffen sind neben Trusted-Platform-Modulen des Chipherstellers Infineon auch *IDPrime.net* Smartcards des Schweizer Herstellers Gemalto, auf denen der Großteil der estnischen Personalausweise beruht. Seit November 2017 können die Zertifikate betroffener Ausweise aktualisiert werden [152].

Neben dem I-Voting besitzen die estnischen Bürger noch weitere Möglichkeiten, ihre Stimme abzugeben. Die am häufigsten genutzte Möglichkeit ist nach wie vor, am Wahltag an der papierbasierten Wahl im eigenen Wahlbezirk teilzunehmen. Für wen das nicht möglich ist, gibt es mehrere Möglichkeiten, trotzdem an der Wahl teilzunehmen. Diese werden unter dem Begriff „Advance Voting" zusammengefasst. Außerdem gibt es für estnische Bürger, die im Ausland leben, die Möglichkeit in einer Auslandsvertretung oder per Brief (Abroad Voting) an der Wahl teilzunehmen. Für die Internetwahl ist das Estonian National Electoral Committee (NEC) zuständig. In diesem Kapitel soll das eingesetzte Wahlverfahren erklärt werden.

Dazu muss man erst den generellen Ablauf der Wahl in Estland verstehen. Die estnische Wahl geht über einen Zeitraum von zehn Tagen. Wobei die Stimmabgabe über das Internet innerhalb der ersten sieben Tage, das heißt von Tag eins bis Tag sieben, erfolgen kann. In den darauffolgenden Tagen acht und neun der Wahlperiode darf nicht gewählt werden und zum Abschluss findet am zehnten Tag eine papierbasierte Präsenzwahl statt. Während der I-Voting Periode kann ein Wähler seine Stimme beliebig oft abgeben, wobei immer nur die zuletzt abgegebene Stimme in das Ergebnis miteinfließt (Vote Updating). In den anschließenden zwei Tagen wird eine Liste der Wähler erstellt, welche über das Internet abgestimmt haben, und an die entsprechenden Wahlbüros gesendet. Am Tag der papierbasierten Präsenzwahl kann jeder, der seine Stimme über das Internet abgegeben hat, erneut wählen. Hierdurch wird die über das Internet abgegebene Stimme ungültig. Dazu muss das

Wahlbüro die Liste der Internetwähler an das NEC senden, welches die entsprechenden Stimmen aus der elektronischen Wahlurne entfernt [153–155].

5.1.1 Generelles Konzept des estnischen I-Voting

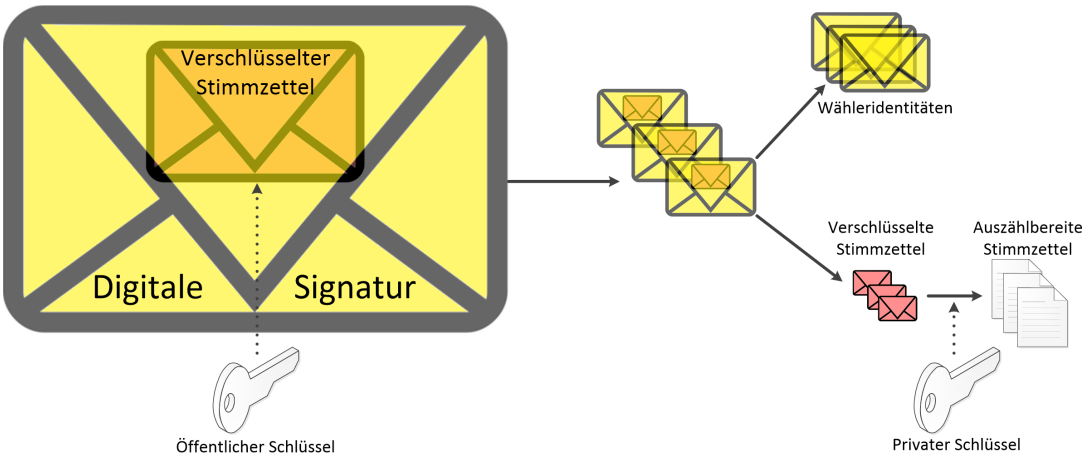


Abbildung 12: Konzept des elektronischen Wahlsystems Estlands in Anlehnung an [153].

Das estnische I-Voting System ist an die Briefwahl, welche auch im Advance Voting oder Abroad Voting eingesetzt wird, angelehnt[1]. Dazu identifiziert sich der Wähler im Wahllokal und erhält einen Stimmzettel. Anschließend füllt der Wähler den Stimmzettel aus und steckt ihn in einen nicht beschrifteten Umschlag. Dieser wird dann zusammen mit den Daten des Wählers einem zweiten Umschlag beigefügt. Der Umschlag wird zum Wahllokal geschickt, dort geöffnet und der Umschlag mit dem Stimmzettel von den Wählerdaten getrennt. Dann wird der Umschlag mit dem Stimmzettel anonym geöffnet und in die Wahlurne geworfen.

Das I-Voting Protokoll ist im Prinzip äquivalent aufgebaut. Anstatt des inneren Umschlags wird die Stimme asymmetrisch mit dem öffentlichen Schlüssel der Wahlautorität verschlüsselt. Der äußere Umschlag stellt eine Signatur der verschlüsselten Stimme dar, welche mit dem Zertifikat, das sich im Personalausweis des Wählers befindet, unterschrieben wird [153].

5.1.2 Architektur

Das estnische Wahlsystem besteht aus zwei Hauptkomponenten. Zum einen ist das die **I-Voting Client Application (IVCA)**, welche es dem Wähler erlaubt, seine Stimme zu verschlüsseln und zu signieren. Die IVCA ermöglicht es dem Wähler, auf den Plattformen Linux, Windows und MacOS zu wählen. Die IVCA kommuniziert mit dem **I-Voting System (IVS)**, welches für die Sammlung, Speicherung und Auszählung der Stimmen zuständig ist. Das System hat Schnittstellen zur Kandidatenliste sowie zu einer Liste der Wähler (Wählerregister). Um das Wahlgeheimnis zu wahren, besteht Aufgabentrennung zwischen drei Hauptkomponenten des IVS. Die Hauptkomponenten sind Vote Forwarding Server (VFS), Candidate List (CL), Vote Storing Server (VSS) und Vote Counting Server (VCS). Im Folgenden werden die einzelnen Aufgaben der Komponenten genauer beschrieben (siehe dazu auch Abbildung 13):

- Der **Vote Forwarding Server (VFS)** ist dafür zuständig, den Wähler mittels seines Personalausweises zu authentifizieren. Außerdem besitzt er ein Zertifikat, um sich gegenüber dem Wähler zu authentifizieren. Des Weiteren besitzt er Zugriff auf die Kandidatenliste, um diese der IVCA

1 Vergleichbar mit der Briefwahl in Deutschland.

zu übermitteln und anschließend die Stimme von der IVCA entgegen zu nehmen und an den Vote Storing Server (VSS) weiterzuleiten.

- Die **Candidate List (CL)** ist eine Liste, welche jedem Kandidaten eine eindeutige Nummer zuordnet, d. h. eine injektive Abbildung zwischen dem Namen des Kandidaten und einer eindeutigen Nummer.
- Der **Vote Storing Server (VSS)** speichert die vom VFS empfangenen Stimmen ab. Nach der Online-Wahlphase ist der VSS für die Überprüfung der Signaturen der Stimmen sowie die Entfernung von mehrfach abgegebenen Stimmen zuständig. Außerdem anonymisiert der VSS die abgegebenen Stimmen, indem er Signatur und verschlüsselte Stimme trennt. Um die Sicherheit der gespeicherten Stimmen zu erhöhen, befindet sich der VSS hinter einer Firewall, welche nur Verbindungen vom VFS zulässt.
- Der **Vote Counting Server (VCS)** ist für die Auszählung der Stimmen zuständig. Da dies den kritischsten Prozess im Wahlsystem darstellt, ist dies das sich am meisten lohnende Angriffsziel für Angreifer, welche die Wahl manipulieren möchten. Um dies zu verhindern, besitzt der VCS zu keiner Zeit eine Verbindung zum Netzwerk, weshalb die anonymisierten Stimmen mittels DVD an den VCS übertragen werden müssen. Der VCS besitzt ein **Hardware Security Module (HSM)**, auf welchem das Key Pair (ivs_{sk}, ivs_{pk}) für die Ver- und Entschlüsselung der Stimmen (innerer Umschlag) erzeugt wird, welches für den Schutz des Wahlgeheimnisses zuständig ist. Der öffentliche Schlüssel ivs_{pk} wird zusammen mit der IVCA veröffentlicht. Der private Schlüssel ivs_{sk} bleibt im HSM, in welchem alle kritischen Operationen, wie die Generierung des Key Pairs oder die Verwendung des privaten Schlüssels ivs_{sk}, ausgeführt werden. Aufgrund des hohen Sicherheitsrisikos ist das HSM durch ein vier aus sieben Multiparty-Authentication-Protokoll geschützt, sodass mindestens vier der sieben Wahlleiter anwesend sein müssen, um diese kritischen Operationen durchführen zu können.

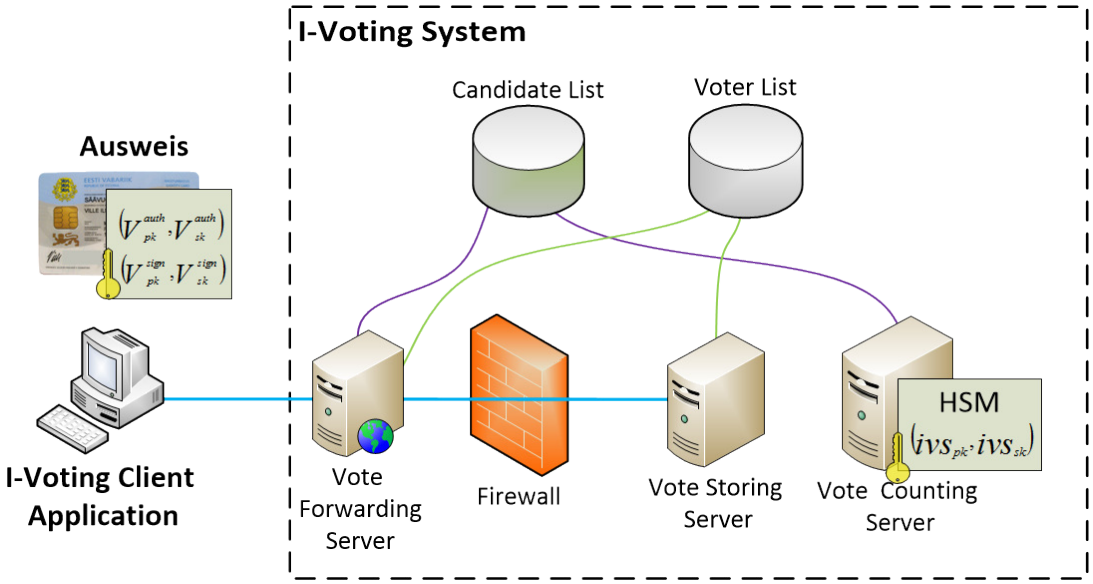


Abbildung 13: Architektur des estnischen Wahlsystems in Anlehnung an [153].

Eine weitere Komponente des estnischen Wahlsystems ist der estnische **Personalausweis**, der entweder eine Smartcard oder eine sogenannte Mobile-ID[2] ist, mit welchem sich der Wähler gegenüber dem VFS authentifizieren kann und seine Stimme signiert. Um die Funktionen zu erfüllen, besitzt der Ausweis zwei RSA Schlüsselpaare. Zum einen das Paar $(V_{sk}^{sign}, V_{pk}^{sign})$, welches die Hauptkomponente

2 Methode, bei der das Smartphone durch eine spezielle SIM-Karte die Funktion der Smartcard übernimmt.

des Zertifikats $CertV^{sign} = (V_{pk}^{sign}, PCI, A, S)$ ist, und dazu dient, Dokumente (im Fall der Wahl die Stimme) zu signieren. Das zweite Schlüsselpaar $(V_{sk}^{auth}, V_{pk}^{auth})$ dient zur Authentifizierung des Besitzers (in diesem Fall des Wählers). Es wird in einem TLS-Zertifikat $CertV^{auth} = (V_{pk}^{auth}, PCI, A, S)$ verwendet, wobei PCI für eine eindeutige Kennung für jeden Bürger, A für verschiedene Attribute und S für die Signatur der ausstellenden Instanz steht. Um einen Verlust der privaten Schlüssel zu vermeiden, werden die kryptografischen Operationen direkt auf dem Ausweis ausgeführt. Außerdem sind die privaten Schlüssel mit einem PIN-Code gesichert [153–155].

5.1.3 Protokoll ohne individuelle Verifizierbarkeit

Das hier beschriebene Protokoll wurde von 2005 bis 2014 für mehrere Parlaments- und Europa-Wahlen eingesetzt. Während dieser Zeit blieb der zu Grunde gelegte Wahlablauf im Kern unverändert.

ABLAUF Der Ablauf gliedert sich in vier Phasen: Vorbereitung, I-Voting, Revocation und die Stimmenauszählung.

VORBEREITUNGSPHASE: Während der Vorbereitungsphase wird der VCS zusammen mit der Kandidatenliste und dem HSM aufgesetzt. Im HSM wird das Schlüsselpaar (ivs_{pk}, ivs_{sk}) erzeugt. Anschließend wird der VFS zusammen mit dem Wählerregister, der Kandidatenliste sowie einem HTTPS-Authentifizierungszertifikat aufgesetzt. Zugleich wird der VSS aufgesetzt sowie eine Firewall, die sicherstellt, dass nur der VFS mit dem VSS kommunizieren kann. Zum Schluss wird die IVCA zusammen mit dem ivs_{pk} und dem VFS Zertifikat auf der Webseite des NEC veröffentlicht. Um eine Manipulation der IVCA zu vermeiden, wird die IVCA vom NEC signiert. Die Fingerprints werden zusammen mit der Download-URL auf der Webseite des NEC sowie in der Zeitung veröffentlicht [155].

I-VOTING PHASE: Für den eigentlichen Wahlvorgang authentifiziert sich der Wähler V mit dem TLS-Zertifikat $CertV^{auth}$ seines Personalausweises gegenüber dem VFS und der VFS authentifiziert sich mit seinem TLS Zertifikat gegenüber der IVCA (bei der Verwendung von Mobile-ID wird die Authentifizierung des Wählers nicht durchgeführt [155]). Der VFS überprüft, ob der Wähler wahlberechtigt ist. Ist alles korrekt, wird eine HTTPS-Verbindung aufgebaut. Dann sendet der VFS die Kandidatenliste CL zusammen mit der Bemerkung, ob schon eine Stimme abgegeben wurde, an die IVCA.

Anschließend kann der Wähler V seine Wahl treffen, indem er einen Kandidaten-Namen aus der auf dem Monitor angezeigten Kandidatenliste auswählt. Die IVCA ermittelt die zugehörige Kandidaten-Nummer c aus der Kandidatenliste CL. Dann wird vom IVCA eine Zufallszahl r gewählt, mit welcher er $b = RSA - OAEP_{enc}(ivs_{pk}, (c, r))$ ausführt und anschließend wird $s_V = SIGN_{RSA}(V_{sk}^{sig}, b)$ berechnet. Nun sendet die IVCA die Nachricht $b_V = (b, s_V, CertV^{sign})$ an den VFS. Der VFS prüft die formale Korrektheit von b_V und überprüft, ob die Signatur s_V von der gleichen Person kommt, welche sich authentifiziert hat, indem sie prüft, ob beide Zertifikate dieselbe PCI haben. Ist die Signatur korrekt, leitet der VFS b_v an den VSS weiter, welcher wiederum die Signatur sowie die Gültigkeit von $CertV^{sign}$ überprüft. Wenn diese Überprüfung korrekt ist, wird b_v gespeichert und eine Meldung, dass die Stimme erfolgreich abgegeben wurde, an die IVCA gesendet, welche die Meldung anzeigt. Außerdem wird die Annahme der Stimme in einer Log-Datei (Log1) auf dem VSS für ein späteres Audit protokolliert, indem die PCI zusammen mit einem SHA1-Hash von b_v in eine Datei geschrieben wird.

Hat der Wähler V bereits eine Stimme abgegeben, nutzt er also Vote-Updating, so muss der Wähler zusätzlich einen Grund r für die Wahlwiederholung angeben. Dieser Grund wird zusammen mit b_v an den VSS gesendet. Besteht die Stimme alle Überprüfungen des VFS und des VSS (siehe oben), wird die alte Stimme auf dem VSS direkt gelöscht. Zu Auditzwecken werden diese Stimmen in einer weiteren Log-Datei (Log2) protokolliert. Dazu wird PCI zusammen mit einem SHA1-Hash von b_v sowie dem angegebenen Grund r für die Wahlwiederholung gespeichert [153–156].

REVOCATION-PHASE: In der Revocation Phase wird dem Wähler nach der I-Voting Phase noch einmal die Möglichkeit gegeben, seine Wahl durch die Teilnahme an der papierbasierten Präsenzwahl zu widerrufen. Dazu wird eine Liste der I-Voter erstellt, indem die Informationen aus den Zertifikaten aller Stimmen um Informationen aus der Wählerliste ergänzt werden. Diese Liste wird dann an die Wahllokale verteilt. Dabei muss auf die Integrität und die Authentizität geachtet werden. Während der Wahl werden auf der Liste alle Wähler abgehakt, die ihr I-Vote widerrufen. Nach der Wahl wird eine Liste mit den Wählern, deren Stimmen gelöscht werden müssen, in den VSS eingegeben, welcher die Stimmen dann entfernt. Hier werden ebenfalls alle gestrichenen Stimmen in Log2 protokolliert [153, 155].

AUSZÄHLUNGSPHASE: Nachdem alle mehrfachen Stimmen entfernt wurden, werden diese nach den zwölf estnischen Wahlbezirken geordnet, indem die PCI des Zertifikats und die Informationen aus der Wählerliste korreliert werden. Anschließend werden die Stimmen anonymisiert, indem der VSS die Signaturen von den verschlüsselten Stimmen trennt. Genauer gesagt, wird die Stimme $b_V = (b, s_V, CertV^{sign})$ in $b = RSA - OAEP_{enc}(ivs_{pk}, (c, r)$ und $s_V = SIGN_{RSA}(V_{sk}^{sig}, b)$ aufgeteilt. Anschließend werden die verschlüsselten und anonymisierten Stimmen b auf DVD gebrannt und an den VCS übermittelt. Die Signaturen s_v werden hingegen auf dem VSS behalten. Alle Stimmen, die an den VCS übermittelt wurden, werden in einer Log-Datei (Log3) in der Form des PCI zusammen mit einem SHA1-Hash von b_v gespeichert. Die Stimmen von der DVD werden vom VCS eingelesen und von diesem entschlüsselt. Genauer gesagt berechnet dieser $RSA - OAEP_{dec}(ivs_{pk}, b))$ und erhält c. Aus c wird mit Hilfe der Kandidatenliste CL der Name des Kandidaten ermittelt und dessen Stimmenzähler um eins erhöht. Gilt $c \notin CL$, ist die Stimme ungültig. Dies wird in einer Log-Datei (Log4) vermerkt. Ist die Stimme hingegen korrekt, so wird dies in einer anderen Log-Datei (Log5) protokolliert [153–155].

Protokollierung und Audits

Um eine Kompromittierung zu verhindern, werden mehrere organisatorische Maßnahmen getroffen, welche alle überwacht werden. So müssen zur Anonymisierung der Stimmen mindestens zwei Wahlleiter sowie ein Auditor anwesend sein. Des Weiteren wird vor der Wahl das Verhalten aller verwendeten Funktionen dokumentiert und während der Wahl werden alle Outputs auf einem Band gespeichert, um nur einige Maßnahmen zu nennen.

Vor allem die Möglichkeit, das Wahlergebnis zu beeinflussen, ist natürlich problematisch, weswegen hier Maßnahmen zur Überprüfung der Integrität der Wahlurnen getroffen werden, um dies zu verhindern. Wie bereits erklärt, werden die Stimmen je nach Aktion, die mit der Stimme ausgeführt wird, in unterschiedlichen Log-Dateien protokolliert. Zur Erinnerung sollen die existierenden Log-Dateien noch einmal kurz beschrieben werden:

- Log_1: Alle Stimmen, die vom VSS angenommen wurden.
- Log_2: Alle Stimmen, die vom VSS abgelehnt wurden.
- Log_3: Alle Stimmen, die vom VSS zum VCS gesendet wurden.
- Log_4: Alle Stimmen, die vom VCS abgelehnt wurden.
- Log_5: Alle Stimmen, die in das Ergebnis eingeflossen sind.

Dies soll die Überprüfung der Wahlurne ermöglichen. Die Wahlautorität kann nun überprüfen, ob folgende Annahme gilt:

$$LOG_1 = LOG_2 \cup LOG_3 \quad \wedge \quad LOG_3 = LOG_4 \cup LOG_5$$

Gilt dies nicht, wurden unerlaubterweise Stimmen der Wahlurne hinzugefügt oder entfernt. Allerdings folgt die Umkehrung nicht, da es keinen Beweis gibt, dass die Log-Dateien nicht zusammen mit der Software manipuliert wurden, oder die Software bereits manipulierte Logeinträge generiert.

Angriffe

In der Literatur werden bereits einige Angriffe beschrieben, welche alle von Außerhalb des IVS ausgeführt werden. In diesem Abschnitt werden diese Angriffe auf das System beschrieben und teilweise mögliche Abwehrmechanismen vorgeschlagen.

STUDENT-ANGRIFF Der Student-Angriff erhielt seinen Namen, da er während der Wahl 2011 von einem anonymen Studenten entdeckt und als Proof-of-Concept ausgeführt wurde. Der Angriff kompromittiert sowohl die Stimmenintegrität als auch das Wahlgeheimnis. Die Idee hinter dem Angriff ist eine Malware zu entwickeln, welche die Ausgabe der IVCA manipuliert und dem Wähler einen anderen Kandidaten c_1 unterschiebt, indem der Kandidat c_2, den der Wähler in die IVCA ausgewählt hat, entsprechend angezeigt wird, aber der Kandidat c_1 zur Verschlüsselung und Signierung intern in der IVCA weitergegeben wird. Da der Wähler vom IVS zwar eine Meldung erhält, dass seine Stimme auf dem VSS eingegangen ist, aber nicht verifizieren kann, was der VSS tatsächlich empfangen hat, bleibt er im Glauben, dass alles in Ordnung ist.

Eine derartige Malware wurde von diesem Studenten als Proof-of-Concept in der Skriptsprache *AutoIT* implementiert. Diese Malware legt eine Fake-IVCA über die eigentliche Applikation, welche aus einem einfachen Screenshot der IVCA besteht. Außerdem benutzt die Malware Optical Character Recognition (OCR), um die Eingaben des Wählers abzufangen, welche anschließend mittels Mouse-Events an die IVCA weitergegeben werden. Um Statusänderungen in der Fake-IVCA zu erkennen, benutzt er die *AutoIT PixelChecksum* Funktion. Falls der Wähler einen anderen Kandidaten auswählt, wird ein Screenshot der Kandidatenliste mit dem ausgewählten Kandidaten generiert und angezeigt. Wenn der Wähler dann den Button zum Signieren und Verschlüsseln seiner Wahl betätigt, wird in der Orginal-IVCA verdeckt mittels Mouse-Event ein anderer Kandidat ausgewählt. Anschließend werden die Orginaldaten mittels OCR ermittelt und in eine Log-Datei geschrieben

Die oben beschriebene Malware ist für einen großflächigen Angriff nicht geeignet. Allerdings ist es möglich, diesen Angriff mittels Malware so durchzuführen, dass das Verhalten der IVCA direkt entsprechend manipuliert wird. Als Beispiel für eine solche Malware soll der Ghost-Clicking-Angriff dienen, welcher in Abbildung 5.1.4 beschrieben wird. Die für den Angriff verantwortliche Schwachstelle im Protokoll ist, dass der Wähler die Stimme, die beim IVS eingegangen ist, nicht verifizieren kann [155].

MANIPULATING CANDIDATE LIST ATTACK Dieser Angriff basiert darauf, dass es dem Angreifer gelingt, sich als Man-in-the-Middle in die HTTPS-Verbindung einzuklinken. Dies ist dann möglich, wenn sich der Client nicht gegenüber dem Server authentifizieren muss, wie dies bei der Verwendung von Mobile-ID geschieht. Eine weitere Möglichkeit ist, dass der IVCA unter dem Betriebssystem Windows[3] ausgeführt wird und der Angreifer ein gültiges Zertifikat für das IVS besitzt, welches von einer Root-CA signiert ist, die sich im Zertifikats-Store des Systems befindet. Diese Voraussetzungen sind normalerweise schwer zu erfüllen, allerdings zeigt dies eine fundamentale Schwäche des Wahlsystems und ist für einen Angreifer mit vielen Ressourcen nicht unrealistisch, da zum einen schon mehrere Root-CAs gehackt und so gefälschte Zertifikate für bereits existierende Seiten ausgestellt wurden [157]. Zum anderen muss bedacht werden, dass auch ausländische Regierungen und Geheimdienste, die ein Interesse haben könnten, Einfluss auf die Wahl zu nehmen, oft Kontrolle über eine Root-CA besitzen oder davon auszugehen ist, dass sich diese leicht ein entsprechendes Zertifikat besorgen können.

Angenommen dem Angreifer gelingt es, sich durch die oben genannten Maßnahmen als Man-in-the-Middle in die Verbindung zwischen IVCA und VFS zu setzen, so kann er offensichtlich nicht das Wahlgeheimnis brechen, da dieses durch die Verschlüsselung der Stimme mit dem ivs_{pk} gesichert ist. Auch die Integrität der Stimme ist auf den ersten Blick durch die Signatur des Wählers gesichert. Das eigentliche Problem besteht darin, dass der Angreifer die Möglichkeit besitzt, die Kandidatenliste zu manipulieren, was möglich ist, da die Integrität der Kandidatenlisten nur über die Integrität

3 Es ist auch möglich, dass der Angriff unter Linux oder MacOS funktioniert, dies wurde in dem entsprechenden Paper jedoch [155] nicht untersucht.

der HTTPS-Verbindung gesichert wird. Wie bereits erwähnt, besteht die Wählerliste aus einer Zuordnung zwischen dem Namen des Kandidaten und einer Kandidaten-Nummer. Durch Änderung dieser Zuordnung kann der Angreifer den Wähler dazu zwingen, ungültig zu Wählen, indem er von allen Kandidaten die Kandidatennummern durch Nummern ersetzt, die in der originalen Kandidatenliste nicht vorkommen. Außerdem kann der Angreifer den Wähler dazu zwingen, die Wahl des Angreifers zu treffen, indem er alle Kandidatennummern auf die gleiche Nummer setzt, welche der Wahl des Angreifers entspricht. Etwas weniger auffällig kann der Angreifer die Nummer des Kandidaten, der in den Umfragen führt, durch die Kandidatennummer seines präferierten Kandidaten und den Rest mit ungültigen Nummern ersetzen oder so permutieren, dass mit einer hohen Wahrscheinlichkeit der Kandidat des Angreifers die Wahl gewinnt.

Abhilfe ist hier zu schaffen, indem die Integrität der Wählerliste nicht über den Zertifikatsspeicher des Betriebssystems, sondern durch ein Zertifikat des IVS gewährleistet wird. Dies ist zu realisieren, indem auf dem VCS ein zusätzliches Schlüsselpaar für das Signieren der Wählerliste erstellt und mit diesem die Wählerliste vom VCS signiert wird. Offensichtlich muss der Schlüssel zur Verifikation der Wählerliste zusammen mit der IVCA ausgeliefert werden, sodass die IVCA die Signatur der Wählerliste prüfen kann. Ein solches Vorgehen führt auch zu einer Verbesserung der Sicherheit innerhalb des IVS, da die Wählerliste nur vom VCS verändert werden kann [155].

5.1.4 Protokoll mit individueller Verifizierbarkeit vor der Auszählung (ab 2013)

Um die Manipulationsmöglichkeiten der Internet Voting Client Application (IVCA) zu verringern, welche aufgrund der im Unterabschnitt 5.1.3 beschriebenen Angriffszenarien bestehen, wurde der Wahlablauf so erweitert, dass der Wähler überprüfen kann, ob seine Stimme korrekt beim IVS eingegangen ist. Allerdings sei erwähnt, dass diese Überprüfung nicht der Anforderung individueller Verifizierbarkeit genügt, da der Wähler nicht überprüfen kann, ob seine Stimme tatsächlich in das Wahlergebnis eingeflossen ist. Durch diese Maßnahme ist der Wähler in der Lage, die Arbeitsweise der IVCA zu verifizieren, wodurch er eine Kompromittierung des Clients mit hoher Wahrscheinlichkeit feststellen kann. Da die Verifizierung nur nach der eigentlichen Stimmabgabe erfolgen kann, wird die Verifizierung über einen zweiten, unabhängigen Kanal, dem sogenannten Post-Channel, durchgeführt. Um die Benutzbarkeit zu gewährleisten, wird für den Post-Channel ebenfalls das Internet verwendet. Um zu verhindern, dass die Verifikation auf dem gleichen Computer stattfindet, auf dem auch gewählt wurde, ist es notwendig, dass die Verifikation über ein zweites Gerät erfolgt. Dieses zweite Gerät soll ein Smartphone oder ein Tablet sein. Dies soll dadurch sichergestellt werden, dass zur Verifikation nur Apps für mobile Endgeräte existieren.

Die eigentliche Verifizierung basiert darauf, dass die Stimme zusammen mit einem zufälligen Wert verschlüsselt wird. Kennt man diesen zufälligen Wert r und die Stimme b, so kann man beide Werte verschlüsseln und überprüfen, ob die verschlüsselte Stimme s_V vom Server mit der selbst erstellten Stimme übereinstimmt.

Der modifizierte Wahlablauf wird in Abbildung 14 schematisch dargestellt und im Folgenden detailliert beschrieben. Der Wähler und das IVS authentifizieren sich gegenseitig mit ihren Zertifikaten. Anschließend erhält der Wähler die Kandidatenliste CL und trifft seine Wahl, indem er $b = RSA - OAEP_{enc}(ivs_{pub}, (c, r))$ berechnet, dann seine Wahl signiert $s_V = SIGN_{RSA}(V_{sk}^{sig}, b)$ und s_V an VFS sendet.

Bis hier unterscheidet sich der Ablauf nicht von dem in Unterabschnitt 5.1.3 vorgestellten Protokoll. Anschließend wird die Stimme s_V zusammen mit einer Wahlreferenznummer vr auf dem VSS gespeichert. Die Wahlreferenznummer dient der späteren eindeutigen Identifikation der Stimme. Der VSS gibt bei erfolgreichem Speichern vr an den VFS zurück, welcher vr an die IVCA weiterleitet. Die IVCA gibt vr zusammen mit r auf dem Bildschirm aus. Alle diese Schritte werden mittels der IVCA auf dem Wahlcomputer durchgeführt. Um die Wahl zu verifizieren, muss der Wähler r und vr auf sein Mobilgerät übertragen, dieses baut eine HTTPS-Verbindung zum VFS auf und lädt sich seine Stimme b, welche mittels vr identifiziert wird, zusammen mit der Kandidatenliste CL herunter.

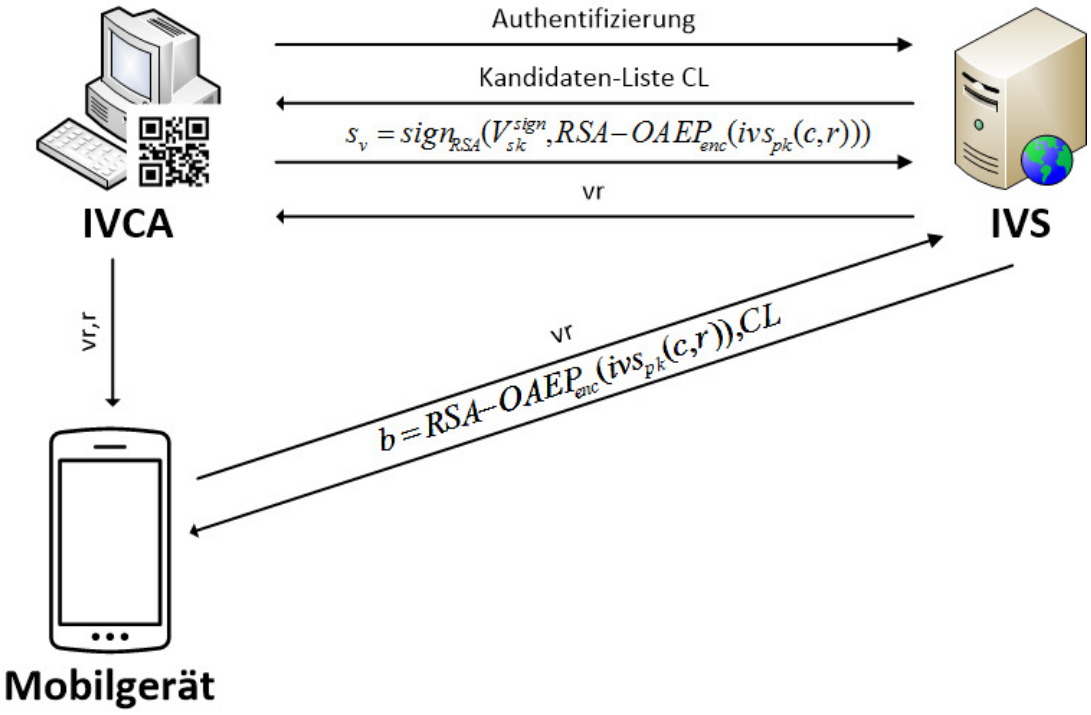


Abbildung 14: Wahlablauf mit erhöhter Verifizierbarkeit in Anlehnung an [156].

Die Übertragung von r und vr vom Wahlcomputer auf das Verifikations-Gerät wird aufgrund der besseren Benutzbarkeit über einen QR-Code realisiert. Das mobile Gerät berechnet nun für jedes $c' \in$ CL, $c_{enc} = RSA - OAEP_{enc}(ivs_{pub}, (c', r))$. Falls ein $c_{enc} = b$ existiert, wird c' zusammen mit dem zugehörigen Kandidatennamen ausgegeben. Der Wähler kann nun überprüfen, ob c' seiner Wahl c, die er zuvor auf dem Wahlcomputer getroffen hat, entspricht. Somit kann der Wähler verifizieren, ob seine Wahl korrekt auf dem VSS gespeichert wurde.

Dieses Vorgehen zur Verifizierung kann jetzt in einem Zeitraum von 60 Minuten[4] maximal drei mal durchgeführt werden. Wird bei dem Verifizierungsprozess vom Wähler ein Fehler festgestellt, empfiehlt die Wahlbehörde NEC von einem anderen PC erneut zu Wählen und die anschließende Verifizierung mit einem anderen Mobilgerät durchzuführen. Tritt der Fehler erneut auf, sollte der Wähler an der Präsenzwahl teilnehmen und die Wahlbehörde NEC informieren [154, 156].

Diskussion der von Heiberg und Willemson getroffenen Sicherheitsannahmen

In diesem Abschnitt werden die Voraussetzungen, unter welchen die Integrität der Stimme sowie das Wahlgeheimnis während der Verifizierung gewahrt bleiben, genannt und diskutiert.

DAS MOBILGERÄT KENNT DIE WÄHLERIDENTITÄT NICHT: Falls das Smartphone oder das Tablet die Wähleridentität kennt, lernt dieses offensichtlich auch die Stimme des Wählers, da das $c' \in$ CL, für welches $b = RSAenc(ivs_{pub}, OAEP(c', r))$ gilt, das c' ist, für welches der Wähler votiert hat [156].

Natürlich sei an dieser Stelle erwähnt, dass diese Sicherheitsmaßnahme überhaupt nicht gelten kann, da auf vielen mobilen Geräten personenbezogene Daten gespeichert sind. Diese personenbezogenen Daten können einfach mit einer Person oder zumindest einer Personengruppe in Verbindung gebracht werden, da es unwahrscheinlich ist, dass eine Person ihre Stimme auf einem fremden Gerät verifiziert. Man bedenke, dass jedes Smartphone eine Telefonnummer besitzt, welche eindeutig einer Person zugeordnet werden kann. Weitere Beispiele sind Daten, die einfach auf die Identität des Benutzers

4 Bei der Wahl 2013 betrug der Zeitraum 30 Minuten.

schließen lassen und auf vielen mobilen Geräten vorhanden sind. Beispiele sind E-Mail-Adressen, Benutzernamen in sozialen Netzwerken und GPS-Daten.

PC UND MOBILGERÄT SIND NICHT KOMPROMITTIERT: Des Weiteren wird angenommen, dass das mobile Gerät und der Wahl-PC nicht gleichzeitig mit Malware infiziert sind, sodass ausgeschlossen werden kann, dass beide Devices einen Angriff koordinieren können. In einem solchen Fall könnte der PC dem Mobilgerät mitteilen, welchen Kandidaten die Verifikations-App ausgeben soll, ohne die entsprechende Berechnung zur Verifizierung durchzuführen. Die Autoren des Papers sagen, dass dies vom Benutzer mit hoher Wahrscheinlichkeit entdeckt wird, da der Angreifer nicht davon ausgehen kann, dass das Mobilgerät im Besitz des angegriffenen Wählers ist und es außerdem Wähler geben wird, welche die Wahl über ein fremdes Mobilgerät verifizieren [156].

Auch hier kann davon ausgegangen werden, dass diese Annahme nur in wenigen Fällen Gültigkeit besitzt. Denn es gibt durchaus Situationen, in denen der PC mit dem mobilen Gerät verbunden ist und in denen eine Malware das Mobilgerät infizieren kann. Oft wird das Mobilgerät z. B. zum Laden oder zum Datentransfer an den PC angeschlossen. Des Weiteren werden häufig gemeinsame Cloud Services, wie z. B. Dropbox, genutzt, um nur einige Infektionswege zu nennen [154, 158]. Ein solcher Angriff wurde von Springall et al. [154] als Proof-of-Concept implementiert und beschrieben.

Die Argumentation, die Wahl über ein fremdes Gerät zu verifizieren, ist ein Risiko, da dieses (und konsequenterweise auch sein Besitzer) die Wahlentscheidung erfährt. Folglich wird so etwas häufig in familiären Strukturen durchgeführt, in welchen die Geräte oft im gleichen Netzwerk sind und dadurch die Wahrscheinlichkeit steigt, dass diese Geräte auch mit Malware infiziert sind.

Bei der Wahl kann eine Malware auf dem Wahlcomputer die vr zusammen mit der abgegebenen Stimme c auf einen Command-and-Control-Server übertragen. Bei der Verifikation ermittelt die Malware auf dem Mobilgerät durch eine Anfrage an den Command-and-Control-Server über die vr den richtigen Kandidaten c und zeigt diesen Kandidaten an. Dies ist nur dann möglich, wenn eine große Anzahl an Geräten infiziert ist. Falls es Wähler gibt, die eine solche Verifikation auf fremden Geräten durchführen, die nicht mit dieser Malware infiziert sind, wird ein solcher Angriff sehr wahrscheinlich erkannt. Es besteht allerdings die Frage, wie in diesem Fall weiter vorgegangen wird.

Angriffe

GHOST-CLICKING-ATTACK Die Ghost-Clicking-Attack ist eine Erweiterung des Student-Angriffs (vgl. Unterabschnitt 5.1.3), welche unter der Voraussetzung funktioniert, dass der Wähler seinen Ausweis nach der Wahl im Lesegerät vergisst. Die Annahme ist, dass der Computer des Wählers mit einer Malware infiziert ist, welche den eigentlichen Wahlvorgang des Wählers beobachtet und die Eingabe der PIN aus dem Speicher liest und speichert. Anschließend wartet die Malware, bis die Zeit zur Verifizierung abgelaufen ist und überprüft dann, ob der Ausweis noch im Lesegerät ist. Ist dies der Fall, so wählt die Malware erneut. Der Wähler bekommt davon nichts mit, da dies verdeckt geschieht. Ist der Ausweis nicht mehr im Lesegerät, wartet die Malware bis der Ausweis wieder präsent ist. Da mit dem Ausweis viele Anwendungen betrieben werden, ist die Wahrscheinlichkeit, dass dieser während der Wahlphase noch einmal benutzt wird, ziemlich hoch. Der Angriff wurde von seinen Autoren [154] als Proof-of-Concept für das Betriebssystem Linux entwickelt. Die dabei entstandene Malware injiziert sich dabei vom Userspace aus in die IVCA, indem es den `ptrace` Befehl nutzt. Da die Wahl verdeckt abläuft, bekommt der Wähler keine Wahlreferenznummer und hat somit keine Möglichkeit, seine Wahl zu verifizieren. Er geht deshalb weiter davon aus, dass seine Wahl in das Ergebnis einfließen wird. Nur falls der Wähler nach der Wahl Zugriff auf die Log-Dateien besitzt und seine PCI im Log2 öfter auftaucht, als er gewählt hat, kann der Wähler eine Manipulation feststellen. Allerdings darf die Log-Datei nicht veröffentlicht werden, wie in Unterabschnitt 5.1.3 bereits erwähnt wurde [154].

5.1.5 Zwischenfälle bei Wahlen

Im Folgenden Kapitel werden einige Zwischenfälle aufgezeigt welche weitere Schwächen des estnischen Wahlverfahrens aufzeigen. Einer dieser Zwischenfälle ereignete sich während der Auszählung der Präsidentschaftswahl 2011. Bei der Auszählung fiel eine ungültige Stimme auf. Da das estnische Wahlsystem eigentlich keine ungültigen Stimmen zulässt[5], wurde seitens des NEC eine Untersuchung angeordnet. Dabei sollte das Wahlgeheimnis nach der Auszählung gebrochen werden. Dies ist möglich, da in Log1 die PCI mit dem SHA1-Hash der anonymen Stimme b gespeichert wird. Allerdings kam es dann nach einer Abwägung doch nicht dazu, wobei nicht geprüft werden kann, ob die NEC die Stimme nicht doch entschlüsselt hat. Dieser Fall zeigt, dass die bestehende Möglichkeit einer Entschlüsselung eventuell irgendwann doch ausgenutzt werden könnte, obwohl es vorab nicht vorgesehen war [155].

Weitere „alltägliche“ Zwischenfälle zeigt die Sicherheitsanalyse von Springall et al. [154]. Die Autoren Kitcat, Hursti, MacAlpine und Halderman haben die Kommunalwahl im Oktober 2013 als Wahlbeobachter begleitet und die Wahl analysiert. Dabei haben sie eine Reihe von Problemen festgestellt [154]. Zum einen wurde festgestellt, dass die Prozesse, welche die Sicherheit gewährleisten sollen, nicht eingehalten werden oder sogar ganz fehlen. So wurde festgestellt, dass Prozesse, die festlegen, wie mit Anomalien im System umgegangen wird, nicht vorhanden oder nicht ausreichend beschrieben sind, weswegen auftretenden Anomalien im System nicht nachgegangen wird. Stattdessen wird versucht, diese möglichst einfach zu lösen. Oft werden diese Entscheidungen zur Problembehandlung von einer einzelnen Person getroffen. Eine weitere Folge ist, dass die Prozesse sich während der Wahl unerklärlich ändern. So durften die Autoren des Papers am Anfang der Wahl ihre Mobiltelefone mit in den Serverraum nehmen und ein paar Tage später war dies nicht mehr gestattet.

Ein Problem mit den Vorschriften und vorgegebenen Prozessen war, dass selbst wenn diese klar definiert sind, sie vom Personal nicht eingehalten wurden. So beobachteten die Autoren, dass Updates und Backups von einer einzelnen Person ausgeführt wurden, obwohl nach den Statuten eigentlich immer zwei Personen anwesend sein müssten. Somit war nach diesem Zeitpunkt die Integrität des Systems von nur einer Person abhängig, welche alle Sicherheitsanforderungen hätte kompromittieren können.

Des Weiteren wird in dem Paper der schlechte Umgang mit der Sicherheit des Systems beschrieben. So wird sorglos mit Passwörtern und physikalischen Schlüsseln zum Serverraum umgegangen. Außerdem wurde die IVCA mit einem System erstellt, welches nicht neu und genau für diesen Zweck installiert war. Das bedeutet, wenn dieses System mit Malware infiziert gewesen ist, könnte das Verhalten der IVCA beim Kompilieren modifiziert worden sein. Ein weiteres Sicherheitsproblem ist, dass jeder, der an den Servern etwas ändert, dies prinzipiell mit Root-Rechten tut, was es unmöglich macht, gewissen Personen oder Gruppen bestimmte Rechte oder Rollen zuzuweisen und die Aktionen einer Person dadurch zu protokollieren. Weiter wurde festgestellt, dass Backups und die Wählerliste nicht verschlüsselt und ohne die Sicherung der Integrität transportiert wurden. Der schwerwiegendste Fehler wurde allerdings während der Auszählungsphase gemacht, als das ausgezählte Ergebnis vom VCS mit einem privaten USB-Stick auf einen Windows-Rechner übertragen wurde, auf welchem dann das Ergebnis signiert wurde. Eigentlich ist es vorgesehen, das Ergebnis per DVD vom VCS auf einen anderen Client zu übertragen. Grund für die Abweichung war, dass es nicht genauer spezifizierte Probleme mit dem DVD-Brenner gab.

Bemängelt wurde auch die fehlende Transparenz der Vorgänge innerhalb des IVS. Die Transparenz soll durch Wahlbeobachtung, Veröffentlichung von Videos der wichtigsten Vorgänge und durch die Veröffentlichung des Quellcodes gewährleistet werden. Hierbei stellten die Autoren während ihrer Beobachtung fest, dass kritische Fehler aufgetreten waren, welche auf dem späteren Video nicht zu sehen waren, da die Kamera gerade etwas anderes filmte. Außerdem mussten die Autoren feststellen, dass ein Wahloffizieller bei einem kritischen Fehler schnell die Konsolenausgabe löschte und die Autoren dann aufforderte, den Raum zu verlassen, um andere Wahlbeobachter herein zu lassen, sodass der Wahloffizielle einen Moment unbeaufsichtigt war, um das Problem zu beheben.

5 Da die IVCA nicht die Möglichkeit bietet, ungültig zu wählen.

Ein weiteres Problem ist, dass nicht der gesamte Quellcode der verwendeten Software öffentlich zugänglich ist, was die Überprüfung freilich erschwert. Ein wichtiger Teil, welcher nicht öffentlich und sogar noch durch Obfuskationsmechanismen verschleiert ist, ist der Quellcode der IVCA. Die Begründung der NEC ist, dass es erschwert werden soll, Malware zu schreiben, welche die IVCA angreift oder imitiert. Wie im Rahmen der Analyse dargestellt, ist dies mit erhöhtem Aufwand durch Reverse Engineering trotzdem möglich [154]. Der Nachteil ist allerdings, dass Sicherheitslücken so nur schwer entdeckt werden können.

Der wichtigste Kritikpunkt an den Maßnahmen zur Schaffung von Transparenz ist, dass die Maßnahmen nicht zeigen und auch nicht zeigen können, was wirklich auf der Festplatte des Systems geschieht. Sie können höchstens Indizien liefern. Zusammen mit den oben genannten Beobachtungen ist dies sogar gefährlich, da die Maßnahmen den Anschein von Sicherheit bieten.

Zum Schluss bleibt noch zu erwähnen, dass der veröffentlichte Quellcode aus 17.000 Zeilen besteht, wobei die Codebasis sehr komplex ist und viele externe Abhängigkeiten besitzt, was zum einen eine Suche nach Sicherheitslücken sehr schwierig gestaltet und zum Anderen die Sicherstellung der korrekten Funktionsweise des Wahlablaufs nahezu unmöglich macht [154].

Die beschriebenen Beobachtungen und Zwischenfälle erschüttern das Vertrauen in die Wahloffiziellen sowie in die organisatorischen Maßnahmen zur Sicherung der Wahlgrundsätze. Die Beobachtungen zeigen vor allem, dass das System in dieser Form zu komplex ist, um die Einhaltung der Wahlgrundsätze mit organisatorischen Mitteln zu gewährleisten, da nicht alle Eventualitäten durch vorgegebene Prozesse beschrieben werden können, es immer menschliche Fehler geben wird und die Verifizierung der eingesetzten Software zu komplex ist. Ein Lösungsansatz wäre es, die organisatorischen Maßnahmen durch kryptografisch verifizierbare Methoden zu ersetzen, siehe dazu Kapitel 4 sowie Abschnitt 5.3.

5.1.6 Bewertung

Diese Bewertung umfasst beide Varianten des estnischen Wahlsystems. Die ursprüngliche Version, welche in Unterabschnitt 5.1.3 beschrieben wird, wird im Folgenden mit Version 1 und das in Unterabschnitt 5.1.4 beschriebene Protokoll mit Version 2 bezeichnet.

INDIVIDUELLE VERIFIZIERBARKEIT In der Version 1 gibt es keine individuelle Verifizierbarkeit. Erst Version 2 unterstützt dies. Allerdings besteht nur die Möglichkeit zu verifizieren, ob die Stimme auf dem Vote Storage Server eingegangen ist. Da der Wähler die entsprechenden Parameter besitzt, kann er überprüfen, ob die Stimme den korrekten Kandidaten enthält. Folglich erfüllt das Wahlprotokoll die Eigenschaft der inneren individuellen Verifizierbarkeit vor der Auszählung (IV.1.2). Sobald der Angreifer allerdings die Kontrolle über das Mobilgerät sowie über den Wahlcomputer besitzt, kann diese Eigenschaft nicht eingehalten werden.

UNIVERSELLE VERIFIZIERBARKEIT Bei beiden Versionen kann der Wähler nicht verifizieren, ob das Ergebnis korrekt ist. Selbst die Auditoren, welche Zugriff auf die Log-Dateien haben, können das nicht, da nicht gewährleistet werden kann, dass der Eintrag der Log-Dateien dem entspricht, was der Computer tatsächlich berechnet hat. Das bedeutet allerdings nicht, dass das Ergebnis immer falsch ist, sondern nur, dass die Korrektheit des Ergebnisses nie nachvollzogen werden kann. Deshalb kann ein Angreifer, welcher die Kontrolle über das Wahlsystem erlangt, das Ergebnis unbemerkt manipulieren. Das bedeutet beide Versionen unterstützen keine Form der universellen Verifizierbarkeit.

WAHLGEHEIMNIS Das Wahlgehemnis wird bei beiden Versionen unter der Voraussetzung eingehalten, dass kein unerlaubter Zugriff auf das Wahlsystem oder den Wahlcomputer erfolgt, da die Stimme verschlüsselt übertragen wird. Aufgrund der nicht gegebenen individuellen Verifizierbarkeit der Version 1 besitzt diese sogar die Quittungsfreitheit und die Unmöglichkeit von Stimmenkauf (QF.2). Die Voraussetzung hierfür ist, dass die Log-Dateien gegenüber der Öffentlichkeit geheim bleiben, was offiziell so vorgesehen ist. Version 2 des estnischen Wahlsystems verliert die Quittungsfreitheit und die

Unmöglichkeit von Stimmenkauf durch Einführung der individuellen Verifizierbarkeit. Der Grund dafür ist, dass der Wähler seine Parameter (zumindest 30 Minuten) als Quittung nutzen kann. D. h. der Wähler gibt dem Angreifer die Wahlreferenznummer sowie die verwendete Zufallszahl. Bekommt der Angreifer Zugriff auf das Wahlsystem, so kann er das Wahlgeheimniss brechen, da er Zugriff auf den entsprechenden Schlüssel besitzt, beispielsweise kann der Angreifer die verschlüsselten Stimmen vom VSS zum VCS weiterleiten, ohne die Signatur zu entfernen. Hat der Wähler Zugriff auf den Wahlcomputer, kann er die Eingabe des Wählers einfach beobachten. Der Grund dafür ist, dass das estnische Wahlsystem das Secure Platform Problem nicht beachtet.

NICHT ERPRESSBARKEIT Das estnische Wahlsystem in der Version 1 ist resistent gegen Randomisierungsangriffe (RA.1), falls der Angreifer keinen Zugriff auf das Wahlsystem sowie den Wahlcomputer besitzt, da das System keine Informationen nach außen gibt. Auch hier wird angenommen, dass die Log-Dateien gegenüber der Öffentlichkeit geheim bleiben. Für Angreifer, die Zugriff auf das Wahlsystem bzw. den Wahlcomputer haben, sind Randomisierungsangriffe unnötig, da somit bereits das Wahlgeheimnis gebrochen ist. Die Version 2 hingegen kann Randomisierungsangriffen nicht standhalten, da das System nicht quittungsfrei ist. Das bedeutet, dass der Angreifer durch die erzwungene Herausgabe der Wahlreferenznummer sowie der Zufallszahl die Wahl lernt.

Gegen Abwesenheitsangriffe (AA.2) ist das Wahlsystem (Version 1 und 2) für Angreifer, welche keinen Zugriff auf das Wahlsystem haben, zumindest resistent. Der Angreifer kann zwar beobachten, ob der Wahlcomputer eine Verbindung zum Wahlsystem aufbaut, allerdings kann der Wähler dies umgehen, indem er zur Präsenzwahl antritt. Hat der Angreifer allerdings Zugriff auf die Log-Dateien bzw. den Vote Storage Server, kann er entscheiden, ob der Wähler gewählt hat, indem er überprüft, ob eine Stimme mit seiner Signatur existiert.

Im estnischen Wahlsystem sind Simulationsangriffe ausschließlich in Kombination mit Abwesenheitsangriffen möglich. Der Grund hierfür ist, dass der Angreifer verhindern muss, dass der Wähler die Wahl durch die Präsenzwahl widerruft. Folglich ist das Wahlsystem resistent gegen Simulationsangriffe für Angreifer, welche keinen Zugriff auf die Log-Dateien bzw. den Vote Storage Server haben. Hat der Angreifer Zugriff auf den Vote Storage Server oder die Log-Dateien, kann er überprüfen, ob der Wähler seine Wahl bei der Präsenzwahl (oder auch durch Re-Voting) widerrufen hat und so feststellen, dass der Wähler seinen Anweisungen nicht gefolgt ist. Eine Erschwerung von Simulationsangriffen ist, dass die Preisgabe der Zugangsdaten die Herausgabe des Personalausweises miteinschließen muss. Da sich mit dem Personalausweis elektronisches Banking sowie die Unterzeichnung rechtskräftiger Verträge tätigen lassen, stellt die Herausgabe des Personalausweises eine erhebliche Hürde für eigentlich kooperationswillige Wähler dar.

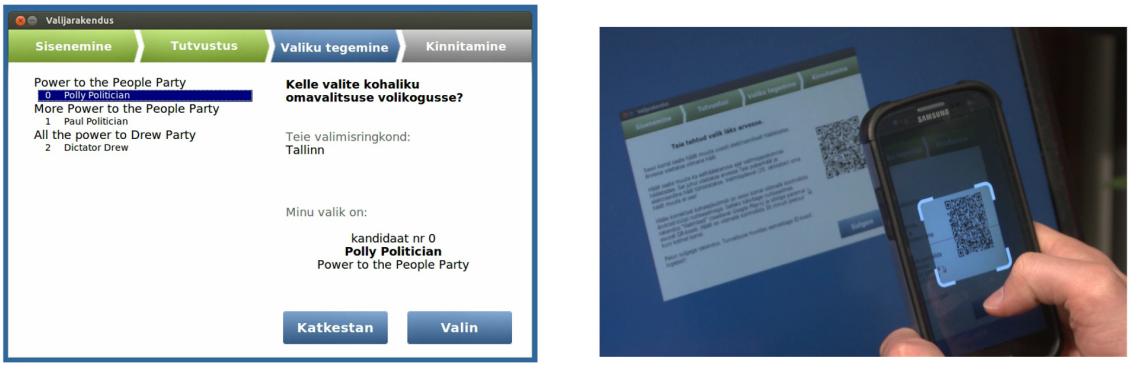
ROBUSTHEIT Die innere Robustheit des Systems ist bei beiden Versionen gegeben, da für das Hardware Security Modul ein vier aus sieben Multiparty-Authentication-Protokoll zum Einsatz kommt, was bedeutet, dass es ausreicht, wenn vier der sieben Wahloffiziellen der Entschlüsselung der Stimmen zustimmen. Somit muss sich mehr als die Hälfte der Wahloffiziellen weigern, die Stimmen zu entschlüsseln. Dadurch erfüllt das estnische Wahlsystem die Eigenschaft der inneren Robustheit (RI.1). Die äußere Robustheit der Wahl wird ebenfalls von beiden Versionen erfüllt, da das I-Voting optional ist und somit im Fall eines Angriffs alle Wähler die Möglichkeit besitzen, an der Präsenzwahl teilzunehmen.

BENUTZBARKEIT Die Benutzbarkeit bei der Wahl ist bei beiden Versionen gleich. Die Wahlentscheidung kann der Wähler einfach durch einen Klick auf den entsprechenden Kandidaten ausdrücken. Am eigentlichen Wahlvorgang ist nur eine einmalige Teilnahme nötig. Außerdem werden außer dem Personalausweis keine weiteren Hilfsmittel benötigt. Da der Personalausweis nicht extra für die Wahl angefertigt wird, sondern ein wichtiger Bestandteil des estnischen e-Government-Konzepts darstellt, wird dieser hier nicht explizit als Hilfsmittel angesehen. Allerdings stellt dies hinsichtlich der Benutzbarkeit eine Einschränkung gegenüber anderen Wahlverfahren dar, was im Bewertungssystem aller-

dings nicht berücksichtigt wird. Folglich erfüllt die I-Vote Client Application die Eigenschaft Klick-Voting mit einmaliger Teilnahme ohne Hilfsmittel (BW1.1).

Die Benutzbarkeit der Verifizierung wird ausschließlich für die Version 2 beschrieben, da Version 1 keine individuelle Verifizierbarkeit unterstützt. Wie in Unterabschnitt 5.1.4 beschrieben, muss der Wähler einen QR-Code scannen (siehe Abbildung 15). Anschließend wird seine Stimme vom VSS heruntergeladen und die entsprechende Kandidatennummer ausgegeben. Der Wähler muss dann die Kandidatennummer mit der Nummer vergleichen, welche neben seinem Kandidaten ausgegeben wird. Folglich stellt dies einen Vergleich zweier sehr kurzer Nummern dar und erfüllt somit die Eigenschaft (BV.1).

Alles in allem ist das estnische Wahlsystem mit Hinblick auf das Systemdesign eine der am einfachsten zu benutzenden Internetwahlsysteme[6], was allerdings zulasten des Wahlgeheimnisses und der Verifizierbarkeit geht.



(a) I-Vote Client Application. **(b)** Individuelle Verifizierung.

Abbildung 15: Benutzbarkeit des estnischen Wahlsystems [154].

5.2 POLYAS

Zwar unterliegen die Wahlen, für die Polyas bisher erfolgreich eingesetzt wurde, nicht direkt den durch das Grundgesetz festgelegten Anforderungen, da es sich um Wahlen dritter Ordnung handelt (siehe Abschnitt 2.1). Aufgrund der anzunehmenden fachlichen Expertise der durchführenden Institutionen [34], zu denen unter anderem die Gesellschaft für Informatik (GI), die Association for Computing Machinery (ACM) und die Deutsche Forschungsgemeinschaft (DFG)) gehören, sowie der Überprüfung durch das Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI) nach Common Criteria Schutzprofil BSI-CC-PP-0037-2008 [35] und darauf aufbauender Zertifizierung durch das BSI [159], erscheint es den Autoren jedoch notwendig, die eingesetzte Lösung im Rahmen dieser Arbeit trotzdem genauer zu betrachten.

5.2.1 Komponenten

Das Polyas Internetwahlsystem besteht aus mehreren Komponenten, welche jeweils bestimmte voneinander getrennte Funktionen in verschiedenen Phasen der Wahl übernehmen und zur Gewährleistung sicherer Kommunikation vor der Wahl jeweils ihre öffentlichen Schlüssel untereinander austauschen. Diese Komponenten werden auf Basis der öffentlich verfügbaren Literatur [34, 36, 37, 160] im Folgenden kurz erläutert.

6 Hier ist nicht die praktische Umsetzung der Benutzbarkeit bewertet, sondern nur, was nach formaler Spezifikation möglich wäre.

Erzeugungsserver (CS)

Der Erzeugungsserver (Creation Server) *CS* erzeugt getrennte Listen von Transaktionsnummern (TANs) und deren jeweiligen Hashwerten. Die TANs werden zusammen mit Namen und Adressen der Wahlberechtigten verschlüsselt an den Druckdienstleister übermittelt. Die Hashes der TANs werden zusammen mit den IDs der Wahlberechtigten an das elektronische Wählerverzeichnis weitergeleitet. Zusätzlich werden lediglich die Hashes an den Validierungsserver weitergeleitet.

Druckdienstleister (PS)

Der Druckdienstleister (Printing Service) *PS* druckt die erhaltenen TANs und versendet sie an die Wahlberechtigten.

Validierungsserver (VS)

Der Validierungsserver (Validation Server) *VS* validiert vom elektronischen Wählerverzeichnis empfangene TANs und generiert bei erfolgreicher Validierung ein Wahltoken T, das er an das elektronische Wählerverzeichnis und den Urnenserver weiterleitet.

Elektronisches Wählerverzeichnis (ERS)

Das elektronische Wählerverzeichnis (Electoral Registry Server) *ERS* besitzt TAN-Hashes sowie IDs der Wahlberechtigten. Laut Neumann et al. [36] sowie Menke und Reinhard [37] stellt das elektronische Wählerverzeichnis außerdem das Web-Frontend bereit, das Wahlberechtigte zur Abgabe ihrer Wahl nutzen können. Olemba et al. [160] beschreiben für diese Funktionalität jedoch eine eigene Komponente (Vote Casting Interface / Stimmenabgabe-Interface), die auf dem selben Server wie das elektronische Wählerverzeichnis laufen kann, aber nicht muss. Das Vote Casting Interface wird im folgenden Abschnitt deshalb separat behandelt.

5.2.2 Stimmenabgabe-Interface (VCI)

Das auf HTML basierende Stimmenabgabe-Interface (Vote Casting Interface) *VCI* läuft auf einem Webserver, der den Point of Entry für die Wahlberechtigten darstellt und ihnen gegenüber Authentifizierungs- sowie die eigentliche Wahlfunktionalität anbietet. Das Interface ist aus Gründen der Barrierefreiheit so gestaltet, dass es von möglichst vielen Browsern (unter anderem auch dem textbasieren Browser „Lynx“[7]) unterstützt wird.

Urnenserver (BBS)

Der Urnenserver (Ballot Box Server) *BBS* prüft das Wahltoken T und stellt Wahlberechtigten bei erfolgreicher Prüfung einen elektronischen Stimmzettel bereit. Der Urnenserver speichert den elektronischen Stimmzettel dann mit dem öffentlichem Schlüssel der Auszählungskomponente verschlüsselt, an die die Stimmen nach dem Ende der Wahl übertragen werden.

Auszählungskomponente (TC)

Die Auszählungskomponente (Tallying Component) *TC* erzeugt in der Vorbereitungsphase ein asymmetrisches Schlüsselpaar, wobei der öffentliche Schlüssel zum Verschlüsseln der Wählerstimmen während der Wahlphase verwendet wird. Der private Schlüssel wird durch den Einsatz zweier separater Passwörter, die von zwei unabhängigen Wahloffiziellen verwaltet werden, verschlüsselt und gespeichert. Mit Hilfe dieses privaten Schlüssels werden nach der Wahl die vom Urnenserver verschlüsselten elektronischen Wahlzettel wieder entschlüsselt und ausgezählt. Laut Olembo et al. [160] wird die Auszählungskomponente offline betrieben.

7 http://lynx.browser.org/

Verifikationskomponente (VC)

Die Verifikationskomponente (Verification Component) *VC* überprüft unter Zuhilfenahme der auf dem Urnenserver gespeicherten, verschlüsselten Stimmen, der signierten Hashwerte des elektronischen Wählerverzeichnisses und des privaten Schlüssels der Auszählkomponenten, ob die Anzahl der Stimmen mit der Anzahl der wahlberechtigten Personen, die tatsächlich gewählt haben, übereinstimmt und ob die Werte der vom Urnenserver erstellten Hash-Chain-Blöcke valide sind. Außerdem werden die Stimmzettel entschlüsselt und die Stimmen / Wahlentscheidungen ausgezählt [160], sodass die Ergebnisse mit denen der Auszählungskomponente verglichen werden können.

Wahlausschuskomponente (CT)

Wie die Bezeichnung erahnen lässt, wird die Wahlausschusskomponente (Committee Tool) *CT* dem Wahlausschuss zur Verfügung gestellt und dient der Steuerung der Wahl. Mit ihr kann die Wahl gestartet und gestoppt werden.

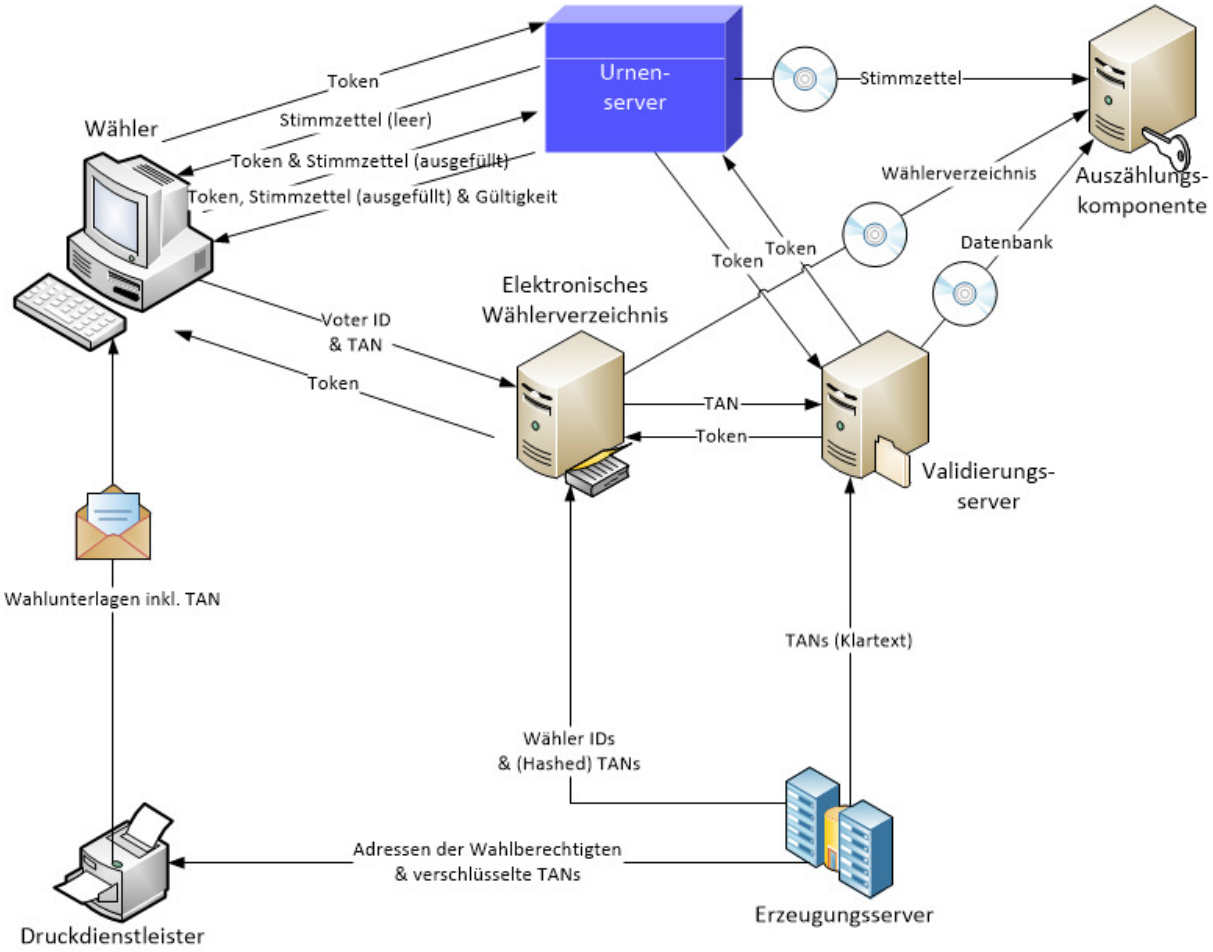


Abbildung 16: Vereinfachte Darstellung der Architektur von Polyas in Anlehnung an [34, 36, 37, 160].

5.2.3 Funktionsweise

Der Ablauf des Wahlvorgangs gliedert sich in drei Phasen. Dabei wird davon ausgegangen, dass der wahlberechtigten Person ihre ID bereits *a priori* bekannt ist. Das kann z. B. eine Personalausweis- oder aber, wie bei den Wahlen der Gesellschaft für Informatik, eine Mitgliedsnummer sein.

Pre-Wahlphase

In der Pre-Wahlphase werden vom Erzeugungsserver *CS* Wahl-TANs generiert und gespeichert. Von diesen wird unter Zuhilfenahme einer Einweg-Hashfunktion jeweils eine Prüfsumme berechnet und diese Prüfsumme der ID einer wahlberechtigten Person zugewiesen, woraufhin die Liste in dieser Form gespeichert wird. Außerdem werden der TAN-Liste die Adressen der Wahlberechtigten hinzugefügt, verschlüsselt und ebenfalls gespeichert, sodass letztendlich drei TAN-Listen existieren, eine mit TANs im Klartext, mit den Prüfsummen der TANs sowie eine Liste, die die verschlüsselten TANs enthält. Die Liste mit den Klartext TANs wird ohne weitere Informationen in die Datenbank von *VS* geladen. Die Liste mit den Prüfsummen der TANs sowie den IDs der Wahlberechtigten wird an *ERS* gesendet. Die verschlüsselte Liste wird von *CS* an *PS* gesendet, der sie entschlüsselt, druckt und die TANs (geschützt hinter einem Rubbelfeld) zusammen mit anderen Wahlunterlagen postalisch an die jeweils zugewiesene Adresse der Wahlberechtigten versendet. Abbildung 17 skizziert die Verteilung der TANs sowie der jeweils zugehörigen Informationen.

Nun werden die von den Servern benötigten asymmetrischen Schlüsselpaare für Ver- und Entschlüsselung sowie zur Signierung erzeugt und die öffentlichen Schlüssel bei Bedarf (d. h. wenn Kommunikation zwischen verschiedenen Komponenten stattfindet) jeweils untereinander ausgetauscht. Die Fingerprints der öffentlichen Schlüssel von *BBS* und *ERS* werden auf der Website der Wahl sowie in den Wahlunterlagen veröffentlicht.

Außerdem wird das Wählerverzeichnis in *ERS* geladen, sichergestellt, dass die Datenbank des Urnenservers *BBS* leer ist sowie die digitalen Wahlzettel-Vorlagen und andere Wahl-Informationen in eben diesen Server geladen. Zu Absicherung von *BBS*, *ERS* und *VS* werden für jeden der Server jeweils zwei Token generiert, die nötig sind, um remote auf sie zuzugreifen. Diese Tokens werden unter sechs unabhängigen Mitgliedern des Wahlkomitees verteilt. Abschließend werden die drei Server konfiguriert und zusätzliche Programme, wie z. B. Anti-Virus Software, installiert.

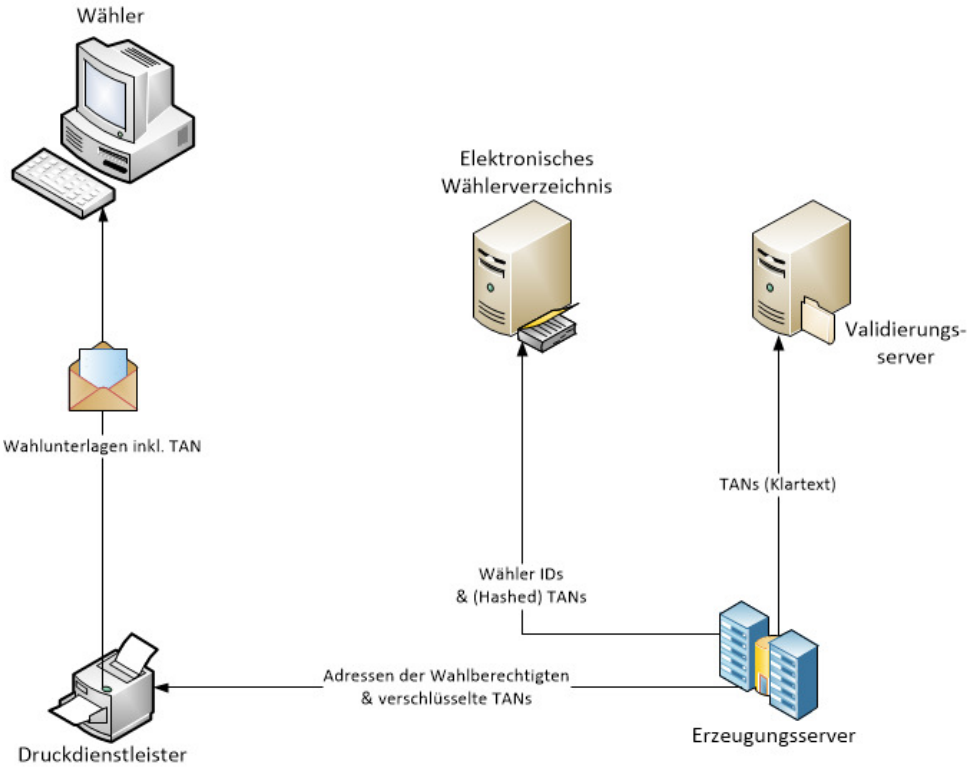


Abbildung 17: Vereinfachte Darstellung der Kommunikation während der Pre-Wahlphase von Polyas in Anlehnung an [34, 36, 37, 160].

Wahlphase

Nach Abschluss der Pre-Wahlphase, in der die beteiligten Komponenten vorbereitet und die Credentials an die Wahlberechtigten verteilt wurden, startet die eigentliche Wahlphase. Hierfür besucht die wahlberechtigte Person die auf *ERS* gehostete Wahl-Website und vergleicht idealerweise den mit den Wahlunterlagen versandten Fingerprint des SSL/TLS-Zertifikats von *ERS* mit dem tatsächlich im Browser angezeigten Fingerprint.

Abbildung 18 zeigt die detaillierte Abfolge der einzelnen, im Folgenden beschriebenen Nachrichten des Polyas-Wahlprotokolls zwischen den unterschiedlichen Komponenten. Die Vertraulichkeit und Integrität der Nachrichten ist jederzeit durch SSL/TLS geschützt.

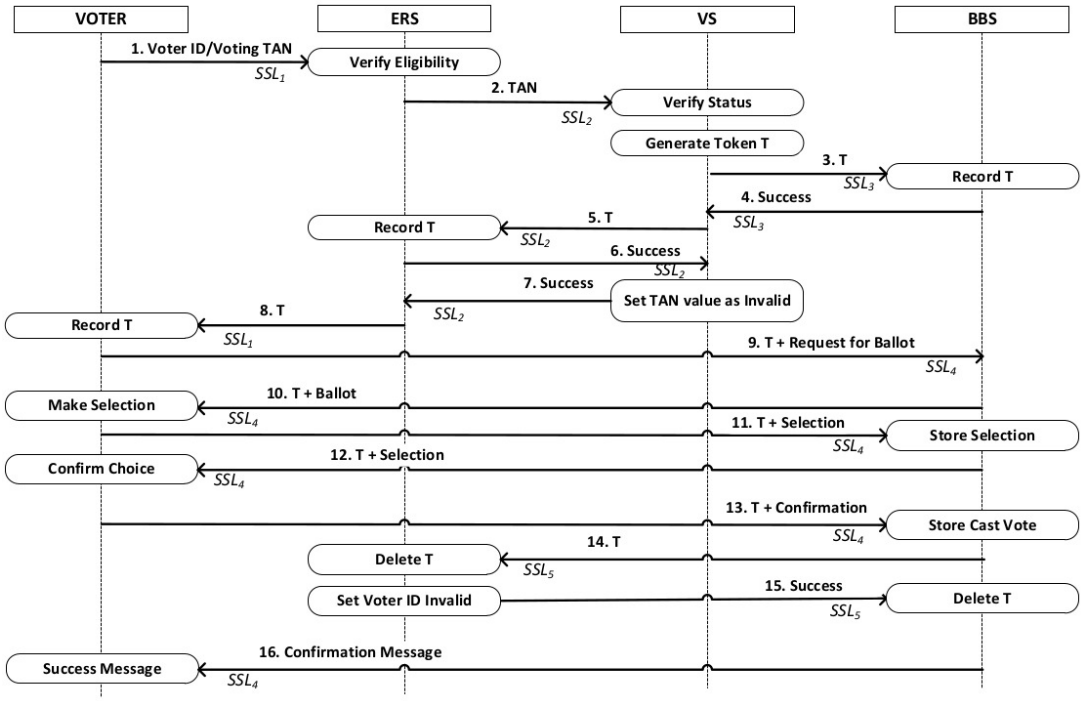


Abbildung 18: Ablaufdiagramm des Polyas-Wahlprotokolls [160].

Nachdem sich der Wähler mit Hilfe seiner ID und TAN gegenüber dem System authentifiziert hat, überprüft *ERS*, ob die entsprechende ID / TAN Kombination im Wählerverzeichnis existiert. Bei positiver Überprüfung leitet *ERS* die TAN an *VS* weiter, wo wiederum überprüft wird, ob die TAN in der Datenbank enthalten ist. Bei positiver Überprüfung erzeugt *VS* ein Wahl-Token T und schickt es an *BBS*. *BBS* sendet eine Bestätigung über den Erhalt zurück an *VS*, woraufhin *VS* T nun auch an *ERS* sendet und bei erfolgreicher Übermittlung wiederum eine Bestätigung von *ERS* erhält. Sollte die TAN bereits verwendet und von *VS* ein zugehöriges T erzeugt worden sein, sendet *VS* das bereits bestehende T ausschließlich an *ERS*.

ERS leitet T an den Computer des Wählers weiter, welcher T automatisch benutzt, um einen Wahlzettel von *BBS* anzufordern. *BBS* überprüft hierfür, ob es sich bei dem vom Wähler übertragenen T um einen gültigen Token handelt, indem es T mit der Liste der von *VS* enthaltenen und noch nicht wieder gelöschten Tokens vergleicht. Befindet sich T in eben dieser Liste, erachtet *BBS* das Token als gültig und sendet den digitalen Stimmzettel an den Wähler. An dieser Stelle sollte der Wähler idealerweise noch einmal den im Browser angezeigten Fingerprint überprüfen, um die Authentizität des SSL/TLS-Zertifikats von *BBS* zu gewährleisten. Ist der Fingerprint gültig, kann der Wähler sich zwischen den verschiedenen angezeigten Optionen entscheiden und seine Wahl zusammen mit T wieder zurück an *BBS* senden. *BBS* speichert die Wahl zusammen mit T verschlüsselt (unter Benutzung des öffentlichen Schlüssels der Auszählungskomponente *TC*) sowie markiert als „selection“ in seiner Datenbank und

sendet die Wahl / *T* Kombination im Klartext zusammen mit einer Information, ob die Wahl gültig oder ungültig ist über eine sichere HTTPS-Verbindung zurück an den Wähler. Dieser kann seine Wahl durch Wiederholung der vorherigen Schritte nun korrigieren oder aber bestätigen, indem er eine *T* enthaltenden Bestätigungs-Nachricht an *BBS* sendet. Wenn *BBS* das übertragene *T* weiterhin als gültig erachtet, ändert er die Markierung des entsprechenden Datenbank-Eintrags von „selection “ zu „cast vote“. Daraufhin sendet *BBS T* an *ERS*. *ERS* löscht *T*, markiert die zugehörige ID als inaktiv und sendet eine Bestätigung an *BBS* zurück. Sobald *BBS* die Bestätigungsnachricht erhält, werden alle noch verfügbaren Kopien von *T* restlos gelöscht, wodurch jede Verbindung zwischen Wähler und der abgegebenen Stimme gekappt wird. Sobald der Wahl-Prozess abgeschlossen ist, wird dem Wähler eine Bestätigung über das erfolgreiche Einreichen seiner Wahl gesendet.

Um die Integrität der abgegebenen Stimmen auch bei auftretenden Fehlern so weit es geht sicherzustellen, werden immer n verschlüsselte Stimmen (wobei n im Auslieferungszustand 30 entspricht) in zufälliger Reihenfolge als Block zusammengefasst und dienen mit der Prüfsumme des vorherigen Blocks zusammen als Eingabe in eine Einweg-Hashfunktion, deren Ausgabe eine weitere Prüfsumme ist, die wiederum mit dem nächsten Block konkateniert wird und als Eingabe für die Hashfunktion dient. Die Prüfsumme des ersten Blocks wird mit einem zufälligen Initialwert der selben Länge (z. B. 256 Bit im Falle des eingesetzten SHA-256) berechnet. Die Prüfsummen werden von *BBS* jeweils digital signiert und als Nachricht an *ERS* gesendet. Wenn *BBS'* Signatur von *ERS* erfolgreich überprüft wurde, wird die Nachricht von *ERS* gespeichert und eine Bestätigung an *BBS* gesendet.

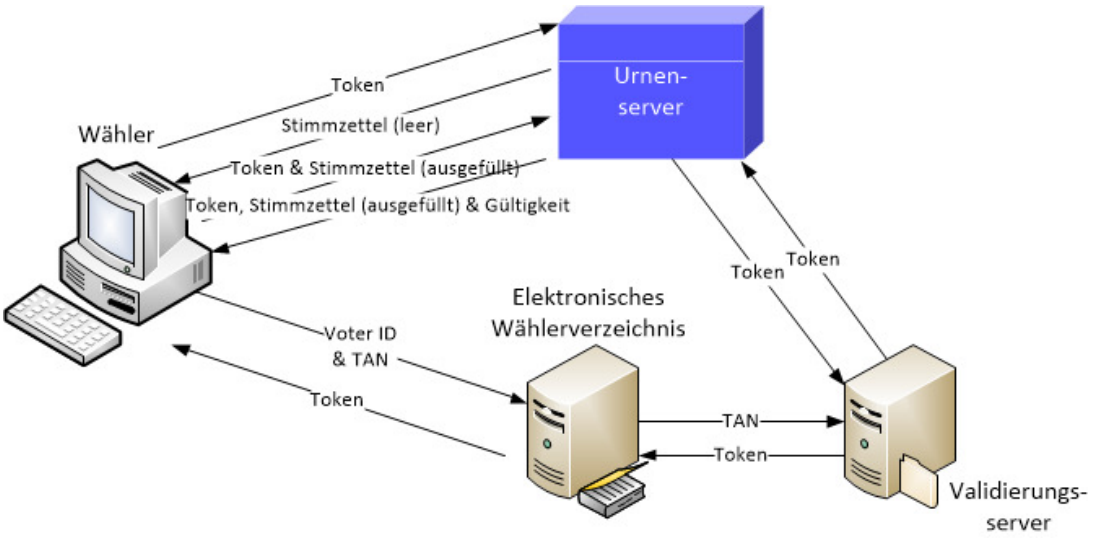


Abbildung 19: Vereinfachte Darstellung der Kommunikation während der Wahlphase von Polyas in Anlehnung an [34, 36, 37, 160].

Neumann et al. [36] schlagen zusätzlich ein Verfahren vor, welches dem Wähler ermöglicht, zu überprüfen, ob seine Stimme ordnungsgemäß vom Urnenserver empfangen wurde. Bei der beschriebenen Eigenschaft handelt es sich um die sogenannte *Cast-As-Intended*-Verifizierbarkeit, welche in etwa mit der in Unterabschnitt 3.2.1 definierten Eigenschaft der inneren individuellen Verifizierbarkeit vor der Auszählung (IV.1.2) zu vergleichen ist, wobei diese jedoch auch die sogenannte *Stored-As-Cast*-Verifizierbarkeit einschließt. Zur Umsetzung dieser Eigenschaft bedienen sich Neumann et al. eines Verfahrens des Code Votings. Konkret werden dem Wähler hierfür vom Druckdienstleister zusätzlich zu den bereits beschriebenen Wahlunterlagen jeweils ein eindeutiger Bestätigungscode sowie ein Offline-Authentifizierungscode (letzterer verborgen unter einem Rubbelfeld) zur Verfügung gestellt. Wenn der Wähler nun die Wahl beginnt, erhält er von *ERS* nicht nur das Token, sondern zusätzlich auch eine mit dem öffentlichen Schlüssel des Urnenservers verschlüsselte Liste von zufälligen Bestätigungscodes, welche er unbearbeitet an den Urnenserver *BBS* weiterleitet. Dieser entschlüsselt die Liste, fügt an der Stelle, der vom Wähler getätigten Wahloption den Bestätigungscode des Wäh-

lers ein und übermittelt die Liste wieder zurück zum Wähler. Dieser überprüft nun wiederum, ob der Bestätigungscode an der Stelle seiner getätigten Wahloption mit dem ihm per Post zur Verfügung gestellten Bestätigungscode übereinstimmt. Ist dies der Fall, kann er sicher sein, dass seine Wahl korrekt an *BBS* übermittelt wurde. Stimmt der Bestätigungscode nicht überein, so öffnet der Wähler den Offline-Authentifizierungscode, wendet sich (unter Benutzung eines anderen Kanals - typischerweise telefonisch) an die Wahlbehörde, authentifiziert sich ihr gegenüber mit Hilfe des Offline-Authentifizierungscodes und meldet die Diskrepanz. Token und Stimmzettel werden daraufhin gelöscht und der Wähler für eine neue Stimmabgabe freigeschalten, welche er idealerweise von einem anderen Endgerät durchführen sollte.

Post-Wahlphase

Der Ablauf des offiziellen Wahlzeitraums markiert sowohl das Ende der Wahlphase als auch den Beginn der Post-Wahlphase, welche in Abbildung 20 schematisch dargestellt ist und auf die im Folgenden näher eingegangen wird. Um die Wahlphase zu beenden und die Post-Wahlphase einzuleiten, werden die sechs in der Pre-Votingphase erstellten Token, die auf die gleiche Anzahl unabhängige Mitglieder des Wahlkomitees verteilt worden sind, benötigt. Mit ihrer Hilfe werden die Datenbanken des *BBS*, des *ERS* und des *VS* heruntergeladen. Die Datenbank des *BBS* wird (üblicherweise offline, also z. B. über DVDs) in die *TC* geladen, dort unter Eingabe der beiden Passphrasen entschlüsselt und ausgezählt. Außerdem wird die Anzahl an abgegebenen Stimmen laut *ERS* und *VS* ermittlet und mit der Anzahl des *BBS* bzw. der *TC* verglichen. Zusätzlich werden die abgegebenen Stimmen ausgedruckt, um manuell verifiziert werden zu können und die Datenbanken der Server werden archiviert.

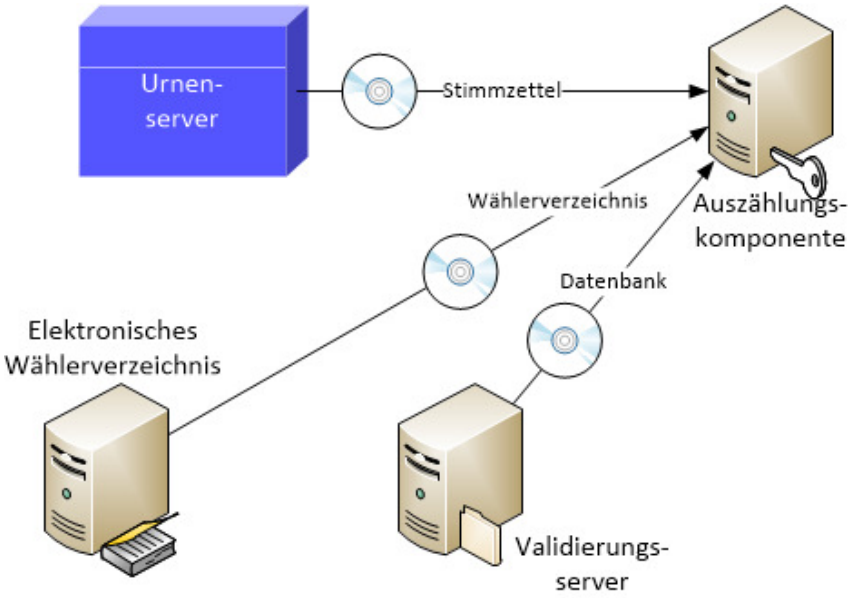


Abbildung 20: Vereinfachte Darstellung der Kommunikation während der Post-Wahlphase von Polyas in Anlehnung an [34, 36, 37, 160].

5.2.4 Bewertung

Olembo et al. [160] erwähnen, dass die Konfiguration der Server und Installation etwaiger „Sicherheitssoftware“ erst erfolgt, nachdem sensible Daten auf die Server geladen wurden. Grundsätzlich sollte die Sicherstellung der Integrität der Systeme sowie die Installation und Konfiguration sicherheitsrelevanter Programme und Parameter jedoch immer schon erfolgen, bevor sensible Daten auf die Systeme geladen oder diese mit jedweder Art von Netzwerken verbunden werden.

INDIVIDUELLE VERIFIZIERBARKEIT Durch die von Neumann et al. [36] vorgeschlagenen Änderungen verfügt Polyas über innere individuelle Verifizierbarkeit vor der Auszählung (IV.1.2).

UNIVERSELLE VERIFIZIERBARKEIT Olembo et al. [160] beschreiben die Verifikationskomponente *VC*, welche es jeder Person ermöglicht, das Wahlergebnis im Sinne der Eigenschaften der bedingten Wahlberechtigungs-Verifizierbarkeit (WV.2), der bedingten Einmaligkeits-Verifizierbarkeit (EV.2) sowie der kontinuierlichen Korrektheits-Verifizierbarkeit (KV.1) zu verifizieren.

WAHLGEHEIMNIS Das Wahlgeheimnis soll unter anderem durch die Funktionstrennung sichergestellt werden, welche durch den Einsatz der verschiedenen Server realisiert wird. Zusätzlich werden alle Stimmzettel mit Hilfe des öffentlichen Schlüssels der Auszählungskomponente verschlüsselt. Der zugehörige private Schlüssel wird lediglich auf der Auszählungskomponente selbst gespeichert. Diese wird im Idealfall offline betrieben und der Datentransfer findet lediglich unidirektional in Form von gebrannten und finalisierten DVDs von den drei Servern *BBS*, *ERS* und *VS* zur Auszählungskomponente statt, ein elektronischer Rückkanal existiert zumindest im theoretisch beschriebenen Systemaufbau nicht.

Hier ist es wichtig zu erwähnen, dass diese Sicherheitsvorkehrungen das Wahlgeheimnis lediglich retrospektiv schützen. Wenn ein Angreifer *ERS* und *BBS* jedoch in Echtzeit kontrolliert, können durchaus Rückschlüsse auf die Wahl einer einzelnen Person gezogen werden, da die Identität des Wählers über den Token dem jeweiligen Stimmzettel zugeordnet werden kann. Helfen könnte hier, wenn der Stimmzettel bereits auf dem vom Wähler benutzten Computersystem mit *TC*s öffentlichem Schlüssel verschlüsselt und somit nie unverschlüsselt an *BBS* übertragen werden würde. Bei der üblichen Nutzung als Web-Applikation wäre dafür jedoch z. B. JavaScript notwendig, was wiederum andere Sicherheitsprobleme mit sich bringen würde.

Falls der Angreifer Kontrolle über das zur Wahl benutzte Computersystem besitzt, ist das Wahlgeheimnis ebenfalls sofort gebrochen. In diesem Fall würde es logischerweise auch nicht helfen, den Stimmzettel vor der Übertragung mit Hilfe von *TC*s öffentlichem Schlüssel zu verschlüsseln.

Als Maßnahme gegen nachträglich Stimmmanipulation werden vom Urnenserver jeweils immer 30 abgegebene Stimmen in zufälliger Reihenfolge als Block gespeichert. Von jedem Block wird nach dessen Vervollständigung ein Prüfwert berechnet, der vom Urnenserver zusammen mit der Signatur des vorherigen Blocks digital signiert und an das elektronische Wählerverzeichnis übermittelt wird [36, 37]. Zwar dient diese Maßnahme tatsächlich der Verhinderung nachträglicher Stimmmanipulation, allerdings kann das elektronische Wählerverzeichnis durch zeitliche Korrelation in besonderen Fällen Rückschlüsse auf die Wahl des Wählers treffen, z. B. wenn alle Wähler eines Blockes die selbe Wahl getroffen haben. In diesem Fall wäre das Wahlgeheimnis gebrochen.

Zusammenfassend bietet Polyas unter Anlegung der strengen Maßgaben für Wahlen erster Ordnung keinen ausreichenden Schutz des Wahlgeheimnisses.

NICHT-ERPRESSBARKEIT Bei der Nutzung von Polyas ist lediglich eine teilweise Nicht-Erpressbarkeit gegeben. Zwar kann ein potentieller Erpresser aufgrund der Informationen, die das Wahlsystem bereitstellt, nicht herausfinden, ob der Wähler gewählt hat oder nicht (Kriterium AA.2). Ein Abwesenheitsangriff wird dadurch, wenn auch nicht verhindert, zumindest erschwert. Er kann den Wahlvorgang aufgrund der fehlenden Möglichkeit, seine Stimme nachträglich zu ändern (*Re-Voting*), jedoch z. B. aufzeichnen lassen oder aber der Wahl selbst beiwohnen. Auch einer erzwungenen Herausgabe der für die Wahl notwendigen Credentials ist möglich.

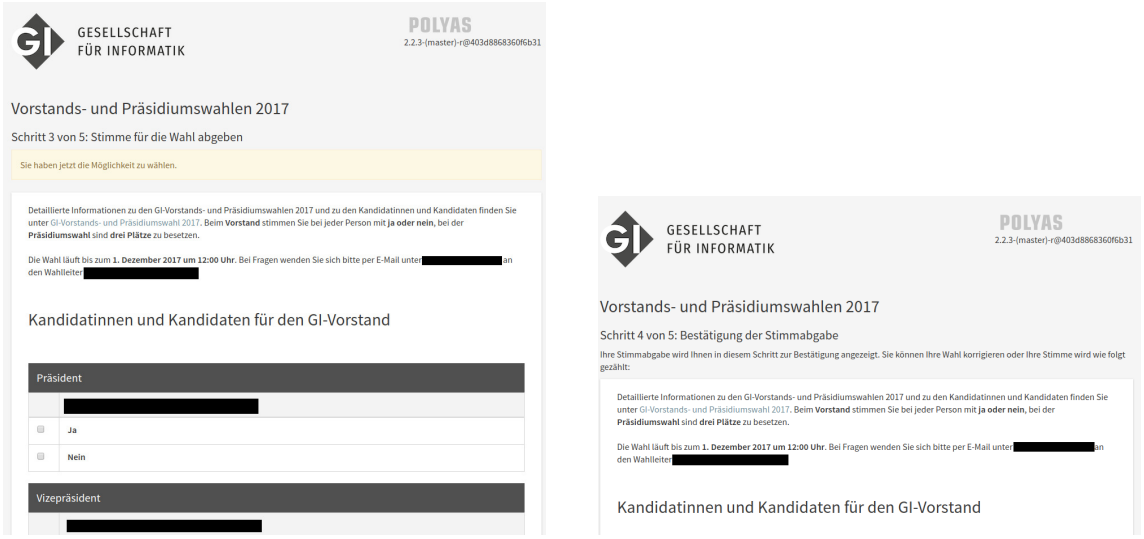
ROBUSTHEIT Zur Wahrung des Wahlgeheimnisses und zum Schutz vor Manipulationen ist ein Fernzugriff auf *BBS*, *ERS* und *VS* jeweils nur unter Eingabe zweier verschiedener und für jede Maschine abweichender Tokens möglich. Diese insgesamt sechs verschiedenen Tokens werden während der Pre-Wahlphase auf sechs Mitglieder des Wahlkomitees verteilt. Da der Zugriff auf die Komponenten kritisch ist und es keinerlei Threshold-Mechanismen gibt, könnte hier von Einzelpersonen ein Angriff auf die erfolgreiche Auszählung der Wahl durchgeführt werden. Da die Tokens jedoch lediglich für

den Remote-Zugriff kritisch sind, kann im Ernstfall nach wie vor eine Auszählung vor Ort ausgeführt werden. Kritischer ist die doppelte Verschlüsselung des privaten Schlüssels der Auszählungskomponenten *TC*, der zur Entschlüsselung der mit dem zugehörigen öffentlichen Schlüssel verschlüsselten Stimmen notwendig ist. Da auch hier keine Threshold-Verschlüsselung eingesetzt wird, kann einer der beiden Wahloffiziellen die Auszählung des Ergebnisses verhindern. Polyas enthält keine wirksamen Mechanismen zur Gewährleistung innerer Robustheit (Kriterium RI.1).

BENUTZBARKEIT DER WAHL Die Benutzbarkeit der Wahl wurde am Beispiel der Vorstands- und Präsidiumswahlen 2017 der GI getestet. Abgesehen von der Mitgliedsnummer und der PIN, die per Brief zugestellt wird und zur Authentifizierung dient, ist keine komplizierte Eingabe nötig. Es handelt sich bei dieser Verwendung von Polyas deshalb um *Klick Voting* mit einmaliger Teilnahme und ohne Zuhilfenahme von Hilfsmitteln (Kriterium BW.1.1).

(a) Anmeldung mit Mitgliedsnummer und PIN. **(b)** Anmeldung erfolgreich.

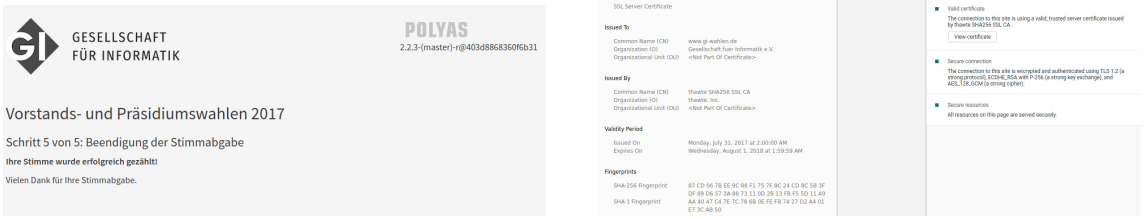
Abbildung 21: Anmeldung zu den GI Vorstands- und Präsidiumswahlen 2017 (Polyas) [161].



(a) Digitaler Stimmzettel mit KandidatInnen. **(b)** Bestätging der Stimmabgabe.

Abbildung 22: GI Vorstands- und Präsidiumswahlen 2017 (Polyas) [161].

BENUTZBARKEIT DER INDIVIDUELLEN VERIFIZIERUNG Die Möglichkeit zur individuellen Verifizierbarkeit vor der Auszählung wurde im Falle der GI Wahlen 2017 nicht benutzt, weshalb sie nicht in der Praxis gestestet werden konnte. Für den Fall, dass diese Funktion aktiviert wird, sollten theoretisch jedoch keine anderen Schritte nötig sein, als die bereits beschriebenen, also der Vergleich des



(a) Beendigung der Stimmabgabe. **(b)** Überprüfung des eingesetzten Zertifikats.

Abbildung 23: Beendigung der GI Vorstands- und Präsidiumswahlen 2017 (Polyas) [161].

Bestätigungscodes und im Falle, dass der Code nicht übereinstimmt, die Übermittlung des Offline-Authentifizierungscodes. Die Eigenschaft BV.1 ist deshalb als erfüllt anzusehen.

5.3 DU-VOTE

Du-Vote steht für „**D**evices that are **U**ntrusted used to **Vote**"und ist bisher nur ein theoretisches Verfahren, welches folglich noch nicht für eine Wahl benutzt wurde. Die Idee hinter diesem Verfahren ist, dass der Wahlvorgang auf die einzelnen Komponenten des Systems verteilt wird. Die Entscheidung des Wählers wird dann vom System aus den Informationen von allen am Wahlvorgang beteiligten Geräten berechnet. Ziel des Protokolls ist es, das Wahlgeheimnis und die Ende-zu-Ende Verifizierbarkeit (d. h. individuelle und universelle Verifizierbarkeit) mittels einer dynamisch erstellten Code Page zu gewährleisten. Die Auszählung der Stimmen kann durch die Entschlüsselung der homomorphen Kombination der Stimmen oder durch verifiable Re-Encryption Mix-Nets durchgeführt werden [148].

5.3.1 System Aufbau

Du-Vote besteht aus vier Komponenten: einem Server S, einem Wahlcomputer P, einem Hardware-Token H und einem Bulletin Board BB. Neben diesen Komponenten gibt es Teilnehmer, welche mit dem System interagieren. Zum einen ist dies natürlich der Wähler V sowie eine Menge von sogenannten Decryption Tellers $T = \{T_1 \dots T_m\}$. Im Folgenden sollen die Aufgaben und Funktionsweisen der einzelnen Komponenten und Teilnehmer beschrieben werden.

DECRYPTION TELLER: Die Decryption Teller $\{T_1 \dots T_m\}$ sind für die Erstellung des Schlüsselpaars zuständig, mit dem die Stimmen verschlüsselt werden. Ihnen muss prinzipiell nicht vertraut werden, da der Entschlüsselungs-Schlüssel zwischen den Decryption Tellers aufgeteilt ist. Dazu wird ein verteiltes ElGamal-Verschlüsselungs-Protokoll verwendet, welches in Unterabschnitt A.1.3 beschrieben wird.

HARDWARE-TOKEN: Aufgabe des Hardware-Tokens ist es, den Code, welcher die Wahl von V codiert darstellt, entgegen zu nehmen und diesen erneut zu verschlüsseln. Dazu ist auf dem Hardware-Token ein Schlüssel $K = y^k$ sowie h, ein weiterer Generator der Gruppe G, gespeichert. y ist der öffentliche Schlüssel der Decryption Tellers und k ist eine, für jeden Hardware-Token zufällig gewählte, Zahl, wobei diese schon bei der Produktion des Hardware-Tokens generiert wird.

Das Hardware-Token hat keine Verbindung zum Internet. Alle Interaktionen des Hardware-Tokens mit den anderen Komponenten des Wahlsystems erfolgen durch Eingabe des Wählers vom Wahlcomputer in das Token und umgekehrt, weshalb das Hardware-Token zur Eingabe eine Tastatur und zur Ausgabe ein Display besitzt. Um diese Interaktion so benutzerfreundlich wie möglich zu gestalten,

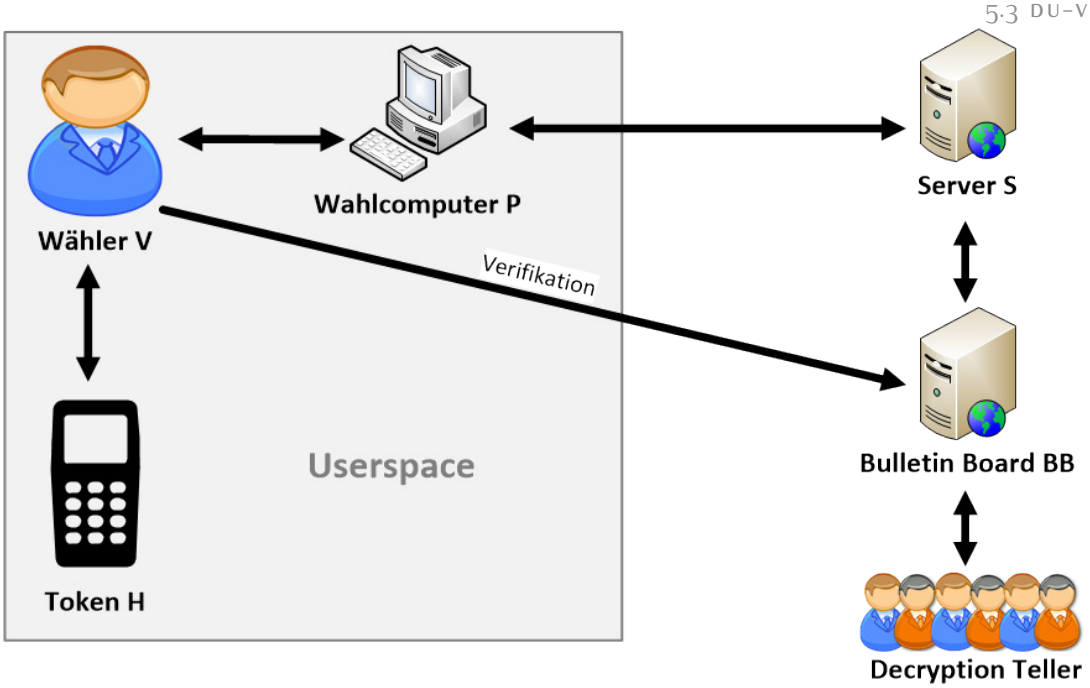


Abbildung 24: Systemaufbau Du-Vote in Anlehnung an [148].

werden die Ein- sowie Ausgaben verkürzt. Die Verkürzung hängt dabei von dem Sicherheitsparameter $\mathcal{K}$ ab. Grewal et al. [148] empfehlen den Parameter $\mathcal{K} = 4$ zu wählen.

Die Verschlüsselung des Hardware-Tokens läuft folgendermaßen ab: Der Wähler gibt eine dezimale Zahlenfolge d in das Token ein, welche als Dezimalzahl interpretiert wird. Getriggert durch eine Bestätigungstaste berechnet das Token Kh^d. Das Ergebnis der Berechnung wird wieder als Dezimalzahl interpretiert und es werden nur $\mathcal{K}$ Ziffern des Ergebnisses ausgegeben. Prinzipiell ist es egal, welche Ziffern der Berechnung ausgegeben werden, diese müssen nur vorab festgelegt werden. Im Folgenden wird die Ausgabe des Tokens unter der Eingabe d mit $H(d)$ bezeichnet und davon ausgegangen, dass die letzten vier, geringwertigsten Ziffern der Berechnung ausgegeben werden. Siehe dazu auch Algorithmus 1.

Algorithm 1 Hardware-Token $H(d)$.

Input: d Dezimalzahl
Output: die letzten, geringwertigsten $\mathcal{K}$ Ziffern von Kh^d
1: $c = Kh^d$
2: $c^* =$ die letzten, geringwertigsten $\mathcal{K}$ Ziffern von c
3: **return** c^*

WAHLCOMPUTER: Der Wahlcomputer verschlüsselt alle möglichen Stimmen und generiert die Code Page. Die Codes der Code Page bestehen aus den letzten $\mathcal{K}$ Ziffern der verschlüsselten Stimmen. Außerdem ist der Wahlcomputer dafür zuständig, die Ausgabe des Hardware-Tokens, welche die Wahl des Wählers codiert, an den Server weiterzuleiten.

SERVER: Der Server ist für die Authentifizierung des Wählers sowie die Überprüfung der Berechnungen des Wahlcomputers zuständig. Alle Informationen, die dafür benötigt werden, werden auf dem Bulletin Board veröffentlicht, sodass jeder diese Überprüfung nachvollziehen kann. Außerdem ist es seine Aufgabe, die Wahl des Wählers aus den Informationen des Hardware-Tokens und den Informationen des Wahlcomputers zu berechnen, sowie diese neu zu verschlüsseln und auf dem Bulletin

Board zu veröffentlichen. Die Korrektheit seiner eigenen Berechnungen wird durch mehrere nicht-interaktive Zero-Knowledge-Beweise gewährleistet [148].

5.3.2 Ablauf der Wahl

Die Wahl mit Du-Vote gliedert sich in vier Phasen. Die Vorbereitungsphase, in welcher alle benötigten Keys und Parameter erzeugt und verteilt werden. Die Wahlphase, in der die Wahl stattfindet sowie die Auszählungs- und die Auditphase. In den folgenden Kapiteln werden die einzelnen Phasen des Protokolls beschrieben, sodass am Ende der Ablauf des Protokolls klar sein sollte.

Vorbereitungsphase

Zu Beginn der Vorbereitungsphase wählen die Decryption Teller zufällig eine große Primzahl p sowie zwei Generatoren g und h der Gruppe $\mathbb{Z}_p$. Anschließend generiert jeder Decryption Teller $T_j \in \{T_1 \dots T_m\}$ zufällig einen privaten Schlüssel z_j und berechnet den Teil seines öffentlichen Schlüssels $y_j = g^{z_j}$. Anschließend wird y_j veröffentlicht, und der öffentliche Schlüssel $y = \prod_{j=1}^{m} y_j$ kann berechnet werden.

Außerdem bekommen alle V eine Wähler-ID und ein Passwort, um sich beim Server S zu authentifizieren. Du-Vote benötigt für diese Verteilung zwar kein explizites Protokoll, durchaus aber einen sicheren Kanal, normalerweise wird hierfür der Postweg gewählt. Neben dem Passwort und der Wähler-ID benötigt V zusätzlich einen Hardware-Token. Dazu wird das Geheimnis k zufällig generiert und $K = y^k$ auf dem Token gespeichert. Gleichzeitig wird auf dem Server eine Zuordnung zwischen k und der Wähler-ID gespeichert. Da K geheim gehalten werden muss, wird davon ausgegangen, dass der Token H über einen privaten Kanal ausgeliefert wird.

Kurz vor Beginn der Wahl wird die Kandidatenliste $\{c_0, c_1, \dots c_n\}$ auf dem Bulletin Board veröffentlicht. Des Weiteren wird eine Nonce I für die Initialisierung der Wahl benötigt, welche öffentlich verifizierbar und nicht vorhersagbar ist. I wird zusammen mit dem Algorithmus zur Generierung der Nonce I ebenfalls auf dem Bulletin Board veröffentlicht. Die Generierung der Nonce kann beispielsweise auf Basis von Börsendaten geschehen [162]. Diese sind nicht vorhersagbar, in ausreichender Menge verfügbar und öffentlich zugänglich, sodass jeder mit diesen Daten und Kenntnis über den Algorithmus die Nonce I verifizieren kann.

Wichtig ist zu beachten, dass I erst nach der Verteilung der Hardware-Tokens generiert werden darf, da sonst die Integrität der Wahl gefährdet wird. Dies ist auch der Grund, warum H kein Input-Interface außer einer Tastatur haben darf. Falls H eine Verbindung zum Internet hätte, könnte H die Nonce vom Bulletin Board erfahren [148].

Wahlphase

Die Wahlphase gliedert sich in die vier Abschnitte

- Generierung der Code Page auf S,
- Verschlüsselung des Wahlcodes auf H,
- Generierung der Stimme aus dem Code auf S und
- Beweis für korrektes Vorgehen von S und Veröffentlichung der Stimme auf BB.

Im Folgenden werden diese Abschnitte zusammen mit der Interaktion des Wählers erklärt.

GENERIERUNG DER CODE PAGE V authentifiziert sich durch die Eingabe seines Passworts und der VID (Wähler-ID) über P bei S. P lädt sich die Kandidatenliste $C = \{c_0, c_1, \dots c_{n-1}\}$ sowie I von BB herunter. Nun generiert P eine Code Page, wie in Abbildung 25 dargestellt. Im Folgenden wird die Generierung der Code Page genauer beschrieben (siehe dazu auch Algorithmus 2).

Kandidat	Spalte A	Spalte B
c_1	a_1	b_1
c_2	a_2	b_2
$\vdots$	$\vdots$	$\vdots$
c_n	a_n	b_n

Abbildung 25: Du-Vote Code Page.

- P generiert dazu eine Menge $\{d_0, \dots d_{n-1}, d_n, \dots, d_{2n-1}\}$ an pseudozufälligen Codes, sodass für jeden Kandidaten zwei Codes zur Verfügung stehen. Jeder dieser Codes besteht aus genau $\mathcal{K}$ Ziffern und hängt von der VID sowie von I ab. Dies geschieht, indem P einen Pseudozufallsgenerator mit einem $hash(I, VID)$ seeded und eine festgeschriebene Anzahl an Bits anfordert bis $2n$ unterschiedliche Codes entstehen.
- Anschließend wählt P zufällig ein $1 \leqslant \alpha \leqslant n$ und $n+1 \leqslant \beta \leqslant 2n$. Dann wird für jeden Kandidaten $c_i \in C$ der Code $a_i = d_{(\alpha+i) \mod n}$ und der Code $b_i = a_{n+((\beta+i) \mod n)}$ zugeordnet. Das bedeutet, jeder Kandidat bekommt zwei Codes, wobei einer aus der unteren Hälfte der Codes stammt und der andere aus der oberen Hälfte. Die Zuordnung beginnt bei der zufällig gewählten Zahl und macht einen Wrap-Around, falls die Zuordnung die Anzahl der Kandidaten übersteigt. Nach dieser Zuordnung besitzt jeder Kandidat die Werte (c_i, a_i, b_b). Dieses Tupel stellt später eine Zeile der Code Page dar.
- Im nächsten Schritt werden zu den Codes verschlüsselte Werte der Kandidaten gesucht, sodass die letzten $\mathcal{K}$ Dezimalstellen des verschlüsselten Kandidaten dem ihm zugeordneten Code entsprechen. Dies geschieht durch eine exponentielle ElGamal-Verschlüsselung, wie in Unterabschnitt A.1.3 beschrieben, indem für alle Kandidaten c_i ein zufälliges r gewählt wird und anschließend $A_i = (x_1, x_2) = enc(c_i, y, r+j) = (g^{r+j}, y^{r+j} g^m)$ für alle $j \in \{0, 1, 2, \dots\}$ bis die letzten $\mathcal{K}$ Stellen des Chiffrats dem Code $a_{i,a}$ entsprechen. Ist dies gelungen, wird das Gleiche für alle b_i durchgeführt. Sei B_i das Chiffrat, das die Bedingung b_i erfüllt, dann stehen für jeden Kandidaten die Informationen $(c_i, a_i, A_i, a_i, A_i)$ bereit.
- Im folgenden Schritt kann P die Code Page darstellen. In der ersten Spalte wird der Kandidat $c_i \in C$ alphabetisch sortiert eingetragen, in der zweiten Spalte wird a_i und in der dritten Spalte b_i eingetragen (siehe Abbildung 25). Zusätzlich zur Code Page wird die Stimmen-ID $SID = hash(A_{lex}, B_{lex})$ ausgegeben, wobei A_{lex} und B_{lex} der lexikografischen Ordnung aller A_is bzw. B_is entspricht. Die Stimmen-ID ist deshalb wichtig, da sich P dadurch auf die Verschlüsselung der Kandidaten gegenüber V festlegt.

VERSCHLÜSSELUNG DES WAHLCODES Nachdem die Code Page mit der SID angezeigt wird, kann V seine Wahl treffen. Dazu überprüft V, ob die Code Page richtig angezeigt wird, d. h. ob alle Kandidaten in alphabetischer Reihenfolge angezeigt werden und jeder zwei Codes besitzt. Anschließend wählt V echt zufällig, welche der Spalten zur Bestimmung der Code-Reihenfolge dient, die sogenannte Audit-Spalte. Aus der anderen Spalte wählt V seinen Kandidaten aus, diese wird deshalb Wahl-Spalte genannt. Die Auswahl der Audit- und Wahl-Spalte wird durchgeführt, indem V eine Münze wirft und falls das Ergebnis des Münzwurfs Kopf ist, ist $\{a_0, a_1, \dots, a_{n-1}\}$ die Audit-Spalte und $\{b_0, b_1, \dots, b_{n-1}\}$ die Wahl-Spalte. Falls das Ergebnis des Münzwurfs Zahl ist, ist dies entsprechend umgekehrt. Angenommen der Münzwurf ergibt Kopf, dann trifft V seine Wahl aus $x* \in \{b_0, b_1, \dots, b_{n-1}\}$ und gibt anschließend $a_0 \| a_1 \| \cdots \| a_{n-1} \| x^*$ in das Hardware-Token ein. Eine entsprechende Anleitung wird auf dem Bildschirm des Wahlcomputers angezeigt. Das Hardware-Token berechnet nun aus der Eingabe, wie in Unterabschnitt 5.3.1 beschrieben, $c^* = H(d)$. V tippt c^* in P ein und notiert sich für die spätere Verifikation die SID sowie c^*.

Algorithm 2 Generierung der Code Page.

Input: $I, VID, C = \{c_1, c_2 \dots c_n\}$
Output: $\{a_0, \dots a_{n-1}, a_n, \dots, a_{2n-1}\}$
1: $a = prf(hash(I, VID))$
2: generiere aus a, $2n$ disjunkte Codes $\{d_0, \dots d_{n-1}, d_n, \dots, d_{2n-1}\}$.
3: $\alpha \in_R \{1 \dots n\}$, $\beta \in_R \{n+1 \dots 2n\}$
4: $CS = \emptyset$
5: **for all** $c_i \in C$ **do**
6: $\quad a_i = d_{(i+\alpha \mod n)}$
7: $\quad b_i = d_{(n+(i+\beta \mod n))}$
8: $\quad CS = CS \cup (c_i, a_{i,a}, a_{i,b})$
9: **end for**
10: $A = \emptyset$, $B = \emptyset$
11: **for all** $(c_i, a_i, b_i) \in CS$ **do**
12: $\quad r_1 \in_R \mathbb{Z}_p^*$, $r_2 \in_R \mathbb{Z}_p^*$
13: $\quad$ finde ein j, sodass die letzten, geringwertigsten $\mathcal{K}$ Ziffern von $A_i = enc(c_i, y, r+j)$, a_i entsprechen.
14: $\quad$ finde ein k, sodass die letzten, geringwertigsten $\mathcal{K}$ Ziffern von $B_i = enc(c_i, y, r+k)$, b_i entsprechen.
15: $\quad A = A \cup \{A_i\}$, $B = B \cup \{B_i\}$
16: **end for**

Um den Wahlvorgang abzuschließen, sendet P die Daten A_{lex}, B_{lex} und c^* an P. Ziel der Umsortierung ist es, α bzw. β vor S geheimzuhalten. Es wird also die Verbindung zwischen dem Code bzw. dem Chiffrat und dem Kandidatennamen gekappt. Dies ist für die Gewährleistung des Wahlgeheimnisses unbedingt erforderlich, da bei Nichteinhaltung dieses Vorgehens S eine Zuordnung zwischen x^* und dem Kandidaten herstellen kann, wie im folgenden Abschnitt gezeigt wird.

GENERIERUNG DER STIMME AUS DEM CODE Der Server prüft im ersten Schritt, ob sich P ordnungsgemäß verhalten hat, indem er berechnet, ob die Codes entsprechend der vorgegebenen Berechnungen erstellt wurden.

Algorithm 3 Überprüfung der Du-Vote Codes.

Input: $I, VID, C = \{c_1, c_2 \dots c_n\}, A_{lex}, B_{lex}$
Output: $D = \{d_0, \dots d_{n-1}, d_n, \dots, d_{2n-1}\}$
1: $a = prf(hash(I, VID))$
2: Generiere aus a, $2n$ disjunkte Codes $\{d_0, \dots d_{n-1}, d_n, \dots, d_{2n-1}\}$.
3: **for all** $z_1 \in \{d_0, \dots d_{n-1}\}$, $z_2 \in \{d_n, \dots, d_{2n-1}\}$ **do**
4: $\quad$ **if** $z_1 \notin A_{lex} \vee z_2 \notin B_{lex}$ **then**
5: $\quad\quad$ **return** Abbruch
6: $\quad$ **end if**
7: **end for**
8: Veröffentliche A_{lex}, B_{lex}, c^* auf BB

Hierfür berechnet der Server den pseudozufälligen Bitstream (initialisiert mit I und VID) und generiert daraus die Codes $\{d_0, \dots d_{n-1}, d_n, \dots, d_{2n-1}\}$. Dann wird überprüft, ob für jeden der Codes $\{d_0, d_1 \dots d_{n-1}\}$ ein Element in A_{lex} und für jeden Code $\{d_n, d_{n+1} \dots d_{2n}\}$ ein Element in B_{lex} existiert, welches jeweils die Bedingung erfüllt, dass die letzten $\mathcal{K}$ Stellen gleich sind. Stellt S hier eine Unstimmigkeit fest, bricht er den Vorgang ab. Ist hingegen alles korrekt, werden A_{lex}, B_{lex} und c^* auf BB veröffentlicht. Siehe dazu auch Algorithmus 3.

Als nächstes rekonstruiert S die abgegebene Stimme aus dem Code c^*, indem er alle möglichen Kombinationen berechnet und überprüft, ob die Berechnung c^* ergibt. Dazu muss S die folgende Menge berechnen

$$\begin{array}{lll}
H(d_0\|d_1\|\cdots\|d_{n-1}\|d_n), & H(d_{n-1}\|d_0\|\cdots\|d_{n-2}\|d_n), & \ldots, H(d_2\|d_3\|\cdots\|d_1\|d_n),\\
\vdots & \vdots & \vdots\\
H(d_0\|d_1\|\cdots\|d_{n-1}\|d_{2n-1}), & H(d_{n-1}\|d_0\|\cdots\|d_{n-2}\|d_{2n-1}), & \ldots, H(d_2\|d_3\|\cdots\|d_1\|d_{2n-1})\\
H(d_n\|d_{n+1}\|\cdots\|d_{2n-1}\|d_1), & H(d_{2n-1}\|d_n\|\cdots\|d_{2n-2}\|d_1), & \ldots, H(d_{n+2}\|d_{n+3}\|\cdots\|d_{n+1}\|d_1)\\
\vdots & \vdots & \vdots\\
H(d_n\|d_{n+1}\|\cdots\|d_{2n-1}\|d_{n-1}), & H(d_{2n-1}\|d_n\|\cdots\|d_{2n-2}\|d_{n-1}), & \ldots, H(d_{n+2}\|d_{n+3}\|\cdots\|d_{n+1}\|d_{n-1})
\end{array}$$

und überprüfen, ob eines dieser Elemente c^* entspricht. Diese Berechnung von $H(x) = Kh^x$ setzt Kenntnis von K voraus. Wie in Unterabschnitt 5.3.2 erklärt wurde, hat der Server Zugriff auf die ks aller Wähler. Nach diesen Berechnungen veröffentlicht S die Audit-Spalte A oder B (Spalte, welche V komplett eingegeben hat) in der angezeigten Reihenfolge auf dem Bulletin Board. Außerdem findet S den Code x^*, mit welchem er die verschlüsselte Stimme berechnen kann. Siehe dazu Algorithmus 4.

Algorithm 4 Bestimmen von x^*.

Input: D, c^*
Output: Offset, AuditSpalte, WahlSpalte, $x^* = z_2$

for all $z_1 \in \{d_0, \ldots d_{n-1}\}, z_2 \in \{d_n, \ldots, d_{2n-1}\}$ **do**
  for $i = 0$ to $n-1$ **do**
    if $H(d_{i \mod n}\|d_{(i+1) \mod n}\|\cdots\|d_{(i+n-1) \mod n}\|z_1) = c^*$ **then**
      Offset= i, AuditSpalte=A, WahlSpalte=B ,$x^* = z_1$
      Abbruch aller Schleifen
    end if
    if $H(d_{n+i \mod n}\|d_{n+(i+1) \mod n}\|\cdots\|d_{n+(i+n-1) \mod n}\|z_2) = c^*$ **then**
      Offset= i, AuditSpalte=B, WahlSpalte=A, $x^* = z_2$
      Abbruch aller Schleifen
    end if
  end for
end for
Veröffentliche AuditSpalte mit Offset auf BB

Zur weiteren Überprüfung von P fordert S von P alle zufällig gewählten Werte $x_0, x_1, \ldots x_{n-1}$ der ElGamal-Verschlüsselung für die Audit-Spalte an. Um zu vermeiden, dass S die falsche Spalte anfordert, überprüft P, ob S die Codes der Audit-Spalte auf dem Bulletin Board veröffentlicht hat. Ist dies der Fall, so sendet P die zufällig gewählten Werte der ElGamal-Verschlüsselung in der angezeigten Reihenfolge an S. Nun überprüft S, ob die verschlüsselten Werte korrekt erstellt wurden. Dies geschieht, indem S alle Kandidaten erneut verschlüsselt und überprüft, ob das berechnete Chiffrat mit dem entsprechenden Chiffrat der Audit-Spalte übereinstimmt.

Schlägt eine dieser Überprüfungen fehl, so kann S sicher sein, dass entweder P oder H sich nicht an den spezifizierten Ablauf gehalten oder V sich bei der Eingabe vertippt hat. Folglich bricht S in so einem Fall den Vorgang ab.

Sind alle Überprüfungen korrekt, so kann S davon ausgehen, dass sich die Teilnehmer an den vorgeschriebenen Ablauf gehalten haben. S berechnet aus dem Code $x*$ nun die verschlüsselte Stimme x, indem er das Chiffrat wählt, bei welchem die letzten $\mathcal{K}$ Ziffern übereinstimmen. $x = (x_1, x_2)$ wird erneut verschlüsselt, indem $E = xg^r = (x_1 g^r, x_2 g^r)$ für $r \in_R \mathbb{Z}_p^*$ berechnet wird. x wird nun auf dem Bulletin Board veröffentlicht. Die erneute Verschlüsselung ist deshalb notwendig, da der Wahlcomputer P sonst in der Lage wäre, das Wahlgeheimnis zu brechen, da er zum Chiffrat den zugehörigen Klartext kennt. Dieses Vorgehen ist in Algorithmus 5 formalisiert.

Algorithm 5 Re-Encryption.

Input: Offset, AuditSpalte, WahlSpalte, $x^* = z_2$
Output: E

1: Verlange $x_0, x_1, \ldots, x_{n-1}$ von AuditSpalte von P.
2: Veröffentliche $x_0, x_1, \ldots, x_{n-1}$ auf BB
3: **if** AuditSpalte mit Offset enthält alle Kandidaten genau einmal = false **then**
4: **return** Abbruch
5: **end if**
6: Finde $x \in$ WahlSpalte sodass letzte $\mathcal{K}$ stellen von $x = x^*$
7: $r \in_R \mathbb{Z}_p^*$
8: $E = (x_1 g^r, x_2 g^r)$
9: Veröffentliche E auf BB

Anschließend führt der Server einen interaktiven Zero-Knowledge-Proof aus, um zu beweisen, dass er sich selbst an den Protokollablauf gehalten hat. Dieser Zero-Knowledge-Proof wird auf dem Bulletin Board veröffentlicht, sodass jeder diesen Beweis nachvollziehen kann. Dieses Vorgehen wird im nächsten Kapitel beschrieben.

Zum Schluss loggt sich V vom Server aus und überprüft mittels der Stimmen-ID, ob die Wahl korrekt beim Bulletin Board eingegangen ist. Das Problem ist hierbei, dass die Verifikation von einem vertrauenswürdigen Computer aus geschehen oder an eine vertrauenswürdige Person delegiert werden muss. Auf jeden Fall sollte die Verifikation nicht vom Wahlcomputer aus vorgenommen werden [148].

S beweist seine Vertrauenswürdigkeit

Der Beweis dafür, dass der Server sich an den Protokollablauf gehalten hat, besteht aus drei verschiedenen Signature-Proof-of-Knowledges, welche über einer Ring Signatur verwendet werden. Der Server beweist, dass er die richtige verschlüsselte Stimme x aus dem Code c berechnet hat (Proof-of-Selection). Anschließend beweist er, dass er die verschlüsselte Stimme wieder korrekt verschlüsselt hat (Proof-of-Re-Encryption) und als letztes beweist der Server, dass die Wiederverschlüsselung auf x angewandt wurde (Proof-of-Re-Encryption of the Chipertext).

PROOF-OF-SELECTION In diesem Teil muss der Server beweisen, dass er die richtige Stimme $x = (x_1, x_2)$ aus c berechnet hat. Die Idee ist, dass der Server dazu einen Tupel $D = (D_1, D_2) = (g^d, y^d h^{x_2})$ für ein beliebiges d und x_2 berechnet. Anschließend beweist der Server die Struktur von D, indem er zeigt, dass er in Kenntnis eines d ist, sodass $(D_1, D_2) = (g^d, y^d h^{x_2})$ gilt.

Für die genauere Erklärung wird angenommen, dass V, $b_0 || b_1 || \cdots || b_{n-1} || x*$ wobei $x* \in a_0, a_1, \ldots, a_n$ in H eingegeben hat und P $C^* = H(b_0 || b_1 || \cdots$
$|| b_{n-1} || x^*)$ an S gesendet hat. S geht dazu wie folgt vor:

- S berechnet aus C^* den vollständigen Code $C = y^k h^{b_0 || b_1 || \cdots || b_{n-1} || x^*}$. Dies kann von V überprüft werden.
- S berechnet $C_{x^*} = C / h^{(b_0 || b_1 || \cdots || b_{n-1}) \cdot 10^{\mathcal{K}}} = y^k h^{x^*}$. Dies kann von jedem geprüft werden, da der Ciphertext der Audit-Spalte $B = \{B_1, B_2 \ldots B_n\}$ sowie C öffentlich sind.
- Als nächstes wird $x' = \frac{(x_2 - x^*)}{10^{\mathcal{K}}}$ berechnet, was einfach dem Abschneiden der letzten $\mathcal{K}$ Stellen von x_2 entspricht, da x^* der zugehörige Code von x ist, gilt nach dem beschriebenen Vorgehen, dass $x_2 = x' || x^*$ ist.
- Aus x' wird $C_{x'} = y^{r'} h^{10^{\mathcal{K}} x'} = y^r h^{a_{0,b} || \cdots || a_{n,b} \cdot 10^{\mathcal{K}}}$ berechnet, wobei r' eine Zufallszahl ist und $10^{\mathcal{K}} x'$ dem x_2 entspricht, bei dem die letzten $\mathcal{K}$ Ziffern 0 sind. Zur Verifikation wird ein Proof-of-Knowledge für x' und r' geführt und auf dem Bulletin Board veröffentlicht. Im Folgenden wird der Beweis mit $P_{x'}$ bezeichnet.

- Nun kann aus diesen beiden Werten

$$\begin{aligned} D = (D_1, D_2) &= (g^d, C_{x*} C_{x'}) \\ &= (g^d, y^k h^{x^*} y^{r'} h^{10^{\mathcal{K}} x'}) \\ &= (g^d, y^{k+r'} h^{10^{\mathcal{K}} x' + x^*}) \\ &= (g^d, y^d h^{x_2}) \end{aligned}$$

  berechnet werden, wobei $d = k + r'$.

Durch diese Konstruktion kann S beweisen, dass er ein d kennt, sodass $(D_1, D_2) = (g^d, y^d h^{x_2})$. Ist dies der Fall, kann jeder verifizieren, dass S für die Berechnung von (D_1, D_2) die verifizierbaren Werte C_{x^*} und $C_{x'}$ verwendet hat, wobei S diese nur verwenden kann, falls er auch C und x_2 korrekt berechnet hat. Will der Server dennoch betrügen, muss P ein $d' = x'' + r''$ und $x_2' \in \{A_{1,2}, A_{2,2}, \dots A_{n,2}\}$ sein, sodass

$$y^{d'} h^{x_2'} = C_{x^*} C_{x'} = y^d h^{x_2}.$$

Dies ist genau so schwer, wie den diskreten Logarithmus von y zur Basis h zu berechnen [16].

Es bleibt noch zu erwähnen, dass d unbedingt geheim gehalten werden muss, da aus d, dem öffentlichen Schlüssel y und D_2 (was aus $C, C_{x'}, B$ zu berechnen ist) der Wahlcomputer x_2 und somit über x auch die Stimme von V herausfinden kann. Dies kann erfolgen, indem er $y^d h^{x_2'}$ für alle $x_2' \in \{A_{1,2}, A_{2,2}, \dots A_{n,2}\}$ berechnet und prüft, ob die Berechnung gleich D_2 ist. Folglich wird ein Signature-Proof-of-Knowledge über einer Ring-Signatur verwendet, indem der Server folgendes beweist:

$$P_D = \mathrm{SPK}\{d | \exists x_2 \in \{A_{1,2}, A_{2,2}, \dots A_{n,2}\} : g^d = D_1 \wedge y^d h^{x_2} = D_2\}$$

Dieser Beweis ist gleichbedeutend mit dem Beweis, dass

$$P_D = \mathrm{SPK}\left\{ d \mid \exists x_2 \in \{A_{1,2}, A_{2,2}, \dots A_{n,2}\} : (gy)^d = \frac{D_1 D_2}{h^{x_2}} \right\}$$

gilt.

PROOF-OF-RE-ENCRYPTION Da der Server die Stimme aus oben beschriebenen Gründen erneut zu verschlüsseln hat, muss er beweisen, dass der Server diese Verschlüsselung durchgeführt hat. Es wird davon ausgegangen, dass der Server aus x ein Chiffrat $E = (E_1, E_2) = (g^e x_1, y^e x_2)$ erstellt, wobei e zufällig gewählt wird.

Um die Struktur von E nachzuweisen, beweist der Server seine Kenntnis eines e. Dies geschieht, indem der Server folgenden Signature-Proof-of-Knowledge über einer doppelten Ring-Signatur durchführt:

$$P_E = \mathrm{SPK}\left\{ e \mid \exists (x_1, x_2) \in \{A_1, A_2, \dots A_n\} : g^e = \frac{E_1}{x_1} \wedge y^e = \frac{E_2}{x_2} \right\}.$$

Offensichtlich zeigt dieser Beweis nicht, dass die Stimme x, sondern nur, dass einer der Kandidaten aus der Wahl-Spalte $\{A_1, A_2, \dots A_n\}$ erneut verschlüsselt wurde. Da S die Wahl-Spalte kennt, kann er zufällig eines der As wählen und dieses verschlüsseln. Deshalb wird er als nächstes einen Beweis durchführen, der den Proof-of-Selection und den Proof-of-Re-Encryption miteinander verbindet.

PROOF-OF-RE-ENCRYPTION OF THE CHIPERTEXT Um zu beweisen, dass die Stimme $x = (x_1, x_2)$ auch tatsächlich zur Berechnung von $E = (E_1, E_2)$ führt, beweist der Server, dass für die Re-encryption das gleiche x_2 verwendet wird, wie bei der Berechnung des D im Proof-of-Selection.

Im Detail berechnet S dazu das Tupel $F = (F_1, F_2)$, welches wie folgt konstruiert ist:

$$\begin{aligned} F = (F_1, F_2) &= (D_1 E_1, D_2 E_2) \\ &= ((g^d) \cdot (g^e x_1), (y^d h^{x_2}) \cdot (y^e x_2)) \\ &= (g^{d+e} x_1, y^{d+e} h^{x_2} x_2) \\ &= (g^f x_1, y^f h^{x_2} x_2) \end{aligned}$$

wobei $f = d + e$. Durch diese Konstruktion ist S in der Lage zu beweisen, dass er ein f kennt, sodass F wie eben beschrieben konstruiert wurde. Dies wird ebenfalls mit einem Signature-based Proof-of-Knowledge über einer doppelten Ring-Signatur realisiert [148]:

$$P_F = \mathrm{SPK}\left\{ f \mid \exists (x_1, x_2) \in \{A_1, A_2, \ldots A_n\} : g^f = \frac{F_1}{x_1} \wedge y^f = \frac{F_2}{h^{x_2} x_2} \right\}.$$

Auszählungsphase

Für die Auszählung erhalten die Decryption Tellers alle verschlüsselten Stimmen vom Server oder vom Bulletin Board. Die Besonderheit ist, dass bei Du-Vote mehrere verschiedene Techniken für die Auszählungen der Stimmen verwendet werden können, weswegen die Auszählung von einem anderen System ausgeführt werden kann. Im Paper werden hier zwei Möglichkeiten erwähnt:

- Homomorphe Kombination der einzelnen Stimmen gefolgt von anschließender Entschlüsselung, wie sie beispielsweise in Helios [76, 163] verwendet wird. Dies ist möglich, da Du-Vote exponentielles ElGamal verwendet (siehe dazu auch Unterabschnitt A.1.3).
- Re-Encryption-Mix gefolgt von anschließender Entschlüsselung, wie es in JCJ bzw. Civitas [29, 53] verwendet wird.

Zusätzlich muss hierbei allerdings die Korrektheit der Stimmenauszählung in irgendeiner Form sichergestellt werden [148].

Verifikationsphase

Nach der Wahl eines Wählers sind folgende Informationen pro Stimme für die Verifikation auf dem Bulletin Board verfügbar:

- Die Wähler-ID sowie die Stimmen-ID,
- die Chiffrate A und B,
- die Zuordnung zwischen Chiffrat und Kandidat für die Audit-Spalte,
- die verwendeten Zufallswerte $r_1, r_2 \ldots r_n$ für die Audit-Spalte,
- die mehrfach verschlüsselte Stimme E sowie
- der Beweis für die Korrektheit des Servers $C, C_{x'}, D_1, P_x, P_D, P_E$ und P_F.

Die Werte C^*, C_{x^*}, D_2 und F sind aus den oben aufgelisteten Werten berechenbar. Durch diese Informationen kann jeder V durch die Stimmen-ID prüfen, ob seine Stimme auf dem Bulletin Board eingegangen ist und somit die korrekte Code Page, d. h. die Spalten A und B, für die weiteren Berechnungen verwendet wurde. Des Weiteren kann V für seine Stimme prüfen, ob das veröffentlichte C^* gleich ist, wie das, welches er vom Hardware-Token angezeigt bekommen hat, und somit seine Eingabe vom Server korrekt angenommen wurde.

Davon ausgehend, dass alle Wähler ihre Stimme geprüft haben und folglich alle Informationen korrekt sind, kann jeder die folgenden Schritte zur Verifikation durchführen [148]:

1. Überprüfung der Chiffrate in der Audit-Spalte B, indem für alle Zufallszahlen $r_1, r_2 \dots r_n$ und alle Kandidaten $c_1, c_2, \dots, c_n$ überprüft wird, ob $exp - enc(c_i, y, r_i) = B_i$ ist.
2. Überprüfung der Stimmen-ID duch die Berechnung des Hashs aus Spalten A und B der Code Page ($hash(A, B)$), wobei die Spalten jeweils lexikografisch geordnet sind.
3. Überprüfung, ob die letzten $\mathcal{K}$ Stellen von C mit C^* übereinstimmen.
4. Nachvollziehen des Proof-of-Selection. Dazu muss $C_{x^*} = h^{a_{0,b}||\cdots||a_{n,b} \cdot 10^{\mathcal{K}}}$ berechnet werden, wobei $a_{i,b}$ die letzten $\mathcal{K}$ Stellen von B_i sind. Anschließend kann $D_2 = C_{x^*} C_{C_{x'}}$ berechnet werden und die Beweise $P_{x'}$ und P_D nachvollzogen werden.
5. Nachvollziehen des Proof-of-Re-Encryption, indem P_E überprüft wird.
6. Nachvollziehen des Proof-of-Re-Encryption of the Ciphertext, indem $F_1 = D_1 E_1$ und $F_2 = D_2 E_2$ berechnet werden und mit diesen Informationen P_F nachvollzogen wird.
7. Zum Schluss muss die Auszählung der Decryption Tellers überprüft werden, dies ist abhängig von der gewählten Methode und wird hier nicht berücksichtigt.

5.3.3 Bewertung

Im Folgenden wird Du-Vote auf Grundlage der in Kapitel 3 beschriebenen Methodik unter Annahme eines Angreifermodells für Wahlen erster Ordnung mit der Möglichkeit der Manipulation von Hilfsmitteln bewertet.

Verifizierbarkeit

In diesem Abschnitt werden die Voraussetzungen der Verifizierbarkeit des Protokoll erörtert. Die erste Voraussetzung ist, dass das Hardware-Token vertrauenswürdig ist [148]. Diese Annahme kann durch Audits der Hardware und Software sichergestellt werden. Sie ist nicht unrealistisch, da das Hardware-Token eine in sich geschlossene Plattform ist, welche mit einfacher Soft- und Hardware realisiert werden kann.

Im Folgenden wird sich das Kapitel allerdings ausschließlich dem Worst-Case-Szenario widmen und die Verifizierbarkeit unter der Voraussetzung untersuchen, dass der Wahlcomputer P, der Server S und das Hardware-Token H in kooperierender Weise korrupt sind. Dazu wird die Wahrscheinlichkeit ermittelt, mit der Angriffe erkannt werden und somit gezeigt, dass ein Angriff sehr wahrscheinlich entdeckt wird.

SICHERHEITSANAHMEN Für die Bewertung der Verifikation werden die folgenden Sicherheitsannahmen als gegeben angesehen:

1. H hat keinen verdeckten Kanal und kann nur über die Tastatur mit dem Wahlsystem kommunizieren. Dadurch ist der Angreifer nicht in der Lage, das Verhalten von H während des Angriffs zu kontrollieren und müsste es deshalb bereits beim Herstellungsprozess festlegen (indem er z. B. die Firmware manipuliert). In diesem Fall wäre er nach dem Empfang des Geräts durch V nicht mehr in der Lage, das Verhalten von H zu verändern. Die Idee dahinter ist, dass H keine Möglichkeit besitzt, die Wahl-Nonce I in irgendeiner Weise zu lernen.

   Diese Annahme erscheint sehr praktikabel, da es sehr auffällig ist, wenn H einen verdeckten Kanal besitzt. Da das Token in der Regel mit einfacher Soft- und Hardware auskommt, kann diese Annahme durch einfache Audits sichergestellt werden. Wenn H keinen verdeckten Kanal besitzt, bleibt nur die Möglichkeit, I über die Tastatur zu einzugeben. Dies ist unrealistisch, da zum einen die Eingabekapazität der Tastatur limitiert ist und zum anderen V die Kontrolle darüber besitzt, was in das Token eingegeben wird. Außerdem darf die Nonce I erst veröffentlicht

werden, nachdem jeder Wähler sein Token erhalten hat. Da I aus öffentlich verifizierbaren Daten abgeleitet wird, welche erst zu einem festgelegten Zeitpunkt existiert haben, kann V überprüfen, ob das Token zu diesem Zeitpunkt bereits in seinem Besitz war.

2. Eine weitere Annahme ist, dass der Angreifer nicht alle P unter seiner Kontrolle hat. Diese Annahme hört sich realistisch an, da davon auszugehen ist, dass einige Wähler ein unübliches Betriebssystem oder eine besonders sichere Konfiguration ihres Wahlcomputers verwenden. Das große Problem ist allerdings die Wahlsoftware (egal ob im Browser oder in Form einer speziellen Client-Software), welche auf P ausgeführt wird. Falls es einem Angreifer gelingt, das Verhalten der Software zu manipulieren, kann davon ausgegangen werden, dass es ihm auch gelingen wird, alle Ps zu infizieren. Dies kann verhindert werden, indem der Quellcode veröffentlicht und es Wählern freigestellt wird, die Software selbst zu kompilieren und zu überprüfen.

3. Weiterhin wird angenommen, dass der Angreifer nicht die Fähigkeit besitzt, die Verteilung der Hs so zu beeinflussen, dass nur nicht-vertrauenswürdige Ps mit unvertrauenswürdigen Hs zusammenarbeiten. Diese Annahme kann realistisch eingehalten werden, da der Verteilungsweg der Hs komplett unabhängig vom Verteilungsweg der Malware ist. Eine Möglichkeit, die zufällige Verteilung der Tokens sicherzustellen, ist es, V bei der Abholung einen Token aussuchen zu lassen.

   Es stellt außerdem ein Problem für den Angreifer dar, dass er nicht vorhersagen kann, von welchem P der Wähler letztendlich wählen wird. Aufgrund der ausgeklügelten Privatsphären-Eigenschaft des Wahlverfahrens, ist es P (zumindest ohne Hilfe von S oder H) nicht möglich, das Wahlgeheimnis zu brechen. Dies führt dazu, dass Wähler die Wahl ohne Bedenken auf fremden Ps durchführen können.

4. Eine schwer sicherzustellende Anforderung ist es, dass der Wähler die Verifikation seiner Stimmen-ID und seines Codes C^* von einer vertrauenswürdigen Plattform durchführen soll. Denn macht er das nicht, kann diese Plattform die Ausgabe der auf dem Bulletin Board vorhandenen Informationen so manipulieren, dass der Wähler eine Manipulation nicht erkennt.

   Einerseits wird es immer einige Plattformen geben, welche nicht vom Angreifer manipulierbar sind. In einem solchen Fall wäre es denkbar, dass der Wähler die Verifikation von so vielen Geräten wie möglich durchführt und davon ausgegangen wird, dass eines der Geräte nicht vom Angreifer infiziert ist, und so eine Manipulation festgestellt wird. Zum anderen ermöglicht es Du-Vote, die Verifikation an eine vertrauenswürdigen Instanz zu delegieren, ohne dabei das Wahlgeheimnis zu brechen. Ein Problem das hierbei auftritt ist, dass die Übermittlung zu dieser vertrauenswürdigen Instanz einen Kanal benötigt, welcher Integrität gewährleistet. In diesem Szenario delegiert der Wähler die Verifikation an einen Auditor (z. B. die Organisation für Sicherheit und Zusammenarbeit in Europa - OSZE) und dieser Auditor stellt sicher, dass der Verifikationscomputer nicht manipuliert ist.

Mit diesen Sicherheitsannahmen lassen sich zwei Angriffsszenarien aus der Sicht von H konstruieren. Zum einen kann H einen zufälligen Wert zurückgeben. Ein solcher Angriff wird offensichtlich mit hoher Wahrscheinlichkeit erkannt. Zum anderen kann H einen sogenannten Substitutionsangriff durchführen. Dabei werden während der Produktion Daten auf H gespeichert, welche es einem nicht vertrauenswürdigen P ermöglichen, den Code so zu wählen, dass H einen vorab vereinbarten Wert zurück gibt.

Im Detail speichert der Angreifer eine Menge von Tupeln $X = \{(z_1, Q_1), \ldots, (z_t, Q_t)\}$ auf H. Hierbei ist z_i der Code, welcher verschlüsselt werden soll, wenn die Eingabe Q_i auftritt. Diese Tupel stehen auch P zur Verfügung. Bei der Generierung der Codes überprüft P, ob für beide Spalten ein Tupel (z_i, Q_i) existiert, sodass mit den Codes $\{a_0, a_1, \ldots, a_{n-1}\}$ und $\{b_0, b_1, \ldots, b_{n-1}\}$ ein $Q_{1,i}$ und ein $Q_{2,i}$ generiert werden kann, indem der Offset α und β gewählt wird. Außerdem wird vorausgesetzt, dass in der Wahl-Spalte das entsprechende $z_{1,i}$ zu $Q_{1,i}$ und das $z_{2,i}$ zu $Q_{2,i}$ existiert. Angenommen P triggert einen Angriff und V gibt $a_0 \| a_1 \| \cdots \| a_{n-1} \| x_1^*$ in das Hardware-Token ein. Dann stellt H fest, dass

$a_0 \| a_1 \| \cdots \| a_{n-1} = Q_i$ und ersetzt x_1^* durch z_i. Danach führt H die entsprechende Exponentiation durch und gibt c an V aus. Gibt V nun c in P ein, hat er entsprechend für einen anderen Kandidaten gewählt. Durch die Einschränkung ist es dem Angreifer zwar nicht möglich, V für einen bestimmten Kandidaten stimmen zu lassen, er kann V jedoch die Stimme wegnehmen, indem er ihn zufällig wählen lässt. Da V bei der Überprüfung nur das aus dem BB verfügbare c mit dem ausgegebenen c vergleichen kann, geht ein angegriffener Wähler davon aus, dass seine Wahl korrekt in das Ergebnis eingeflossen ist.

WAHRSCHEINLICHKEIT DER ERKENNUNG EINES ZUFALLSANGRIFFS Angenommen der Wähler gibt $a_{0,a} \| a_{1,a} \| \cdots \| a_{n-1,a} \| x^*$ in H ein und H ersetzt nun x^* durch die Zufallszahl x_1^*. Anschließend berechnet das Hardware-Token $c = H(a_{0,a} \| a_{1,a} \| \cdots \| a_{n-1,a} \| x_1^*)$. Damit der Angriff unentdeckt bleibt, muss gelten $x_1^* \in \{a_{0,b}, a_{1,b}, \cdots, a_{n-1,b}\} - \{x^*\}$, da S einen ZKP veröffentlichen muss, welcher zeigt, dass es zu x_1^* ein passendes Chiffrat x in der Wahl-Spalte gibt. H kennt durch die Eingabe bereits $(n+1)$ Codes und muss einen der $n-1$ übrigen Codes der Wahl-Spalte erraten, damit sein Angriff nicht festgestellt werden kann. Die Erfolgswahrscheinlichkeit ist also

$$\Pr[\text{Erfolgswahrscheinlichkeit}] = \frac{n-1}{10^{\mathcal{K}} - (n+1)}$$

Folglich ist die Warscheinlichkeit, dass der Angriff erkannt wird

$$\Pr[\text{Erkennungswahrscheinlichkeit}] = 1 - \frac{n-1}{10^{\mathcal{K}} - (n+1)}$$

Folglich lässt sich durch die richtige Wahl von $\mathcal{K}$ die Erkennungswahrscheinlichkeit beliebig erhöhen. Bei $\mathcal{K} = 4$ und $n = 10$ liegt die Erkennungswahrscheinlichkeit bei $99,91\%$.

WARSCHEINLICHKEIT DER ERKENNUNG EINES SUBSTITUTIONSANGRIFFS In diesem Kapitel soll die Wahrscheinlichkeit, dass der Angreifer erfolgreich ist, mit der Wahrscheinlichkeit, dass der Angriff vom System erkannt wird, verglichen werden. Dazu sei $l = |X|$, p = Wahrscheinlichkeit, mit der ein Q_i in zyklisch permutierter Reihenfolge auftritt, n = Anzahl der Kandidaten und h = Anzahl der vertrauenswürdigen P.

Als erstes wird die Wahrscheinlichkeit berechnet, dass der Substitutionsangriff entdeckt wird. Dies kann prinzipiell nur dann geschehen, wenn ein vertrauenswürdiges P mit einem nicht-vertrauenswürdigen H zusammenarbeitet. Eine weitere Voraussetzung ist, dass P zufällig eine Aktion triggert, indem die Eingabe der Audit-Spalte zufällig Q_i ergibt und z_i nicht in der Wahl-Spalte vorhanden ist. Die Wahrscheinlichkeiten ergeben sich wie folgt:

$$\Pr[\text{Eingabe} = Q_i] = p \wedge \Pr[\text{Offset wird richtig gewählt}] = \frac{1}{n} \cdot p = \frac{p}{n}$$

$$\Pr[z_i \in \text{Wahl-Spalte}] = \frac{n}{10^{\mathcal{K}}}$$

$$\Pr[z_i \notin \text{Wahl-Spalte}] = 1 - \Pr[z_i \in \text{Wahl-Spalte}] = 1 - \frac{n}{10^{\mathcal{K}}}$$

Da l viele Q_is existieren, ergibt sich eine Gesamtwahrscheinlichkeit von:

$$\begin{aligned}\Pr[\text{Angriff wird erkannt}] &= \sum_{i=1}^{l} \Pr[\text{Eingabe} = Q_i] \wedge \Pr[z_i \notin \text{Wahl-Spalte}] \\ &= \sum_{i=1}^{l} \left(\frac{p}{n}\right)\left(1 - \frac{n}{10^{\mathcal{K}}}\right) \\ &= \frac{lp}{n}\left(1 - \frac{n}{10^{\mathcal{K}}}\right)\end{aligned}$$

Als nächstes wird die Wahrscheinlichkeit berechnet, dass der Angriff erfolgreich durchgeführt werden kann. Diese setzt sich aus der Wahrscheinlichkeit, dass der Angriff durchführbar ist und der Wahrscheinlichkeit, dass der Angriff nicht erkannt wird, zusammen.

$$\begin{aligned}\Pr[\text{Angriff durchführbar}] &= \sum_{i=1}^{l} \Pr[\text{Codes} \in Q_i] \wedge \Pr[z_i \in \text{Wahl-Spalte}] \\ &= \sum_{i=1}^{l} (p)\left(\frac{n}{10^{\mathcal{K}}}\right) \\ &= \left(\frac{lpn}{10^{\mathcal{K}}}\right)\end{aligned}$$

$$\begin{aligned}\Pr[\text{Angriff nicht wird erkannt}] &= \sum_{i=1}^{l} \Pr[\text{Eingabe} = Q_i] \wedge \Pr[z_i \in \text{Wahl-Spalte}] \\ &= \sum_{i=1}^{l} \left(\frac{p}{n}\right)\left(\frac{n}{10^{\mathcal{K}}}\right) \\ &= \frac{lp}{n}\left(\frac{n}{10^{\mathcal{K}}}\right) \\ &= \frac{lp}{10^{\mathcal{K}}}\end{aligned}$$

Im Folgenden wird das Verhältnis zwischen der Wahrscheinlichkeit, dass ein Angriff erkannt wird, und der Wahrscheinlichkeit, dass ein Angriff erfolgreich ausgeführt wird, ermittelt. Dies wird durch folgende Formel beschrieben:

$$\begin{aligned}R &= \frac{h \Pr[\text{Angriff wird erkannt}]}{(1-h)\Pr[\text{Angriff durchführbar}] + h \Pr[\text{Angriff nicht wird erkannt}]} \\ &= \frac{h\frac{lp}{n}\left(1 - \frac{n}{10^{\mathcal{K}}}\right)}{(1-h)\frac{lpn}{10^{\mathcal{K}}} + h\frac{lp}{10^{\mathcal{K}}}} = \frac{\frac{lp}{n}h\left(1 - \frac{n}{10^{\mathcal{K}}}\right)}{(1-h)n\frac{lp}{10^{\mathcal{K}}} + h\frac{lp}{10^{\mathcal{K}}}} \\ &= \frac{\frac{lp}{n}h\left(1 - \frac{n}{10^{\mathcal{K}}}\right)}{\frac{lp}{10^{\mathcal{K}}}\left((1-h)n + h\right)} = \frac{lph\left(1 - \frac{n}{10^{\mathcal{K}}}\right)10^{\mathcal{K}}}{lp\left((1-h)n + h\right)n} \\ &= \frac{h\left(1 - \frac{n}{10^{\mathcal{K}}}\right)10^{\mathcal{K}}}{((1-h)n + h)n}\end{aligned}$$

Diese Formel verdeutlicht, dass die Chance einer Erkennung gegenüber der Chance eines erfolgreichen Angriffs durch den Sicherheitsparameter $\mathcal{K}$, unter der Prämisse, dass die oben genannten Sicherheits-

annahmen gelten, beliebig erhöht werden kann. Geht man davon aus, dass es dem Angreifer gelingt, 50% der Wahlcomputer zu infizieren und vier Wahlmöglichkeiten existieren, dann ist für $\mathcal{K} = 4$ die Wahrscheinlichkeit, dass der Angriff erkannt wird, 500 mal höher, als die Wahrscheinlichkeit, dass der Angriff nicht erkannt wird [148].

Im Folgenden wird analysiert, welche Eigenschaften der Verifizierbarkeit das Protokoll nach Unterabschnitt 3.2.1 erfüllt. Offensichtlich ist die innere individuelle Verifizierbarkeit vor der Auszählung durch das in 5.3.2 beschriebene Vorgehen erfüllt. Des Weiteren kann die Auszählungsphase je nach angewandtem Auszählungsverfahren universell verifiziert werden. Somit ergibt sich die Eigenschaft der inneren individuellen Verifizierbarkeit nach der Wahl. Für die universelle Verifizierbarkeit müssen drei Eigenschaften betrachtet werden. Die kontinuierliche Korrektheits-Verifizierbarkeit sowie die bedingungslose Einmaligkeits-Verifizierbarkeit sind durch den obigen Beweis gegeben. Die bedingungslose Wahlberechtigungs-Verifizierbarkeit kann einfach eingeführt werden, indem eine Liste mit allen gültigen Wähler-IDs veröffentlicht wird. Es wäre auch eine Verknüpfung der Wähler-IDs mit den Namen denkbar, somit ist allerdings sehr leicht einzusehen, wer tatsächlich gewählt hat und wer nicht.

Wahlgeheimnis

Zur Erinnerung: k wird in H eingesetzt, um den Code c zu erzeugen, aus welchem S den verschlüsselten Kandidaten generiert. P erzeugt das α und das β, welches die Zuordnung zwischen Code und Kandidaten bestimmt. Das bedeutet, sobald der Angreifer Kontrolle über zwei der drei Instanzen besitzt, kann er das Wahlgeheimnis brechen. Hierbei werden die folgenden drei Fälle unterschieden:

1. **Nicht vertrauenswürdiger P:** Das Wahlgeheimniss bleibt gewahrt, wenn P nicht die Möglichkeit besitzt, die Informationen x^*, x oder k in Erfahrung zu bringen. Es wird angenommen, dass H und S vertrauenswürdig sind. Folglich kann P die Informationen nur über das Bulletin Board in Erfahrung bringen. Kritisch sind hierbei die Informationen $E = (E_1, E_2) = (g^e x_1, y^e x_2)$ und $D = (D_1, D_2) = (g^d, y^d h^{x_2})$ sowie die ZKP-Beweise P_x, P_D, P_E und P_F. Offensichtlich erfüllen die Beweise alle die Zero-Knowledge-Eigenschaft, was bedeutet, dass sie keine Information, außer der Tatsache, ob der Beweis korrekt ist oder nicht, preisgeben. E ist von g^e abhängig, wobei e von S zufällig gewählt wird und D ist von g^d abhängig, wobei $d = k + r'$ und r' zufällig von S gewählt werden. Da P vom Bulletin Board weder d noch e in Erfahrungen bringen kann, hat P keine Chance, die Wahl zu lernen.

2. **Nicht vertrauenswürdiger S:** Im Vergleich zur oben stehenden Analyse besitzt S die Informationen x^*, x und k. Allerdings kann S diese Informationen keinem Kandidaten zuordnen, falls P das Mapping zwischen den Kandidaten und den Chiffraten nicht preisgibt. Unter der Annahme, dass P vertrauenswürdig ist, wählt P die Offsets α und β zufällig und hält den Offset der Wahl-Spalte[8] (α oder β) sowie den Klartext von x geheim.

   Ein weiteres Problem ist, dass die Informationen mit dem öffentlichen Schlüssel y verschlüsselt sind. Falls S in Besitz von $z = \prod_{j=1}^{m} g^{z_j}$ kommt, wobei z_j der private Schlüssel des Decryption Teller T_j ist, kann S jede Stimme entschlüsseln und so den Klartext von x herausfinden. Unter der Annahme, dass es mindestens einem der m Decryption Teller gelingt, sein z_j geheim zu halten, kann S allerdings diesen privaten Schlüssel nicht berechnen.

3. **Nicht vertrauenswürdiges H:** Es wird angenommen, dass H keinen verdeckten Kanal besitzt und somit ausschließlich V entscheidet, welche Informationen H erhält. Folglich kann H das Wahlgeheimnis nur brechen, wenn H mit S oder P zusammenarbeitet. Wenn H mit S zusammenarbeitet, kann H keinen Beitrag zum Brechen des Wahlgeheimnisses leisten, da S bereits alle Informationen besitzt, die auch H besitzt. Das bedeutet, dass dieser Fall gleichbedeutend mit dem Fall ist, dass S nicht vertrauenswürdig ist.

   Sei P nicht vertrauenswürdig und H und P werden vom gleichen Angreifer kontrolliert, dann kann H die Information x^* oder k mit P teilen. Da H allerdings keinen verdeckten Kanal besitzt,

8 Abhängig vom Münzwurf von V.

muss eine Strategie erarbeitet werden, wie dies geschehen kann. Eine Möglichkeit ist es, dass H, anstatt die vorgegebene Funktion zu berechnen, die letzten $\mathcal{K}$ Stellen der Eingabe, was x^* entspricht, zurück gibt. Da P die Zuordnung von c_i zu x^* bekannt ist, lernt P nun die Stimme. Allerdings ist diese Vorgehensweise für V sehr auffällig.

4. **Nicht vertrauenswürdige Decryption Teller:** Offensichtlich kann das Wahlgeheimnis nur eingehalten werden, wenn mindestens einer der Decryption Teller $\{T_1, T_2, \ldots, T_m\}$ vertrauenswürdig ist. Arbeiten alle Decryption Teller zusammen, können diese den privaten Schlüssel z berechnen und die E's der Wähler entschlüsseln $exp - dec(z, E)$. Dabei erhalten die Decryption Teller offensichtlich die Klartext-Stimme der Wähler und können so das Wahlgeheimnis brechen.

Zusammengefasst hält das Wahlprotokoll die Eigenschaft des Wahlgeheimnisses ein, wenn mindestens einer der m Decryption Teller sowie P vertrauenswürdig sind oder wenn mindestens einer der m Decryption Teller, H und S vertrauenswürdig sind [148].

Quittungsfreiheit

Du-Vote ist quittungsfrei unter der Voraussetzung, dass der Angreifer zum Zeitpunkt der Wahl keinen Zugriff auf P hat. Der Wähler hat dann zwar die Möglichkeit, mit den Werten der angezeigten Code Page und dem Hardware-Token das gleiche c^* zu erstellen. Dazu gibt er die gleiche Eingabe, welche er während der Wahl eingegeben hat (z. B. $b_0||b_1||\cdots||b_{n-1}||x^*)$), in das Hardware-Token ein. Kommt dabei das gleiche $c^* = H(b_0||b_1||\cdots||b_{n-1}||x^*)$ heraus, welches auch auf dem BB unter der vom Wähler angegebenen Voter-ID veröffentlicht ist, kann der Angreifer nahezu sicher sein, dass der Wähler den Kandidaten mit dem Code x^* gewählt hat. Allerdings weiß der Angreifer nicht, welchen Kandidaten dieses x^* repräsentiert. Diese Zuordnung kennt der Angreifer nur dann, wenn er die Code Page zur Zeit der Wahl kennt. Dieses kann er nur kennen, wenn er P bereits zum Zeitpunkt der Wahl kompromittiert hat.

Eine weitere Möglichkeit wäre es, dass der Wähler die Code Page abfotografiert und dem Angreifer das Foto zusammen mit seiner Quittung präsentiert. Dies ist allerdings kein eindeutiger Beweis, da der Wähler das Bild (z. B. mit einem Bildbearbeitungsprogramm) manipulieren kann. Dies könnte unterstützt werden, indem dem Wähler die Option „shuffle" angeboten wird, welche die Reihenfolge der Codes der Code Page zufällig ändert. Dies darf allerdings nur für die Wahl-Spalte geschehen, da sich sonst auch die Reihenfolge der Audit-Spalte ändert, was zu einer Änderung des Eingabestrings im vorderen Teil führt. Da P bis zur Abgabe der Stimme geheim halten muss, welches die Audit-Spalte und welches die Wahl-Spalte ist, muss darauf geachtet werden, dass eine solche Permutation erst nach dem erfolgreichen Abschluss der Stimmabgabe möglich ist.

Angenommen der Angreifer kontrolliert S, womit er die Stimme des Wählers x^* kennt. Auch hier kann der Wähler dem Angreifer keine Informationen zur Verfügung stellen, sodass der Angreifer die Zuordnung zwischen Kandidat x^* bereitstellen kann. Somit halten auch hier die im letzten Abschnitt genannten Maßnahmen das Protokoll quittungsfrei. Das Selbe gilt, wenn der Angreifer H kontrolliert. Eine andere Möglichkeit, die Quittungsfreiheit auszuhebeln, ist es, das im vorherigen Abschnitt beschriebene Wahlgeheimnis zu brechen [148].

Nicht-Erpressbarkeit

Du-Vote wurde nicht für die Einhaltung der Nicht-Erpressbarkeit entworfen. Damit ist gemeint, dass Du-Vote nicht gegen Randomisierungsangriffe, Abwesenheitsangriffe und Simulationsangriffe resistent ist. Im Folgenden werden mögliche Angriffe beschrieben, welche selbst für Angreifer, die keine besonderen Fähigkeiten besitzen, auszuführen sind:

- Randomisierungsangriff: Dieser Angriff ist etwas komplizierter auszuführen. Der Angreifer kann den Wähler dazu zwingen, dass das veröffentlichte c^* eine gewisse Eigenschaft besitzen muss. Da bei jedem Wahlgang neue Codes zufällig erstellt werden, muss der Wähler solange die entsprechenden Werte in H eingeben, bis c^* die entsprechende Eigenschaft erfüllt. Ist die geforderte

Eigenschaft gut gewählt, so hat der Wähler nur eine geringe Chance, dass dies tatsächlich seiner eigenen Wahlentscheidung entspricht.

- Abwesenheitsangriff: Dieser Angriff ist sehr einfach durchzuführen und besitzt mehrere Ausprägungen. Die einfachste Möglichkeit ist, dass der Angreifer den Wähler dazu zwingt, das Hardware-Token dem Angreifer auszuhändigen.
- Simulationsangriff: Hierbei kann der Angreifer den Wähler dazu zwingen, das Token sowie den Benutzernamen und das Passwort herauszugeben. Anschließend kann der Angreifer seine Stimme anstelle des Wählers abgeben.

Es ist hier wichtig, organisatorische Maßnahmen zu treffen. Die Autoren von Du-Vote schlagen deshalb vor, den Token mit einer rechtlich bindenden Signatur zu verknüpfen, sodass z. B. Banking oder Einkaufen mit dem Token möglich sind. Durch diese Maßnahme wird die Hemmschwelle, H aus der Hand zu geben, erhöht. Dadurch werden zumindest Abwesenheits- und Simulationsangriffe erschwert [148].

Robustheit

Die Robustheit des Protokolls ist nicht erfüllt, da einer der m Decryption Teller ausreicht, um eine Entschlüsselung des Wahlergebnisses zu verhindern. Eine bessere Lösung wäre es hier, Threshold-Encryption zu verwenden, welche k aus m private Schlüssel benötigt, um die Wahl zu entschlüsseln. Verwendet man ein solches Verfahren, hat dies natürlich sofort einen Einfluss auf die Annahmen, welche bei der Analyse des Wahlgeheimnisses (siehe Punkt 5.3.3) gewählt wurden. Verwendet man k aus m Threshold-Encryption, so muss für die Einhaltung des Wahlgeheimnisses $n - (k + 1)$ vielen Decryption Tellers vertraut werden, weswegen es wichtig ist, k und m richtig zu wählen.

Benutzbarkeit

Die praktische Benutzbarkeit von Du-Vote wurde bei noch keiner Wahl getestet. Es erscheint allerdings nicht benutzerfreundlich, dass man zwingend ein Hardware-Token H benötigt und der Wähler zusätzlich alle Eingaben in H von Hand vornehmen muss (Eigenschaft BW.2.3).

Ein Problem ist, dass die Eingabe abhängig von der Anzahl der Kandidaten n sowie vom Sicherheitsparameter $\mathcal{K}$ ist, wobei $\mathcal{K}$ proportional zu n steigen muss, um das gleiche Sicherheitsniveau zu erhalten. Bei vielen Kandidaten könnte man sich überlegen, ob man nicht alle Kandidaten der Auditspalte in H eingibt, allerdings wird dadurch wiederum die Verifizierbarkeit erschwert, weshalb hier eine Abwägung erforderlich ist.

6 BRIEFWAHL IN DEUTSCHLAND

Aufgrund der zum Zeitpunkt der Arbeit stattfindenden Bundestagswahl und verschiedenen Aufrufen, den ohnehin bereits großen Anteil[1] an Wählerinnen, die per Briefwahl abstimmen, weiter zu erhöhen [165, 166], geriet auch die Sicherheit der in Deutschland durch die Bundeswahlordnung und das Bundeswahlgesetz regulierten Briefwahl in den Fokus der Autoren. Während des Begutachtungsprozesses, der unter anderem Korrespondenz mit dem Statistischen Bundesamt bzw. dem Bundeswahlleiter (siehe Abschnitt A.3) sowie Wahlbeobachtungen vor Ort beinhaltete, wurden einige Sicherheitsmängel festgestellt, die im Folgenden kurz erläutert werden. Diese Sicherheitsmängel in Betracht ziehend, stellten sich die Autoren die Frage, in wie weit die Briefwahl in Deutschland die Anforderungen erfüllt, die in dieser Arbeit zur Bewertung der Internetwahlsysteme herangezogen werden. Aus diesem Grund wird im Folgenden kurz der Briefwahlprozess beschrieben, von den Autoren festgestellte Sicherheitsmängel aufgezeigt und eine Bewertung auf Basis der in Kapitel 3 vorgestellten Methodik vorgenommen.

6.1 PROZESS

Wie die meisten der bisher besprochenen Internetwahlverfahren gliedert sich auch die Briefwahl in insgesamt drei Wahlphasen, die im Folgenden kurz erläutert werden, um dem Leser eine Grundlage für das Verständnis der Sicherheitsmängel zu geben, die daraufhin besprochen werden.

6.1.1 Pre-Wahlphase

Wahlberechtigte können auf verschiedenem Wege Briefwahlunterlagen beantragen. Dazu zählt die persönliche und die schriftliche Beantragung in Form eines Briefes bzw. der entsprechend ausgefüllten amtlichen Wahlbenachrichtigung, per E-Mail oder Fax sowie, abhängig von der Wahlgemeinde, per Online-Antrag. Der Antrag muss in jedem Fall den Vor- und Familiennamen, das Geburtsdatum sowie die Wohn- und evtl. abweichende Versandanschrift enthalten. Die Beantragung für eine andere Person ist lediglich unter Vorlage einer schriftlichen Vollmacht (mit Originalunterschrift) und somit persönlich oder schriftlich, nicht aber auf elektronischem Wege möglich [167].

Frühestens nach der Eintragung aller gemeldeten Personen ins Wählerverzeichnis, also gemäß § 16 Absatz 1 der Bundeswahlordnung 42 Tage bzw. sechs Wochen vor der Wahl, werden diese Briefwahlunterlagen dann an die Wahlberechtigten versandt. Sie enthalten einen Wahlschein (zur Bestätigung der Identität der wahlberechtigten Person), einen Stimmzettel, ein Merkblatt über die korrekte Nutzung, einen Umschlag für den Stimmzettel sowie einen größeren Umschlag für den Stimmzettelumschlag und den Wahlschein. Bei Personen, die Briefwahl beantragt haben, wird in der entsprechenden Zeile der Wählerlisten ein „W“ eingetragen [168]. Sie dürfen am Wahltag zwar noch wählen, jedoch ausschließlich unter Vorlage ihres gültigen Wahlscheins, um auszuschließen, dass sie bereits vorher per Briefwahl abgestimmt haben.

1 Bei der Bundestagswahl 2013 stimmten laut dem Bundeswahlleiter bereits 24,3 Prozent der Wahlberechtigten per Briefwahl ab [164].

6.1.2 Wahlphase

Um an der Wahl teilzunehmen, muss der Stimmzettel ausgefüllt, in den blauen Umschlag gesteckt und dieser verschlossen werden. Der verschlossene blaue Umschlag wird dann zusammen mit dem ausgefüllten Wahlschein, der eine Versicherung Eides statt enthält, in den roten Umschlag gesteckt und dieser verschlossen. Der rote Umschlag mit den Wahlunterlagen kann dann entweder bei der Gemeindebehörde abgegeben oder innerhalb Deutschlands kostenlos auf dem Postweg an eben diese versandt werden. Im Ausland lebende deutsche Staatsbürger können auf Antrag ins Wählerverzeichnis eingetragen werden und dann an der Briefwahl teilnehmen. Sie können die ausgefüllten Unterlagen entweder bei der nächsten Vertretung der Bundesrepublik Deutschland (Botschaft oder Konsulat) abgeben oder aber ebenfalls auf dem Postweg versenden, wobei das Porto hier selbst getragen werden muss.

6.1.3 Post-Wahlphase

Nachdem der Wahlbrief bei der Gemeindebehörde eingegangen ist, wird dieser dort unter Verschluss gehalten und am Wahltag an die entsprechenden Briefwahlvorstände verteilt. Die Briefwahlvorstände bestehen aus fünf bis neun Wahlberechtigten und sollen sich gegenseitig kontrollieren. Um 15:00 Uhr, also drei Stunden vor Ende des offiziellen Wahlzeitraums, beginnen die Briefwahlvorstände damit, die roten Wahlumschläge zu öffnen und die Wahlscheine sowie die blauen Umschläge auf Gültigkeit zu überprüfen. Ist der Wahlschein laut Vorschrift unterschrieben und der blaue Umschlag mit dem Stimmzettel verschlossen und unversehrt, werden die beiden getrennt, indem der blaue Umschlag ungeöffnet in die Urne geworfen wird. Sobald alle blauen Umschläge in der Urne sind und der Wahlzeitraum abgelaufen ist, wird die Urne geöffnet und die Stimmen in den blauen Umschlägen werden ausgezählt. Während der Öffnung der Umschläge und der Befüllung der Urne müssen mindestens drei, bei der darauffolgenden Stimmauszählung mindestens fünf Mitglieder des Wahlvorstandes anwesend sein. Wie auch bei regulären Präsenzwahlen ist der gesamte Prozess öffentlich und kann von Anfang bis Ende beobachtet werden.

6.2 ANALYSE

Im Folgenden werden Schwachstellen und möglichen Angriffe auf den Briefwahlprozess beschrieben, die sich aus Bedenken während der tatsächlichen Durchführung der Briefwahl ergaben.

6.2.1 Beantragungsprozess

Die Briefwahlunterlagen inklusive des Wahlscheins können auf unterschiedlichem Wege beantragt werden, unter anderem auch per E-Mail. Eine Überprüfung der Identität der beantragenden Person findet hierbei nicht statt. Weiterhin ist es möglich, eine von der Meldeanschrift abweichende Versandadresse anzugeben. § 28 Absatz 4 der Bundeswahlordnung sieht eigentlich vor, dass bei abweichender Versandanschrift eine Mitteilung an die Wohnanschrift gesendet wird, um eine nicht autorisierte Beantragung für die betroffene Person erkennbar zu machen und ggf. Gegenmaßnahmen einzuleiten. Die praktische Nachprüfung während der Bundestagswahl ergab jedoch, dass diese Sicherheitsmaßnahme teilweise falsch implementiert ist. Die Mitteilung wurde zwar versandt, allerdings ebenfalls an die von der Wohnanschrift abweichende Versandadresse, sodass die Kontrollfunktion durch die wahlberechtigte Person nicht mehr gegeben war. Die oben skizzierten Umstände erlauben es einem Angreifer, Wahlunterlagen im Namen einer anderen wahlberechtigten Person zu beantragen. Um anonym zu bleiben, kann er dazu unter falschem Namen z. B. einen zusätzlichen Briefkasten an einem großen Wohnblock oder auf einer Baustelle aufhängen. Ein weiterer Weg ist die Kurzzeitanmietung einer kleinen Wohnung, ebenfalls unter falschem Namen. Der Angreifer kann durch die unautorisierte Be-

antragung der Wahlunterlagen zum Einen dafür sorgen, dass wahlberechtigte Personen, die eigentlich planen, am Wahltag zu wählen, einen Wahlscheinvermerk erhalten und ihnen gleichzeitig den Wahlschein vorenthalten. Der eigentlich wahlberechtigten Person wird am Wahltag im Wahllokal nach § 56 Absatz 6 Nummer 2 der Bundeswahlordnung sodann der Wahlgang verweigert, da diese den notwendigen Wahlschein nicht vorzeigen kann. Es kann somit also erzwungen werden, dass eine eigentlich wahlberechtigte Person nicht an der Wahl teilnimmt. In Anlehnung an die von Sweeney et al. [169] vorgestellte Methodik, Adressen und andere Daten über Adress-Händler oder von Darknet-Plattformen zu beziehen, könnte dieser Denial-of-Service-Angriff durchaus skaliert werden. Außerdem kann ein Angreifer, indem er die erschlichenen Briefwahlunterlagen selbst benutzt, unter fremdem Namen eine Stimme abgeben. Zwar ist es hierfür notwendig, dass der Angreifer eine falsche Erklärung Eides statt abgibt, allerdings stellt dies aufgrund der für diesen Angriff ohnehin nötigen hohen kriminellen Energie und des geringen Erkennungsrisikos eine zu vernachlässigende Hürde dar.

6.2.2 Rücksendeprozess / Verifizierung

Sobald die Wahlunterlagen der wahlberechtigten Person erfolgreich zugestellt wurden, kann diese von ihrem Stimmrecht Gebrauch machen und den verschlossenen Wahlbrief postalisch an die jeweilige Gemeindebehörde zurücksenden. Ist der Wahlbrief einmal im Postkasten, verliert die Wählerin jedoch jegliche aktive, aber auch passive Kontrolle über den Wahlbrief. Weder Bundeswahlgesetz, noch Bundeswahlordnung sehen einen Prozess vor, der der Wählerin ermöglichen würde, sich über den Verbleib ihres Wahlbriefs zu erkundigen. Auch beim Feldversuch in Form eines exemplarischen Anrufs beim entsprechenden Wahlamt konnte keine Auskunft über den Verbleib des Wahlbriefes eingeholt werden. Es lässt sich deshalb nicht nachverfolgen, ob der Wahlbrief rechtzeitig bei der Gemeindebehörde eingeht, wenn er erst zeitnah zum Wahltag abgesendet wurde. Schreiber [170] sieht die Verantwortung dafür, dass der Wahlbrief rechtzeitig abgesendet wird, jedoch bei der jeweiligen Wählerin, was aufgrund der Tatsache, dass die Briefwahlunterlagen bereits sechs Wochen vorher versendet werden, durchaus nachvollziehbar ist. Allerdings betrifft ein schwerwiegenderes Problem auch Wählerinnen, die ihren Wahlbrief bereits lange Zeit vor dem eigentlich Wahltag versandt haben. Briefe können entweder auf dem Postweg verloren gehen oder absichtlich entwendet werden [171–173]. Beide Fälle können durch die fehlende gesetzliche Möglichkeit der Verifizierung durch die Wählerinnen nicht erkannt werden. Auf Rückfrage beim Bundeswahlleiter wurde den Autoren mitgeteilt, dass Wählerinnen jedoch folgende freiwillige Möglichkeiten hätten, um über den Verbleib des Wahlbriefes im Bilde zu bleiben (siehe Abschnitt A.3):

- Versand des Wahlbriefes per besonderer Versendungsform, z. B. als Einschreiben mit Rückschein,
- persönlicher Einwurf des Wahlbriefes bei der Wahlbehörde und
- Briefwahl an Ort und Stelle.

Beim Versand per Einschreiben oder einer ähnlichen, verfolgbaren Versendungsform fallen nicht zu vernachlässigende Kosten an, die Wählerinnen selbst tragen müssen. Die beiden letztgenannten Möglichkeiten verringern das Risiko zwar erheblich, jedoch ist auch hier fraglich, inwiefern diese von den Wählerinnen aufgrund der damit verbundenen Komforteinbußen tatsächlich angenommen werden. Weiterhin handelt es sich bei der Briefwahl an Ort und Stelle genau genommen nicht mehr um eine Wahl in unkontrollierten Umgebungen, da die Wahlbedingungen größtenteils der eigentlichen Wahl vor Ort entsprechen.

Zwar können die drei genannten Optionen dazu dienen, das Risiko für Wählerinnen individuell zu senken. Aufgrund der damit verbundenen Kosten bzw. Komforteinbusen ist es jedoch fraglich, ob sie die allgemeine Sicherheit der gesamten Briefwahl signifikant erhöhen. Außerdem sollte es auch Wählerinnen, denen erst nach Versand ihres Wahlbriefes Zweifel kommen, möglich sein, dessen Verbleib festzustellen. Gemäß des Grundsatzes *Security by Design* sollte der gesetzliche Briefwahlprozess als solcher sicher und verifizierbar sein, anstatt die Verantwortung durch fakultative Maßnahmen auf die einzelne Wählerin zu transferieren.

Unabhängig davon, wie hoch die Wahrscheinlichkeit ist, dass der Wahlbrief erfolgreich bei der Wahlbehörde zugestellt wird, stellt sich die Frage, wie die Wählerin feststellen kann, dass ihre Stimme bei der Auszählung am Wahltag auch tatsächlich in der Urne gelandet ist. Auch hierzu schweigen sich Bundeswahlordnung und das Bundeswahlgesetz aus. Eine mögliche Verbesserung dbzgl. wird in Abschnitt 6.3 diskutiert.

6.2.3 Fälschungssicherheit

Mehrere Faktoren begünstigen den Umstand, dass Wahlfälschungen für Wahlbeobachter und Wahloffizielle schwer zu erkennen und deren Urheber noch schwerer nachzuverfolgen sind [174]. So werden Wahlscheine bei der Öffnung der Wahlbriefe z. B. nicht mit dem kompletten Wahlscheinverzeichnis, sondern lediglich mit einer Liste ungültig erklärter Wahlscheine abgeglichen („Blacklisting“). Dies öffnet einen Vektor für Angreifer, die Wahlunterlagen täuschend echt fälschen und mit Fantasienamen- und nummern versehen. Da diese gefälschten Wahlscheine selbstverständlich nicht auf den Listen der für ungültig erklärten Wahlscheine auftauchen, können sie über diesen Weg auch nicht erkannt werden. Täuschend echt fälschen lassen sich die Unterlagen vor allem deshalb, da sie keinerlei Sicherheitsmerkmale enthalten. Das ist erstaunlich, da all diese Sicherheitsmerkmale (sowie auch oben beschriebener Überprüfungsmechanismus) bereits vorhanden waren, im Jahre 1989 durch die Erste Verordnung zur Änderung der Bundeswahlordnung jedoch abgeschafft wurden [175]. So wurden unter anderem sowohl das gestempelte Dienstsiegel (muss nur noch gedruckt werden) sowie die amtliche Unterschrift auf dem Wahlschein als auch Siegelmarken für die Stimmzettelumschläge 1989 ersatzlos abgeschafft und die Sicherheit der Briefwahl dadurch dramatisch verschlechtert. Aufgrund der fehlenden Siegelmarken können Wahlbriefe von Mitarbeitern der Verwaltung oder der Deutschen Post geöffnet und der Wahlschein verändert oder ausgetauscht werden. Die fehlenden Dienstiegel und amtlichen Unterschriften erleichtern es Betrügern, Wahlunterlagen zu fälschen und das Wahlergebnis durch die Abgabe von zusätzlichen Stimmen zu beeinflussen.

6.2.4 Fingerprinting der Papierstruktur

Zusätzlich zu den bereits genannten organisatorischen Angriffspunkten des aktuell in Deutschland eingesetzten Briefwahlverfahrens ist es außerdem möglich, das Wahlgeheimnis mit Hilfe der einmaligen Papierstruktur jedes einzelnen Wahlscheins zu brechen [176]. Hierfür muss vor dem Versand der Briefwahlunterlagen die Struktur eines jeden Wahlscheins erfasst und gemeinsam mit der Identität der wahlberechtigten Person gespeichert werden. Geeignete Gegenmaßnahmen hierfür wären z. B. Plattformen, über welche wahlberechtigte Personen die ihnen jeweils zugewiesenen Stimmzettel austauschen können. Noch besser, da der Austausch noch schlechter überwacht werden kann, sind öffentlich organisierte Treffen, auf denen man seinen Stimmzettel nach dem Zufallsprinzip mit anderen teilnehmenden, aber persönlich nicht bekannten Personen austauschen kann. Zur Erhöhung der Sicherheit kann dieser Vorgang beliebig oft wiederholt werden.

6.3 VERBESSUNGSVORSCHLÄGE

Das Briefwahlverfahren teilt sich einige für Distanzwahlsysteme in unkontrollierten Umgebungen charakteristische Probleme, vor allem im Hinblick auf Quittungsfreiheit sowie Nicht-Erpressbarkeit, mit den gängigen Internetwahlverfahren. Diesen Defiziten ist schwer nachzukommen, ohne den Charakter der Briefwahl, wie sie heute eingesetzt wird, grundlegend zu ändern. Nichtsdestotrotz werden im Folgenden verschiedene Ansätze präsentiert, die die Briefwahl für Bürgerinnen und Bürger nachvollziehbarer, sicherer und weniger anfällig für unabsichtliche Fehler machen würden.

Die in Unterabschnitt 6.2.1 beschriebenen Risiken könnten dadurch wesentlich verringert werden, dass die Beantragung entweder unter Vorlage des Personalausweises persönlich vor Ort oder aber

schriftlich lediglich unter Angabe der Wahlbenachrichtigungsnummer[2] (sowie für zusätzliche Sicherheit etvl. einer Kopie des Personalausweises) erfolgen kann. Im Falle der elektronischen Beantragung wäre ein Portal mit verschlüsselter Kommunikation über https, wie es für die Bundestagswahl 2017 von der Stadt München angeboten wurde (siehe Abbildung 26), zu bevorzugen.

(a) Teil 1 **(b)** Teil 2 **(c)** Teil 3

Abbildung 26: Screenshot des Portals der Stadt München zur Beantragung der Briefwahl [177].

Defizite bzgl. der Verifizierbarkeit des gesetzlich vorgesehenen Briefwahlprozesses, wie sie in Unterabschnitt 6.2.2 genannt werden, könnten durch eine gesetzlich vorgeschriebene Auskunftspflicht der Wahlbehörde gegenüber der Wählerin ausgeglichen werden. Hierzu könnte es der Wählerin z. B. ermöglicht werden, durch einen Anruf bei der Behörde festzustellen, ob die Unterlagen angekommen sind.

Weiterhin könnte den Wählerinnen nach der Wahl Einsicht in die entsprechenden Wahllisten der Briefwahlausschüsse gewährt werden, damit festgestellt werden kann, ob der Wahlschein am Wahltag überprüft und der Stimmzettel seinen weg in die Urne gefunden hat. Alternativ bzw. ergänzend könnte, wie von Reichmann [166] vorgeschlagen, ein QR- oder Strichcode auf die Briefwahlunterlagen gedruckt werden, der bei Ankunft des Wahlbriefs in der Behörde sowie beim Überprüfen des Wahlscheins am Wahltag gescannt wird und von der Wählerin online nachverfolgt werden kann (wobei sich hier natürlich wieder neue Fragen bzgl. der Sicherheit des dafür eingesetzten Systems ergeben würden). Die beiden Fragen, ob der Wahlbrief überhaupt bei der Wahlbehörde ankam und ob der Stimmzettel am Wahltag tatsächlich in der Wahlurne gelandet und damit in das Wahlergebnis eingeflossen ist, könnten auch auf diese automatisierte Art und Weise mit hoher Wahrscheinlichkeit abschließend beantwortet werden.

Der in Unterabschnitt 6.2.3 beklagten Tatsache, dass die Briefwahlunterlagen viel zu leicht zu fälschen sind, könnte entgegengewirkt werden, indem die 1989 abgeschafften Sicherheitsmerkmale, also Dienstsiegel und amtliche Unterschrift auf dem Wahlschein sowie Siegelmarken zum Verschließen des Wahlbriefes, wieder eingeführt werden. Entsprechende Überlegungen gibts es nach Einschätzung des Bundeswahlleiters jedoch nicht (siehe Abschnitt A.3). Ein einfacher Abgleich der eingegangenen Wahlbriefe mit der gesamten Wählerliste („Whitelisting “) am Wahltag würde das Risiko, dass gefälschte Wahlunterlagen nicht erkannt werden, ebenfalls drastisch reduzieren.

2 Wobei die Wahlbenachrichtigungsnummern dann so zufällig wie möglich zugewiesen werden sollten.

6.4 BEWERTUNG

Im Folgenden soll der in Deutschland gegenwärtig etablierte Briefwahlprozess auf Basis der in Kapitel 3 vorgestellten Kriterien für Wahlen erster Ordnung bewertet werden. Diese Methodik wurde eigentlich explizit für die Bewertung von Internetwahlverfahren in unkontrollierten Umgebungen entwickelt und ist an einigen Stellen, wie z. B. dem Wahlgeheimnis oder der Nicht-Erpressbarkeit, deshalb nicht eins zu eins auf die Briefwahl übertragbar. So ist es nach dem Absenden des Wahlbriefes z. B. nur noch Post- und Verwaltungsmitarbeitern in einem größeren Stil möglich, das Wahlgeheimnis zu brechen. Dies bedeutet zwar nach wie vor, dass es möglich ist, das Wahlgeheimnis zu brechen. Allerdings sind die Größe der Personengruppe, die potentiell zum solch einem Angriff in der Lage wäre, sowie die Skalierbarkeit der jeweiligen Angriffe wesentlich geringer, was einer der entscheidenden Vorteile der Briefwahl gegenüber Internetwahlsystemen ist.

6.4.1 Individuelle Verifizierbarkeit

Die innere Verifizierbarkeit ist von vornherein aufgrund der Tatsache ausgeschlossen, dass der Umschlag, in dem sich der Stimmzettel befindet, nach dem erfolgten Versand erst wieder bei der Auszählung geöffnet wird und zu diesem Zeitpunkt bereits in keiner Weise mehr der Wählerin zuzuordnen ist, um das Wahlgeheimnis zu gewährleisten. Laut Statistischem Bundesamt bzw. Bundeswahlleiter ist es derzeit gesetzlich nicht vorgesehen, dass sich Wählerinnen nach Versand des ausgefüllten Wahlbriefes nach dessen Verbleib erkundigen. Auch eine Einsichtnahme in die entsprechenden Wahllisten der Briefwahlausschüsse ist nicht erlaubt (siehe Abschnitt A.3). Aufgrund dessen ist auch die äußere und somit keinerlei individuelle Verifizierbarkeit gegeben.

6.4.2 Universelle Verifizierbarkeit

Aufgrund der grundlegenden Möglichkeit, jeden einzelnen Schritt der Auszählung vom Anfang bis zum Ende kritisch beizuwohnen, ist die diskrete Korrektheits-Verifizierbarkeit (KV.2) gegeben. Alle anderen Arten der universellen Verifizierbarkeit können nicht gewährleistet werden, da nicht kontrolliert werden kann, ob tatsächlich berechtigte Wählerinnen die Wahlunterlagen ausfüllen (keine Wahlberechtigungs-Verifizierbarkeit) und ob tatsächlich jede wahlberechtigte Person nur eine Stimme hat (Einmaligkeits-Verifizierbarkeit) oder z. B. Stimmen von anderen Personen kauft oder aufgrund von Politikverdrossenheit sogar „geschenkt“ bekommt.

6.4.3 Wahlgeheimnis

Dadurch, dass der komplette Wahlprozess bis zum Einwurf in den Postkasten bzw. die Abgabe bei der Wahlbehörde üblicherweise in einer unkontrollierten Umgebung stattfindet, ist es nicht auszuschließen, dass bereits während der Stimmabgabe das Wahlgeheimnis gebrochen wird. Der Umschlag mit dem Stimmzettel sowie der Wahlschein, der den Namen der Wählerin enthält, werden außerdem im selben Umschlag transportiert, der nicht dagegen gesichert ist, dass ihn Dritte öffnen und somit das Wahlgeheimnis verletzen. Zwar sollte nach dem Einwurf in den Postkasten nur noch Post- und Verwaltungspersonal Zugriff auf die Wahlbriefe haben, allerdings stellt auch dies ein potentielles Risiko dar, wie einschlägige Pressemeldungen belegen [171–173, 178, 179]. Das Wahlgeheimnis und die Quittungsfreiheit können deshalb nicht gewährleistet werden.

6.4.4 Nicht-Erpressbarkeit

Bis auf die eidesstattliche Erklärung, dass man den Stimmzettel selbst und ohne Fremdeinwirkung ausgefüllt hat, existieren keinerlei Mechanismen, die entsprechende Angriffe abwehren könnten. Die eidesstattliche Erklärung selbst stellt für potentielle Erpresser kein nennenswertes Hindernis dar. Aller-

dings ist es grundsätzlich möglich, innerhalb einer bestimmten Frist, einen neuen Wahlschein bei der Wahlbehörde zu beantragen, womit der alte ungültig und bei der Stimmauszählung vom Briefwahlvorstand nicht mehr akzeptiert wird. Es ist dadurch in begrenztem Umfang möglich, dem Angreifer eine (end-)gültige Wahl vorzutäuschen, indem man z. B. den Wahlbrief unter seiner Beobachtung in einen Postkasten wirft, danach aber bei der Wahlbehörde einen neuen Wahlschein zu beantragen. Dies stellt eine Maßnahme gegen Abwesenheits- (AA.1), Randomisierungs- (RA.1), sowie Simulationsangriffe (SA.1) dar.

6.4.5 Robustheit

Da die Wahlbriefe von jedermann zu öffnen sind und dieser Vorgang keinerlei Spezialwissen voraussetzt, ist die innere Robustheit (RI.1) gewährleistet.

6.4.6 Benutzbarkeit der Wahl

Eigentlich lassen sich die Kriterien der Benutzbarkeit der Wahl nicht eins zu eins auf Briefwahlen übertragen. Da der eigentliche Vorgang der Wahl bei der Briefwahl jedoch sehr simpel ist, wird dieser an dieser Stelle mit dem bestmöglichen Kriterium, nämlich „Klick-Voting“ mit einmaliger Teilnahme ohne Hilfsmittel (BW1.1), bewertet.

6.4.7 Benutzbarkeit der individuellen Verifizierung

Wie bereits erwähnt, ist keine Möglichkeit der individuellen Verifizierbarkeit gegeben. Demzufolge kann auch ihre Benutzbarkeit nicht bewertet werden.

Tabelle 13: Bewertung des in Deutschland etablierten Briefwahlverfahrens unter Annahme des Angreifermodells für Wahlen erster Ordnung.

	Verifizierbarkeit		**Wahl-geheim-nis**	**Nicht-Erpress-barkeit**	**Robust-heit**	**Benutz-barkeit**	
	ind.	**uni.**				**Wahl**	**Verif.**
§§ 25, 28 Bundes-wahlordnung	x	KV.2	x	AA.1, RA.1, SA.1	RI.1	BW.1.1	x

7 FAZIT UND AUSBLICK

Im Rahmen der während der Abfassung der vorliegenden Arbeit stattfindenden Bundestagswahlen 2017 wurde auch die Sicherheit des offiziell eingesetzten Briefwahlprozesses untersucht. Zwar überwiegen nach wie vor die Sicherheitsbedenken gegenüber Internetwahlverfahren, wenn es um die Frage geht, ob reguläre Papierwahlen durch Internetwahlen ersetzt werden sollen. Aufgrund der festgestellten Mängel im Briefwahlprozess stellt sich jedoch die Frage, ob Internetwahlen nicht wenigstens als Alternative zur Briefwahl für im Ausland lebende deutsche Staatsangehörige angeboten werden sollten, die ihre Wahlbriefe nicht direkt bei deutschen Auslandsvertretungen abgeben können und dadurch auf die Nutzung von durch fremde Regierungen kontrollierte bzw. unzuverlässige Infrastruktur angewiesen sind. Eine großflächige Ablösung der Briefwahl durch Internetwahlverfahren sehen die Autoren im Hinblick auf den gegenwärtigen Stand der Technik hingegen als nicht vertretbar an. Zwar gibt es durchaus Internetwahlverfahren, die (mit Ausnahme der Nicht-Erpressbarkeit) mit dem Briefwahlprozess vergleichbar sind und sogar zusätzliche Eigenschaften, wie z. B. die individuelle Verifizierbarkeit erfüllen. Allerdings hat die Briefwahl den entscheidenden Vorteil, dass Angriffe auf sie nicht bzw. nur sehr schwer skalierbar sind. Außerdem wäre durch relativ einfache organisatorische Vorkehrungen auch bei der Briefwahl eine umfassende individuelle Verifizierbarkeit in mehrere möglichen Formen möglich. Doch selbst ohne diese wünschenswerten zusätzlichen Maßnahmen erscheint den Autoren die Briefwahl aufgrund der nur sehr schwer zu skalierenden Angriffe als die bessere Alternative im direkten Vergleich mit den untersuchten Internetwahlverfahren.

Das liegt nicht zuletzt daran, dass die meisten der heute tatsächlich produktiv eingesetzt Internetwahlsysteme lediglich auf Funktionstrennung und Systemsicherheit gepaart mit üblichen symmetrischen sowie asymmetrischen kryptografischen Verfahren basieren. Doch selbst, wenn man der Funktionstrennung und den Kontrollmechanismen zwischen unterschiedlichen Parteien vertraut, ist es noch ein langer Weg, auch Vertrauen für die jeweiligen Systeme aufzubauen, die für die unterschiedlichen Funktionen zuständig sind. Dies liegt nicht zuletzt an der enormen Komplexität der heutzutage eingesetzten Systeme. Ihre Komplexität ist dabei vor allem eine Folge ihrer Vielseitigkeit. Ein Ansatz, die Komplexität zumindest auf der Server-Seite zu verringern und dadurch Sicherheit und auch das Vertrauen in die Systeme zu erhöhen, ist es deshalb, sie so schlank wie möglich zu gestalten. Das heißt nur diejenigen Funktionen, die für die Ausführung der Wahl zwingend notwendig sind, so hardwarenah wie möglich zu implementieren, um die Angriffsfläche zu minimieren. Einige Ansätze hierfür sind z. B. die Microkernel-Familie *L4*[1], an deren Entwicklung unter anderem *IBM Research*, *National ICT Australia (NICTA)* sowie die US-amerikanische *Defence Advanced Research Projects Agency (DARPA)* beteiligt waren, oder das an der *Yale University* entwickelte *CertiKOS*[2] [180].

Das estnische Wahlsystem realisiert, trotz des unsicheren Designs und fehlender kryptografischer Absicherungen, einige interessante Ideen an organisatorischen Maßnahmen. Eine dieser Maßnahmen ist Re-Voting, was normalerweise in jedes Protokoll integriert werden kann. Dies ermöglicht dem Wähler, sich trotz unkontrollierter Umgebung in gewisser Weise der Kontrolle von physisch anwesenden Angreifern zu entziehen. Ein weiterer guter Ansatz des estnischen Wahlsystems ist die Integration der Internetwahl in ein klassisches Wahlsystem (papierbasierte Präsenzwahl mit Briefwahl), wobei die Internetwahl lediglich optional angeboten wird. Dies hat viele Vorteile bezüglich verschiedenster Kriterien, welche in unserem Bewertungsschema aufgrund der technischen und theoretischen Sichtweise jedoch nicht betrachtet werden. Unter anderem ist somit sichergestellt, dass niemand aufgrund materieller Aspekte (z. B. aufgrund des Zugangs zu einem Computer) oder fehlender technischer Kenntnisse (z. B. Bedienung eines Internetwahlsystems) ausgeschlossen wird. Außerdem ermöglicht dieses Vorge-

1 http://l4hq.org/
2 http://flint.cs.yale.edu/certikos/

hen, dass die Wahl trotz etwaiger Denial-of-Service Angriffe auf das Internetwahlsystem durchgeführt werden kann.

Eine weitere große Herausforderung, vor der Entwickler und Wissenschaftler im Bezug auf Internetwahlverfahren für unkontrollierte Umgebungen stehen, ist das Secure Platform Problem, also die Annahme, dass jegliches Gerät auf Client-Seite, das zur Wahl über das Internet benutzt wird, potentiell durch Malware o. ä. manipuliert sein kann. Es ist deshalb so herausfordernd, da sich die Geräte des Wählers jeglicher Kontrolle durch die Organisatoren einer Wahl entziehen. Mit den Verfahren des Code Votings (siehe Abschnitt 4.5) wurde eine effektive Gegenmaßnahme entwickelt, die es Wählern ermöglicht, auch auf potentiell manipulierten Computern zu wählen, ohne das Risiko einzugehen, dass ihr Wahlgeheimnis verletzt wird. Allerdings bringt Code Voting wieder neue Herausforderungen mit sich. So ist es dem Wähler durch das notwendige Code Sheet z. B. möglich, seine Wahl anderen gegenüber zu beweisen, wodurch wiederum mit der Quittungsfreiheit und der Nicht-Erpressbarkeit gebrochen wird.

Du-Vote (siehe Abschnitt 5.3), ein hybrides Wahlverfahren, baut auf den Stärken des Code Votings und verschiedener anderer Techniken auf, die in bisherigen Wahlverfahren eingesetzt werden, um eben diesen Schwächen beizukommen. Bei aller Raffinesse des Verfahrens hat jedoch auch dieses Schwächen, wie z. B. die sehr umständliche Nutzung, die in der praktischen Anwendung von Wahlverfahren eine Schlüsselrolle spielt, da sie eine Grundanforderung für die Akzeptanz bei der Wählerschaft ist. Auch wenn Du-Vote die Erwartungen nicht zur Gänze erfüllt, ist es doch ein wunderbares Beispiel dafür, wohin die Zukunft der Internetwahlverfahren führt. Einzelne Technologien, wie z. B. blinde Signaturen, homomorphe Verschlüsselung oder Mix-Nets sind wichtige Elemente, bieten für sich allein jedoch keine allumfassende Lösung, um den äußerst komplexen Anforderungen von Internetwahlen gerecht zu werden.

Auch die Blockchain-Technologie (siehe Abschnitt 4.7), welche in der öffentlichen Diskussion seit einiger Zeit als eine Art „Allheilmittel“ für allerlei Probleme angesehen wird, hilft bei Internetwahlen in unkontrollierten Umgebungen nur sehr bedingt. Zwar wurden mit aufkommender Popularität viele Verfahren veröffentlicht, die die Blockchain auf die eine oder andere Art und Weise miteinbezogen. Diese waren im Kern jedoch meist nicht neu, sondern ersetzen lediglich die bereits in den 1980er Jahren aufgekommene, meist lediglich abstrakt beschriebene Idee des öffentlich lesbaren, oftmals nur durch autorisierte Instanzen beschreibbaren und nicht wieder löschbaren Bulletin Boards durch eine konkrete Technik. Der richtig große Durchbruch, der beispielsweise das Vertrauen ersetzt, das man bisher in die Instanzen setzen muss, die die Wahl organisieren, blieb bislang leider aus. Ein weiteres Manko vieler auf Blockchain basierender Verfahren ist, dass sie explizit davon ausgehen, dass die erfolgreiche Authentifizierung der Wähler bereits anderweitig stattgefunden hat, ohne jedoch näher darauf einzugehen. Hier könnten die im Hinblick auf Internetwahlen bisher nur wenig betrachteten Attribute-Based Credentials (siehe Abschnitt 4.6) oder aber die auch bereits anderweitig vielfach eingesetzten blinden Signaturen (siehe Abschnitt 4.2) zum Einsatz kommen und zusammen mit der Blockchain zwei vielversprechende Bausteine für zukünftige Internetwahlverfahren liefern, die dabei helfen können, das Wahlgeheimnis sicherzustellen bzw. Ende-zu-Ende-Verifizierbarkeit zu gewährleisten. Außerdem wäre es interessant, im Rahmen zukünftiger Arbeiten herauszufinden, ob auch andere Internetwahlverfahren auf der Blockchain abgebildet werden können, um z. B. deren Verifizierbarkeit zu verbessern.

Aus oben genannten Gründen werden in Zukunft vermehrt hybride Verfahren notwendig sein, die die Stärken der einzelnen Techniken miteinander kombinieren. Die Autoren der vorliegenden Arbeit hoffen deshalb, ein Orientierungswerk zu bieten, das dabei hilft, die schiere Menge an Ansätzen zu überblicken und trotz ihrer Komplexität ein Verständnis für die verschiedenen Anforderungen an Internetwahlsysteme in unkontrollierten Umgebungen aufzubauen. Wie selbst das Bundesverfassungsgericht bereits bestätigte (siehe Unterabschnitt 2.2.2), ist Verständnis auch einer der entscheidenden Punkte, wenn es darum geht, ob ein Wahlverfahren (zumindest für politische Wahlen) zulässig ist oder nicht. Demzufolge sollte Verständnis auch einer der Eckpfeiler der öffentlichen Debatte um die Einführung von Online-Wahlen an Hochschulen sein. Die zu Beginn der Arbeit gestellte Frage, warum

Online-Wahlen nicht auch an Hochschulen möglich sein sollten, soll hier deshalb unter verschiedenene Blickwinkeln beleuchtet werden.

Zum einen ist da offensichtlich die technische Sichtweise. Im Laufe der Arbeit hat sich immer mehr herauskristallisiert, dass „das eine“ Internetwahlverfahren zum Einsatz in unkontrollierten Umgebungen nicht existiert. So gibt es, wie bereits erwähnt, eine nur sehr schwer zu überblickende Anzahl an verschiedenen Verfahren, die alle ihre Stärken und Schwächen haben. Doch selbst, wenn man diesen Dschungel unterschiedlichster Verfahren überblickt, stellt sich immer noch die Frage, welches Wahlverfahren die notwendigen technischen Anforderungen am ehesten erfüllt bzw. wie diese Anforderungen und das zugehörige Angreifermodell überhaupt aussehen. Die Formulierung „am ehesten“ ist dabei mit Bedacht gewählt, da es eher unwahrscheinlich ist, dass ein Internetwahlverfahren exakt die Anforderungen erfüllt, die im Vorfeld unabhängig definiert wurden. Obwohl diese Frage nach den Anforderungen und dem Angreifermodell durchaus technischer Natur ist, ist sie (selbstverständlich unter Einbeziehung technischer Aspekte) letztlich politisch bzw. juristisch zu beantworten. Wenn man nun davon ausginge, dass diese Anforderungen und das zugehörige Angreifermodell nicht allzu anspruchsvoll definiert werden würden, könnte man zu dem Schluss kommen, dass durchaus Internetwahlsysteme existieren, die zur Wahl studentischer Gremien an Hochschulen eingesetzt werden könnten. Diese Annahme im Hinterkopf behaltend und auf das am Anfang des Absatzes gefallene Stichwort „Verständis“ zurückkommend, stellt sich nun natürlich auch die Frage nach etwaigen Konsequenzen der Einführung eines Internetwahlsystems an Hochschulen. Abgesehen davon, dass potentielle Wähler verstehen sollten, wie das letztendlich eingesetzte System funktioniert, ist es auch wichtig, dass sie über die Komplexität des Themas an für sich aufgeklärt werden. Geschieht diese Aufklärung nicht, besteht die Gefahr, dass junge Menschen, die im Laufe ihres Studiums durchaus auch zur politischen Teilhabe ermutigt und befähigt werden sollen, sich jedoch nicht mehrere Monate mit dem Themenkomplex „Internetwahlverfahren in unkontrollierten Umgebungen“ auseinandersetzen, sich später die Frage stellen, warum Internetwahlsysteme nicht auch in anderen Bereichen des Lebens (möglicherweise bis hin zur Bundestagswahl) eingeführt werden sollten. Dem gegenüber steht selbstverständlich die ebenfalls valide Argumentation, wo Internetwahlsysteme denn sonst praktisch erprobt werden sollten, wenn nicht in einer solch kritischen, informierten und zugleich innovativen Umgebung, wie sie nun mal nur die Hochschule zu bieten hat. Wie aufgezeigt wurde, ist die Beantwortung der Frage nach der Einführung von Internetwahlsystemen an Hochschulen selbst nach monatelangem Abwägen nach wie vor sehr ambivalent und kann von den Autoren deshalb nicht abschließend beantwortet werden. Umso wichtiger ist es deshalb, dass sich Informatiker bzgl. der immer wieder aufkommenden Debatten um die Einführung von Internetwahlen informieren und aktiv in diese einmischen.

LITERATURVERZEICHNIS

[1] U.S. Election Assistance Commission. *Testing and Certification Technical Paper #2: A Survey of Internet Voting*. Technischer Bericht. Sep. 2011 (siehe Seiten 1–3).

[2] Stephan Neumann und Melanie Volkamer. „A Holistic Framework for the Evaluation of Internet Voting Systems“. In: *Design, Development, and Use of Secure Electronic Voting Systems*. Hershey, PA, USA: IGI Global, 2014, Seiten 76–91 (siehe Seiten 1, 3, 18).

[3] Lucie Langer et al. „Towards a Framework on the Security Requirements for Electronic Voting Protocols“. In: *Proceedings of the 2009 First International Workshop on Requirements Engineering for e-Voting Systems*. RE-VOTE '09. Washington, DC, USA: IEEE Computer Society, 2009, Seiten 61–68 (siehe Seiten 1, 3, 4, 13–16, 18, 20).

[4] Landtag von Baden-Württemberg (15. Wahlperiode). „Online-Wahlen an Hochschulen“. In: *Drucksache 15 / 7053* (Juni 2016). URL: http://www.landtag-bw.de/files/live/sites/LTBW/files/dokumente/WP15/Drucksachen/7000/15_7053_D.pdf (siehe Seite 1).

[5] Thüringer Oberverwaltungsgericht. „Urteil vom 30. Mai 2013, Az. 1 N 240/12“. In: (Mai 2013). URL: http://www.thovg.thueringen.de/webthfj/webthfj.nsf/C4CFE728AD19702AC125814B0038FC0F/%5C$File/15-4N-00124-U-A.pdf?OpenElement (siehe Seite 1).

[6] Mona F. M. Mursi et al. „On the development of electronic voting: a survey“. In: *International Journal of Computer Applications* 61.16 (2013) (siehe Seiten 2, 3).

[7] Alexander Schneider, Christian Meter und Philipp Hagemeister. „Survey on Remote Electronic Voting“. In: *CoRR* abs/1702.02798 (2017) (siehe Seite 3).

[8] Komminist Weldemariam und Adolfo Villafiorita. „A Survey: Electronic Voting Development and Trends“. In: *Electronic Voting 2010, EVOTE 2010, 4th International Conference, Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting.CC, July 21st - 24th, 2010, in Castle Hofen, Bregenz, Austria*. 2010, Seiten 119–131 (siehe Seite 3).

[9] Stephan Neumann. „Evaluation and Improvement of Internet Voting Schemes Based on Legally-Founded Security Requirements“. Dissertation. Technische Universität Darmstadt, 2016 (siehe Seite 3).

[10] Krishna Sampigethaya und Radha Poovendran. „A Framework and Taxonomy for Comparison of Electronic Voting Schemes“. In: *Computers and Security* 25.2 (März 2006), Seiten 137–153 (siehe Seiten 3, 14, 15).

[11] Barbara Ondrisek. „Sicherheit elektronischer Wahlen - Eine Methode zur Bewertung und Optimierung der Sicherheit von E-Voting-Systemen“. Dissertation. Technische Universität Wien, 2008 (siehe Seite 4).

[12] Jörg Helbach. „Eingrenzung des Secure Platform Problems bei Internetwahlsystemen mit Hilfe von Code Voting“. Dissertation. Ruhr-Universität Bochum, 2010 (siehe Seiten 5, 23).

[13] Jörg Helbach. „Code Voting - Ein Verfahren für Aktiengesellschaften?“ In: *GI Jahrestagung*. 2008 (siehe Seite 5).

[14] Anna Dopatka. „E-Voting in Deutschland? Zum Problem der Stimmabgabe über das Internet bei politischen Wahlen“. Magisterarbeit. Univärsität Hildesheim, Fachbereich III Informations- und Kommunikationswissenschaften, Institut für Angewandte Sprachwissenschaft, 2005 (siehe Seiten 6, 9).

[15] Wolfgang Schreiber. *Handbuch des Wahlrechts zum Deutschen Bundestag*. Carl Heymanns, 2002. ISBN: 978-3452251411 (siehe Seite 6).

[16] Carmen Kempka Bernhard Löwe. „Kryptographische Wahlverfahren“. Vorlesungsskript. Institut für Kryptographie und Sicherheit Europäisches Institut für Systemsicherheit Fakultät für Informatik Karlsruher Institut für Technologie, 2014 (siehe Seiten 7–10, 36, 80, 112–117).

[17] Bundesverfassungsgericht. „2 BvC 3/07 Urteil vom 03. März 2009“. In: (März 2009). URL: http://www.bundesverfassungsgericht.de/entscheidungen/cs20090303_2bvc000307.html (siehe Seite 7).

[18] Andreas von Arnauld. *Völkerrecht*. Heidelberg: C. F. Müller, 2012. ISBN: 978-3-8114-9761-0 (siehe Seite 7).

[19] Lilian Mitrou, Dimitris Gritzalis und Sokratis Katsikas. „Revisiting Legal and Regulatory Requirements for Secure E-Voting“. In: *Security in the Information Society: Visions and Perspectives*. Herausgegeben von M. Adeeb Ghonaimy, Mahmoud T. El-Hadidi und Heba K. Aslan. Springer US, 2002, Seiten 469–480. ISBN: 978-0-387-35586-3 (siehe Seiten 8, 9).

[20] Horng-Twu Liaw. „A secure electronic voting protocol for general elections“. In: *Computers & Security* 23.2 (2004), Seiten 107–119 (siehe Seiten 8, 9, 25).

[21] Council of Europe. *Legal, Operational anf Technical Standards for E-Voting*. Council of Europe Publishing, 2005. ISBN: 92-871-5635-2. URL: http://www.coe.int/t/dgap/democracy/Activities/Key-Texts/Recommendations/Rec%282004%2911_Eng_Evoting_and_Expl_Memo_en.pdf (siehe Seiten 8–10).

[22] Lucie Langer. „Privacy and Verifiability in Electronic Voting“. Dissertation. Technische Universität Darmstadt, 2010 (siehe Seiten 13, 15, 16, 20).

[23] Ronald L Rivest. „The ThreeBallot voting system“. In: (2006) (siehe Seite 13).

[24] Stéphanie Delaune, Steve Kremer und Mark Ryan. „Verifying privacy-type properties of electronic voting protocols“. In: *Journal of Computer Security* 17.4 (2009), Seiten 435–487 (siehe Seiten 14, 15).

[25] Costas Lambrinoudakis et al. „Secure Electronic Voting: the Current Landscape“. In: *Secure Electronic Voting*. 2003, Seiten 101–122 (siehe Seite 14).

[26] Warren D Smith. „New cryptographic election protocol with best-known theoretical properties“. In: *Proc. of Workshop on Frontiers in Electronic Elections*. 2005, Seiten 1–14 (siehe Seiten 14, 29).

[27] Andreu Riera. *An introduction to electronic voting schemes*. Universitat Autònoma de Barcelona, Departament d'Informàtica, 1998 (siehe Seite 14).

[28] Tal Moran und Moni Naor. „Receipt-Free Universally-Verifiable Voting with Everlasting Privacy“. In: *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*. 2006, Seiten 373–392 (siehe Seite 16).

[29] Ari Juels, Dario Catalano und Markus Jakobsson. „Coercion-resistant Electronic Elections“. In: *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*. New York, NY, USA: ACM, 2005, Seiten 61–70. ISBN: 1-59593-228-3 (siehe Seiten 17, 27, 32, 81).

[30] Ralf Küsters, Tomasz Truderung und Andreas Vogt. „A game-based definition of coercion resistance and its applications“. In: *Journal of Computer Security* 20.6 (2012), Seiten 709–764 (siehe Seite 17).

[31] Maina Olembo und Melanie Volkamer. „E-voting system usability: Lessons for interface design, user studies, and usability criteria“. In: (Jan. 2013), Seiten 172–201 (siehe Seite 18).

[32] International Organization for Standardization. *ISO 9241-11: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs): Part 11: Guidance on Usability*. 1998 (siehe Seite 18).

[33] Atsushi Fujioka, Tatsuaki Okamoto und Kazuo Ohta. „A Practical Secret Voting Scheme for Large Scale Elections“. In: *Advances in Cryptology - AUSCRYPT '92, Workshop on the Theory and Application of Cryptographic Techniques, Gold Coast, Queensland, Australia, December 13-16, 1992, Proceedings*. 1992, Seiten 244–251 (siehe Seiten 19, 23).

[34] Rüdiger Grimm et al. „Erfahrungen mit Online-Wahlen für Vereinsgremien“. In: *Datenschutz und Datensicherheit - DuD* 33.2 (2009), Seiten 97–101 (siehe Seiten 22, 64, 66, 67, 69, 70).

[35] Bundesamt für Sicherheit in der Informationstechnik. *Common Criteria Protection Profile for Basic set of security requirements for Online Voting Products (BSI-CC-PP-0037)*. Technischer Bericht. 2008. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Reporte/ReportePP/pp0037b_engl_pdf.pdf;jsessionid=4947AF4328A53139FB3765299F6757BE.2_cid360?__blob=publicationFile&v=1 (besucht am 14. 07. 2017) (siehe Seiten 22, 64).

[36] Stephan Neumann et al. „Cast-as-intended-Verifizierbarkeit für das Polyas-Internetwahlsystem“. In: *Datenschutz und Datensicherheit - DuD* 39.11 (2015), Seiten 747–752 (siehe Seiten 23, 64–67, 69–71).

[37] Niels Menke und Kai Reinhard. „Compliance of POLYAS with the Common Criteria Protection Profile - A 2010 Outlook on Certified Remote Electronic Voting“. In: *Electronic Voting 2010, EVOTE 2010, 4th International Conference, Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting.CC, July 21st - 24th, 2010, in Castle Hofen, Bregenz, Austria*. 2010, Seiten 109–118 (siehe Seiten 23, 64–67, 69–71).

[38] David Chaum. „Elections with Unconditionally-Secret Ballots and Disruption Equivalent to Breaking RSA“. In: *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*. 1988, Seiten 177–182 (siehe Seite 23).

[39] Colin Boyd. „A New Multiple Key Cipher and an Improved Voting Scheme“. In: *Advances in Cryptology - EUROCRYPT '89, Workshop on the Theory and Application of of Cryptographic Techniques, Houthalen, Belgium, April 10-13, 1989, Proceedings*. 1989, Seiten 617–625 (siehe Seite 23).

[40] Tatsuaki Okamoto und Kazuo Ohta. „Divertible Zero Knowledge Interactive Proofs and Commutative Random Self-Reducibility“. In: *Advances in Cryptology - EUROCRYPT '89, Workshop on the Theory and Application of of Cryptographic Techniques, Houthalen, Belgium, April 10-13, 1989, Proceedings*. 1989, Seiten 134–148 (siehe Seite 23).

[41] Lorrie Faith Cranor und Ron Cytron. „Sensus: A Security-Conscious Electronic Polling System for the Internet“. In: *30th Annual Hawaii International Conference on System Sciences (HICSS-30), 7-10 January 1997, Maui, Hawaii, USA*. 1997, Seiten 561–570 (siehe Seite 24).

[42] Tatsuaki Okamoto. „An electronic voting scheme“. In: *IFIP World Conference on IT Tools*. 1996, Seiten 21–30 (siehe Seiten 24, 25).

[43] Tatsuaki Okamoto. „Receipt-Free Electronic Voting Schemes for Large Scale Elections“. In: *Security Protocols, 5th International Workshop, Paris, France, April 7-9, 1997, Proceedings*. 1997, Seiten 25–35 (siehe Seiten 24, 25).

[44] Miyako Ohkubo et al. „An Improvement on a Practical Secret Voting Scheme“. In: *Information Security, Second International Workshop, ISW'99, Kuala Lumpur, Malaysia, November 1999, Proceedings*. 1999, Seiten 225–234 (siehe Seiten 24, 25).

[45] Mark Allan Herschberg. „Secure electronic voting over the world wide web“. Dissertation. Massachusetts Institute of Technology, 1997 (siehe Seite 25).

[46] Brandon William DuRette. „Multiple administrators for electronic voting“. In: *Bachelor thesis, Massachusetts Institute of Technology, Boston, USA* (1999) (siehe Seite 25).

[47] Rui Joaquim, André Zúquete und Paulo Ferreira. „REVS–a robust electronic voting system“. In: *IADIS International Journal of WWW/Internet* 1.2 (2003), Seiten 47–63 (siehe Seite 25).

[48] Ricardo Lebre et al. „Internet voting: Improving resistance to malicious servers in REVS“. In: *Proc. of IADIS International Conference on Applied Computing*. 2004 (siehe Seite 25).

[49] Wen-Shenq Juang und Chin-Laung Lei. „A collision-free secret ballot protocol for computerized general elections“. In: *Computers & Security* 15.4 (1996), Seiten 339–348 (siehe Seite 25).

[50] Jared Karro und Jie Wang. „Towards a Practical, Secure, and Very Large Scale Online Election“. In: *15th Annual Computer Security Applications Conference (ACSAC 1999), 6-10 December 1999, Scottsdale, AZ, USA*. 1999, Seiten 161–169 (siehe Seite 25).

[51] Gianluca Dini. „A secure and available electronic voting service for a large-scale distributed system“. In: *Future Generation Comp. Syst.* 19.1 (2003), Seiten 69–85 (siehe Seite 25).

[52] Chin-Ling Chen et al. „A secure anonymous e-voting system based on discrete logarithm problem“. In: *Applied Mathematics & Information Sciences* 8.5 (2014), Seite 2571 (siehe Seite 25).

[53] M. R. Clarkson, S. Chong und A. C. Myers. „Civitas: Toward a Secure Voting System“. In: *2008 IEEE Symposium on Security and Privacy*. Mai 2008, Seiten 354–368 (siehe Seiten 26, 28, 81).

[54] David Chaum. „Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms“. In: *Commun. ACM* 24.2 (1981), Seiten 84–88 (siehe Seiten 26, 27, 34, 115).

[55] Choonsik Park, Kazutomo Itoh und Kaoru Kurosawa. „Efficient Anonymous Channel and All/Nothing Election Scheme“. In: *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*. 1993, Seiten 248–259 (siehe Seite 27).

[56] Kazue Sako und Joe Kilian. „Receipt-Free Mix-Type Voting Scheme - A Practical Solution to the Implementation of a Voting Booth“. In: *Advances in Cryptology - EUROCRYPT '95, International Conference on the Theory and Application of Cryptographic Techniques, Saint-Malo, France, May 21-25, 1995, Proceeding*. 1995, Seiten 393–403 (siehe Seite 27).

[57] Josh Cohen Benaloh und Dwight Tuinstra. „Receipt-free secret-ballot elections (extended abstract)“. In: *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*. 1994, Seiten 544–553 (siehe Seiten 27, 31–33).

[58] Rosario Gennaro. *Using non-interactive proofs to achieve independence efficiently and securely*. Massachusetts Institute of Technology. Laboratory for Computer Science, 1994 (siehe Seite 27).

[59] Wakaha Ogata et al. „Fault tolerant anonymous channel“. In: *Information and Communication Security, First International Conference, ICICS'97, Beijing, China, November 11-14, 1997, Proceedings*. 1997, Seiten 440–444 (siehe Seite 27).

[60] Masayuki Abe. „Universally Verifiable Mix-net with Verification Work Indendent of the Number of Mix-servers“. In: *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*. 1998, Seiten 437–447 (siehe Seite 27).

[61] Masayuki Abe. „Mix-Networks on Permutation Networks“. In: *Advances in Cryptology - ASIACRYPT '99, International Conference on the Theory and Applications of Cryptology and Information Security, Singapore, November 14-18, 1999, Proceedings*. 1999, Seiten 258–273 (siehe Seiten 27, 115).

[62] Markus Jakobsson. „A Practical Mix“. In: *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*. 1998, Seiten 448–461 (siehe Seiten 27, 115).

[63] C. Andrew Neff. „A verifiable secret shuffle and its application to e-voting“. In: *CCS 2001, Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, Pennsylvania, USA, November 6-8, 2001*. 2001, Seiten 116–125 (siehe Seite 27).

[64] Jun Furukawa und Kazue Sako. „An Efficient Scheme for Proving a Shuffle“. In: *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*. 2001, Seiten 368–387 (siehe Seite 27).

[65] Ari Juels, Dario Catalano und Markus Jakobsson. „Coercion-Resistant Electronic Elections“. In: *IACR Cryptology ePrint Archive* 2002 (2002), Seite 165. URL: http://eprint.iacr.org/2002/165 (siehe Seiten 27, 28, 34).

[66] David Chaum. „Security Without Identification: Transaction Systems to Make Big Brother Obsolete“. In: *Commun. ACM* 28.10 (Okt. 1985), Seiten 1030–1044 (siehe Seite 27).

[67] Ronald Cramer und Victor Shoup. „A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack“. In: *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*. 1998, Seiten 13–25 (siehe Seite 28).

[68] Philip D. MacKenzie, Thomas Shrimpton und Markus Jakobsson. „Threshold Password-Authenticated Key Exchange“. In: *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*. 2002, Seiten 385–400 (siehe Seite 28).

[69] Stefan G. Weber, Roberto Araujo und Johannes A. Buchmann. „On Coercion-Resistant Electronic Elections with Linear Work“. In: *Proceedings of the The Second International Conference on Availability, Reliability and Security, ARES 2007, The International Dependability Conference - Bridging Theory and Practice, April 10-13 2007, Vienna, Austria*. 2007, Seiten 908–916 (siehe Seite 29).

[70] Roberto Araújo, Sébastien Foulle und Jacques Traoré. „A practical and secure coercion-resistant scheme for remote elections“. In: *Frontiers of Electronic Voting, 29.07. - 03.08.2007*. 2007 (siehe Seite 29).

[71] Roberto Araújo, Sébastien Foulle und Jacques Traoré. „A Practical and Secure Coercion-Resistant Scheme for Internet Voting“. In: *Towards Trustworthy Elections, New Directions in Electronic Voting*. 2010, Seiten 330–342 (siehe Seite 29).

[72] Oliver Spycher et al. „A New Approach towards Coercion-Resistant Remote E-Voting in Linear Time“. In: *Financial Cryptography and Data Security - 15th International Conference, FC 2011, Gros Islet, St. Lucia, February 28 - March 4, 2011, Revised Selected Papers*. 2011, Seiten 182–189 (siehe Seite 29).

[73] Michael Schlaepfer et al. „Efficient Vote Authorization in Coercion-Resistant Internet Voting“. In: *E-Voting and Identity: Third International Conference, VoteID 2011*. Berlin, Heidelberg: Springer, 2012, Seiten 71–88. ISBN: 978-3-642-32747-6 (siehe Seite 29).

[74] F. Shirazi et al. „Robust electronic voting: Introducing robustness in Civitas“. In: *2011 International Workshop on Requirements Engineering for Electronic Voting Systems*. Aug. 2011, Seiten 47–55 (siehe Seite 29).

[75] Sergiu Bursuc, Gurchetan S. Grewal und Mark D. Ryan. „Trivitas: Voters Directly Verifying Votes“. In: *E-Voting and Identity: Third International Conference, VoteID 2011*. Berlin, Heidelberg: Springer, 2012, Seiten 190–207. ISBN: 978-3-642-32747-6 (siehe Seite 29).

[76] Ben Adida. „Helios: Web-based Open-audit Voting“. In: *Proceedings of the 17th Conference on Security Symposium*. SS'08. San Jose, CA: USENIX Association, 2008, Seiten 335–348 (siehe Seiten 29, 34, 81).

[77] Jeremy Clark und Urs Hengartner. „Selections: Internet Voting with Over-the-Shoulder Coercion-Resistance“. In: *Financial Cryptography and Data Security: 15th International Conference*. Berlin, Heidelberg: Springer, 2012, Seiten 47–61. ISBN: 978-3-642-27576-0 (siehe Seite 29).

[78] S. Neumann und M. Volkamer. „Civitas and the Real World: Problems and Solutions from a Practical Point of View“. In: *2012 Seventh International Conference on Availability, Reliability and Security*. Aug. 2012, Seiten 180–185 (siehe Seite 29).

[79] Stephan Neumann et al. „Towards A Practical JCJ/Civitas Implementation.“ In: 2013 (2013), Seite 464 (siehe Seite 29).

[80] Craig Gentry. „A fully homomorphic encryption scheme“. `crypto.stanford.edu/craig`. Dissertation. Stanford University, 2009 (siehe Seite 30).

[81] Josh D. Cohen und Michael J. Fischer. „A Robust and Verifiable Cryptographically Secure Election Scheme (ExtendedAbstract)“. In: *26th Annual Symposium on Foundations of Computer Science, Portland,Oregon, USA, 21-23 October 1985*. 1985, Seiten 372–382 (siehe Seiten 31, 33).

[82] Michael O. Rabin. „Transaction protection by beacons“. In: *Journal of Computer and System Sciences* 27.2 (1983), Seiten 256–267 (siehe Seite 31).

[83] Josh Cohen Benaloh und Moti Yung. „Distributing the Power of a Government to Enhance the Privacy of Voters (Extended Abstract)“. In: *Proceedings of the Fifth Annual ACM Symposium on Principles of Distributed Computing, Calgary, Alberta, Canada, August 11-13, 1986*. 1986, Seiten 52–62 (siehe Seiten 31, 33, 34).

[84] Josh D. Cohen. „Improving Privacy in Cryptographic Elections“. In: Apr. 1994 (siehe Seiten 31, 33).

[85] Adi Shamir. „How to Share a Secret“. In: *Commun. ACM* 22.11 (1979), Seiten 612–613 (siehe Seite 31).

[86] Martin Hirt und Kazue Sako. „Efficient Receipt-Free Voting Based on Homomorphic Encryption“. In: *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*. 2000, Seiten 539–556 (siehe Seiten 32–34).

[87] Daniel Ellard und David Alpert. *Receipt-Free Secure Elections*. Technischer Bericht. 2003. URL: https://dash.harvard.edu/handle/1/25619465 (besucht am 07. 11. 2017) (siehe Seite 32).

[88] Kenneth R. Iversen. „A cryptographic scheme for computerized general elections“. In: *Annual International Cryptology Conference*. 1991, Seiten 405–419 (siehe Seite 32).

[89] Kenneth R. Iversen. „A Novel Probabilistic Additive Privacy Homomorphism“. In: *Proceedings of the International Conference on Finite Fields, Coding Theory, and Advances in Communications and Computing, Lecture Notes in Pure and Applied Mathematics*. Herausgegeben von Marcel Dekker. 1993 (siehe Seite 32).

[90] Aggelos Kiayias und Moti Yung. „The Vector-Ballot e-Voting Approach“. In: *Financial Cryptography, 8th International Conference, FC 2004, Key West, FL, USA, February 9-12, 2004. Revised Papers*. 2004, Seiten 72–89 (siehe Seiten 32, 38).

[91] Aggelos Kiayias, Michael Korman und David Walluck. „An Internet Voting System Supporting User Privacy“. In: *22nd Annual Computer Security Applications Conference (ACSAC 2006),11-15 December 2006, Miami Beach, Florida, USA*. 2006, Seiten 165–174 (siehe Seite 33).

[92] Aggelos Kiayias und Moti Yung. „Self-tallying Elections and Perfect Ballot Secrecy“. In: *Public Key Cryptography, 5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002, Paris, France, February 12-14, 2002, Proceedings*. 2002, Seiten 141–158 (siehe Seite 33).

[93] Berry Schoenmakers. „A simple publicly verifiable secret sharing scheme and its application to electronic voting“. In: *Lecture Notes in Computer Science* (1999), Seiten 148–164 (siehe Seite 33).

[94] Ivan Damgård und Mads Jurik. „A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System“. In: *Public Key Cryptography, 4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001, Cheju Island, Korea, February 13-15, 2001, Proceedings*. 2001, Seiten 119–136 (siehe Seiten 33, 34).

[95] Pascal Paillier. „Public-Key Cryptosystems Based on Composite Degree Residuosity Classes“. In: *Advances in Cryptology - EUROCRYPT '99*. Herausgegeben von Jacques Stern. Berlin, Heidelberg: Springer, 1999, Seiten 223–238 (siehe Seite 33).

[96] Olivier Baudron et al. „Practical multi-candidate election system“. In: *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing, PODC 2001, Newport, Rhode Island, USA, August 26-29, 2001*. 2001, Seiten 274–283 (siehe Seite 34).

[97] Ronald Cramer et al. „Multi-Authority Secret-Ballot Elections with Linear Work“. In: *Advances in Cryptology - EUROCRYPT '96: International Conference on the Theory and Application of Cryptographic Techniques Saragossa, Spain, May 12–16, 1996 Proceedings*. Berlin, Heidelberg: Springer, 1996 (siehe Seite 34).

[98] Ronald Cramer, Rosario Gennaro und Berry Schoenmakers. „A Secure and Optimally Efficient Multi-Authority Election Scheme“. In: *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*. 1997, Seiten 103–118 (siehe Seiten 34, 35).

[99] Alessandro Acquisti. „Receipt-Free Homomorphic Elections and Write-in Ballots“. In: *IACR Cryptology ePrint Archive* 2004 (2004), Seite 105 (siehe Seite 34).

[100] Kazue Sako und Joe Kilian. „Secure Voting Using Partially Compatible Homomorphisms“. In: *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*. 1994, Seiten 411–424 (siehe Seite 34).

[101] Uriel Feige und Adi Shamir. „Witness Indistinguishable and Witness Hiding Protocols“. In: *STOC '90 Proceedings of the twenty-second annual ACM symposium on Theory of computing*. ACM New York, NY, USA, 1990 (siehe Seite 34).

[102] David Chaum. „Surevote: technical overview“. In: *Proceedings of the workshop on trustworthy elections (WOTE'01)*. 2001 (siehe Seite 35).

[103] Ed Gerck et al. „The Business of Electronic Voting“. In: *Financial Cryptography, 5th International Conference, FC 2001, Grand Cayman, British West Indies, February 19-22, 2002, Proceedings*. 2001, Seiten 234–259 (siehe Seite 35).

[104] Rui Joaquim und Carlos Ribeiro. „CodeVoting: protecting against malicious vote manipulation at the voter's PC“. In: *Frontiers of Electronic Voting, 29.07. - 03.08.2007*. 2007 (siehe Seite 36).

[105] Rui Joaquim und Carlos Ribeiro. „CodeVoting Protection Against Automatic Vote Manipulation in an Uncontrolled Environment“. In: *E-Voting and Identity, First International Conference, VOTE-ID 2007, Bochum, Germany, October 4-5, 2007, Revised Selected Papers*. 2007, Seiten 178–188 (siehe Seite 36).

[106] Rui Joaquim, Carlos Ribeiro und Paulo Ferreira. „Improving Remote Voting Security with CodeVoting“. In: *Towards Trustworthy Elections, New Directions in Electronic Voting*. 2010, Seiten 310–329 (siehe Seite 36).

[107] Jörg Helbach und Jörg Schwenk. „Secure Internet Voting with Code Sheets“. In: *E-Voting and Identity, First International Conference, VOTE-ID 2007, Bochum, Germany, October 4-5, 2007, Revised Selected Papers*. 2007, Seiten 166–177 (siehe Seite 36).

[108] Jörg Helbach, Jörg Schwenk und Sven Schäge. „Code Voting with Linkable Group Signatures“. In: *3rd International Conference, Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting.CC, August 6th-9th, 2008 in Castle Hofen, Bregenz, Austria*. 2008, Seiten 209–208 (siehe Seite 36).

[109] Rui Joaquim, Carlos Ribeiro und Paulo Ferreira. „VeryVote: A Voter Verifiable Code Voting System“. In: *E-Voting and Identity, Second International Conference, VOTE-ID 2009, Luxembourg, September 7-8, 2009. Proceedings*. 2009, Seiten 106–121 (siehe Seite 36).

[110] C. Andrew Neff. *Practical high certainty intent verification for encrypted votes*. 2004 (siehe Seite 36).

[111] Rui Joaquim, Paulo Ferreira und Carlos Ribeiro. „EVIV: An end-to-end verifiable Internet voting system“. In: *Computers & Security* 32 (2013), Seiten 170–191 (siehe Seite 36).

[112] Miroslaw Kutylowski und Filip Zagórski. „Scratch, Click & Vote: E2E Voting over the Internet“. In: *Towards Trustworthy Elections, New Directions in Electronic Voting*. 2010, Seiten 343–356 (siehe Seite 36).

[113] Stefan Popoveniuc und Benjamin Hosp. „An Introduction to PunchScan“. In: *Towards Trustworthy Elections, New Directions in Electronic Voting*. 2010, Seiten 242–259 (siehe Seite 36).

[114] Filip Zagórski et al. „Remotegrity: Design and Use of an End-to-End Verifiable Remote Voting System“. In: *Applied Cryptography and Network Security: 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings*. Herausgegeben von Michael Jacobson et al. Berlin Heidelberg: Springer, Seiten 441–457 (siehe Seite 36).

[115] David Chaum et al. „Scantegrity: End-to-End Voter-Verifiable Optical- Scan Voting“. In: *IEEE Security & Privacy* 6.3 (2008) (siehe Seite 36).

[116] David Chaum et al. „Scantegrity II: End-to-End Verifiability by Voters of Optical Scan Elections Through Confirmation Codes“. In: *IEEE Transactions on Information Forensics and Security* 4.4 (2009) (siehe Seite 36).

[117] Alan T. Sherman et al. „Scantegrity III: Automatic Trustworthy Receipts, Highlighting over/Under Votes, and Full Voter Verifiability“. In: *Proceedings of the 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections* (siehe Seite 36).

[118] Peter Y. A. Ryan und Vanessa Teague. „Pretty Good Democracy“. In: *Security Protocols XVII, 17th International Workshop, Cambridge, UK, April 1-3, 2009. Revised Selected Papers*. 2009, Seiten 111–130 (siehe Seite 37).

[119] Peter Y. A. Ryan und Steve A. Schneider. „Prêt à Voter with Re-encryption Mixes“. In: *Computer Security - ESORICS 2006, 11th European Symposium on Research in Computer Security, Hamburg, Germany, September 18-20, 2006, Proceedings*. 2006, Seiten 313–326 (siehe Seite 37).

[120] Markus Jakobsson und Ari Juels. „Mix and Match: Secure Function Evaluation via Ciphertexts“. In: *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*. 2000, Seiten 162–177 (siehe Seite 37).

[121] Pei-Yih Ting und Xiao-Wei Huang. „Distributed Paillier Plaintext Equivalence Test“. In: *I. J. Network Security* 6.3 (2008), Seiten 258–264 (siehe Seite 37).

[122] David Chaum, Peter Y. A. Ryan und Steve A. Schneider. „A Practical Voter-Verifiable Election Scheme“. In: *Computer Security - ESORICS 2005, 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005, Proceedings*. 2005, Seiten 118–139 (siehe Seite 37).

[123] Peter YA Ryan. „Prêt à Voter with Paillier encryption“. In: *Mathematical and Computer Modelling* 48.9 (2008), Seiten 1646–1662 (siehe Seite 37).

[124] Jurlind Budurushi et al. „Pretty Understandable Democracy - A Secure and Understandable Internet Voting Scheme“. In: *2013 International Conference on Availability, Reliability and Security, ARES 2013, Regensburg, Germany, September 2-6, 2013*. 2013, Seiten 198–207 (siehe Seite 37).

[125] Stephan Neumann et al. „Pretty Understandable Democracy 2.0“. In: *IACR Cryptology ePrint Archive* 2014 (2014), Seite 625 (siehe Seite 38).

[126] Joerg Abendroth et al. „Privacy-ABC Usage Scenarios“. In: *Attribute-based Credentials for Trust: Identity in the Information Society*. Herausgegeben von Kai Rannenberg, Jan Camenisch und Ahmad Sabouri. Berlin, Heidelberg: Springer, 2014, Seiten 319–343 (siehe Seite 39).

[127] Christian Paquin. *U-Prove Technology Overview V1.1*. Technischer Bericht. Microsoft Corporation, 2013. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/U-Prove20Technology20Overview20V1.120Revision202.pdf (besucht am 27. 02. 2018) (siehe Seite 40).

[128] Jan Camenisch. *Specification of the Identity Mixer Cryptographic Library Version 2.3.0*. Technischer Bericht. Security Team of IBM Research – Zurich, 2010. URL: http://domino.research.ibm.com/library/cyberdig.nsf/papers/EEB54FF3B91C1D648525759B004FBBB1/%5C$File/rz3730_revised.pdf (besucht am 27. 02. 2018) (siehe Seite 40).

[129] Gergely Alpár und Jaap-Henk Hoepman. „A Secure Channel for Attribute-based Credentials: [Short Paper]“. In: *Proceedings of the 2013 ACM Workshop on Digital Identity Management*. 2013, Seiten 13–18 (siehe Seite 40).

[130] Jan Camenisch und Anna Lysyanskaya. „A Signature Scheme with Efficient Protocols“. In: *Security in Communication Networks*. Herausgegeben von Stelvio Cimato, Giuseppe Persiano und Clemente Galdi. Springer, Berlin Heidelberg, 2003, Seiten 268–289 (siehe Seite 40).

[131] Andreas Put et al. *An Anonymous, Verifiable Internet Service POLL system*. Technischer Bericht. KU Leuven, 2014. URL: https://lirias.kuleuven.be/bitstream/123456789/457673/1/CW669.pdf (besucht am 27. 02. 2018) (siehe Seite 40).

[132] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Technischer Bericht. 2008. URL: https://bitcoin.org/bitcoin.pdf (besucht am 26. 11. 2017) (siehe Seite 41).

[133] Gavin Wood. *Ethereum: A Secure Decentralised Generalised Transaction Ledger*. Technischer Bericht. 2014. URL: http://gavwood.com/Paper.pdf (besucht am 27. 11. 2017) (siehe Seite 41).

[134] Philip Boucher. „What if blockchain technology revolutionised voting?“ In: *What if ...?* (Sep. 2016) (siehe Seite 41).

[135] Robert Riemann und Stéphane Grumbach. „Distributed Protocols at the Rescue for Trustworthy Online Voting“. In: (2017) (siehe Seite 41).

[136] Teogenes Moura und Alexandre Gomes. „Blockchain Voting and Its Effects on Election Transparency and Voter Confidence“. In: *Proceedings of the 18th Annual International Conference on Digital Government Research*. Staten Island, NY, USA, 2017 (siehe Seite 41).

[137] Stefan Konst. „Sichere Log-Dateien auf Grundlage kryptographisch verketteter Einträge“. Diplomarbeit. Technische Universität Braunschweig, 2000 (siehe Seite 41).

[138] Christian Meter. „Design of Distributed Voting Systems“. Master's Thesis. Heinrich-Heine-Universität Düsseldorf, 2015 (siehe Seite 41).

[139] Ahmed Ben Ayed. „A Conceptual Secure Blockchain-Based Electronic Voting System“. In: *International Journal of Network Security and its Applications (IJNSA)* 19.2 (Mai 2017) (siehe Seite 42).

[140] Christopher Brake Andrew Barnes and und Thomas Perry. *Digital Voting with the use of Blockchain Technology*. Technischer Bericht. 2016. URL: https://www.economist.com/sites/default/files/plymouth.pdf (besucht am 26. 11. 2017) (siehe Seite 42).

[141] Kibin Lee et al. „Electronic Voting Service Using Block-Chain“. In: *Journal of Digital Forensics, Security and Law* 11.2 (2016) (siehe Seite 43).

[142] Stefano Bistarelli et al. „An End-to-end Voting-system Based on Bitcoin“. In: *SAC '17 Proceedings of the Symposium on Applied Computing* (Apr. 2017) (siehe Seite 43).

[143] L. Zhu, P. Leach und S. Hartman. *Anonymity Support for Kerberos*. RFC 6112. Apr. 2011 (siehe Seite 43).

[144] Yi Liu und Qi Wang. *An E-voting Protocol Based on Blockchain*. Cryptology ePrint Archive, Report 2017/1043. 2017 (siehe Seite 43).

[145] Jason Paul Cruz und Yuichi Kaji. „E-voting System Based on the Bitcoin Protocol and Blind Signatures“. In: *IPSJ Transactions on Mathematical Modeling and Its Applications* 10.1 (März 2017) (siehe Seite 44).

[146] Zhichao Zhao und T.-H. Hubert Chan. „How to Vote Privately Using Bitcoin“. In: *Information and Communications Security*. 2016 (siehe Seite 44).

[147] Eli Ben-Sasson et al. „SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge“. In: *Advances in Cryptology – CRYPTO 2013* (2013) (siehe Seite 44).

[148] G. S. Grewal et al. „Du-Vote: Remote Electronic Voting with Untrusted Computers“. In: *2015 IEEE 28th Computer Security Foundations Symposium*. Juli 2015, Seiten 155–169 (siehe Seiten 45, 73–75, 79, 81, 82, 86–88).

[149] Estonian Information System Authority (RIA). *Possible Security Vulnerability Detected in the Estonian ID-card Chip*. URL: https://www.ria.ee/en/possible-security-vulnerability-detected-in-the-estonian-id-card-chip.html (besucht am 11. 12. 2017) (siehe Seite 52).

[150] National Institute of Standards und Technology (NIST) - National Vulnerability Database (NVD). *CVE-2017-15361 Detail*. URL: https://nvd.nist.gov/vuln/detail/CVE-2017-15361 (besucht am 11. 12. 2017) (siehe Seite 52).

[151] Matus Nemec et al. „CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security“. In: Dallas, TX, USA: ACM, 2017. ISBN: 978-1-4503-4946-8 (siehe Seite 52).

[152] Estonian Information System Authority (RIA). *PID-cards affected by the security risk can be renewed from November*. URL: https://www.ria.ee/en/id-cards-affected-by-the-security-risk-can-be-renewed-from-november.html (besucht am 11. 12. 2017) (siehe Seite 52).

[153] Estonian National Electoral Committee. *E-Voting System, General Overview*. Technischer Bericht. 2010. URL: http://vvk.ee/public/dok/General_Description_E-Voting_2010.pdf (besucht am 21. 11. 2016) (siehe Seiten 53–56).

[154] Drew Springall et al. „Security Analysis of the Estonian Internet Voting System“. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. Scottsdale, Arizona, USA: ACM, 2014, Seiten 703–715. ISBN: 978-1-4503-2957-6 (siehe Seiten 53, 55, 56, 59–62, 64).

[155] Sven Heiberg, Peeter Laud und Jan Willemson. „The Application of I-Voting for Estonian Parliamentary Elections of 2011“. In: *E-Voting and Identity: Third International Conference, VoteID 2011, Tallinn, Estonia, September 28-30, 2011, Revised Selected Papers*. Herausgegeben von Aggelos Kiayias und Helger Lipmaa. Berlin, Heidelberg: Springer, 2012, Seiten 208–223. ISBN: 978-3-642-32747-6 (siehe Seiten 53, 55–58, 61).

[156] S. Heiberg und J. Willemson. „Verifiable internet voting in Estonia“. In: *2014 6th International Conference on Electronic Voting: Verifying the Vote (EVOTE)*. Okt. 2014, Seiten 1–8 (siehe Seiten 55, 59, 60).

[157] N. Leavitt. „Internet Security under Attack: The Undermining of Digital Certificates“. In: *Computer* 44.12 (Dez. 2011), Seiten 17–20. ISSN: 0018-9162 (siehe Seite 57).

[158] S. Peng, S. Yu und A. Yang. „Smartphone Malware and Its Propagation Modeling: A Survey“. In: *IEEE Communications Surveys Tutorials* 16.2 (2014), Seiten 925–941. ISSN: 1553-877X (siehe Seite 60).

[159] Bundesamt für Sicherheit in der Informationstechnik. *Zertifizierungsreport BSI-DSZ-CC-0862-2016 zu POLYAS CORE, Version 2.2.3 der Micromata GmbH*. Technischer Bericht. 2016. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Reporte/Reporte08/0862a_pdf.pdf?__blob=publicationFile&v=2 (besucht am 14. 08. 2017) (siehe Seite 64).

[160] M. M. Olembo, P. Schmidt und M. Volkamer. „Introducing Verifiability in the POLYAS Remote Electronic Voting System“. In: *Sixth International Conference on Availability, Reliability and Security*. 2011, Seiten 127–134 (siehe Seiten 64–71).

[161] Gesellschaft für Informatik / POLYAS GmbH. *GI Vorstands- und Präsidiumswahlen*. URL: https://gi-wahlen.de (besucht am 26. 11. 2017) (siehe Seiten 72, 73).

[162] Jeremy Clark und Urs Hengartner. „On the Use of Financial Data as a Random Beacon.“ In: *EVT/WOTE* 89 (2010) (siehe Seite 75).

[163] Ben Adida et al. „Electing a University President Using Open-audit Voting: Analysis of Real-world Use of Helios“. In: *Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*. EVT/WOTE'09. Montreal, Canada: USENIX Association, 2009, Seiten 10–10 (siehe Seite 81).

[164] Der Bundeswahlleiter. *Wahl-Lexikon - Briefwahl*. URL: https://www.bundeswahlleiter.de/service/glossar/b/briefwahl.html (besucht am 07. 09. 2017) (siehe Seite 89).

[165] Robert Vehrkamp et al. *Einwurf - Zukunft der Demokratie: Zeitgemäß wählen*. Sonderausgabe 1-2. 2016. URL: https://www.bertelsmann-stiftung.de/fileadmin/files/BSt/Publikationen/GrauePublikationen/ZD_EINWURF_Sonderausgabe_1-2_2016.pdf (besucht am 07. 09. 2017) (siehe Seite 89).

[166] Emilie Reichmann. *Einwurf - Zukunft der Demokratie: Mehr Briefwahl wagen!* Ausgabe 3. 2016. URL: https://www.bertelsmann-stiftung.de/fileadmin/files/BSt/Publikationen/GrauePublikationen/ZD_EINWURF_03_2016.pdf (besucht am 07. 09. 2017) (siehe Seiten 89, 93).

[167] Der Bundeswahlleiter. *Bundestagswahl 2017 - Briefwahl*. URL: https://www.bundeswahlleiter.de/bundestagswahlen/2017/informationen-waehler/briefwahl.html (besucht am 31. 08. 2017) (siehe Seite 89).

[168] Organisation für Sicherheit und Zusammenarbeit in Europa (OSZE) - Office for Democratic Institutions and Human Rights (ODIHR) Wahlexpertenteam (Election Expert Team). *Abschlussbericht: Bundesrepublik Deutschland - Wahl zum Deutschen Bundestag am 27. September 2009*. 2009. URL: http://www.osce.org/de/odihr/elections/germany/40879?download=true (besucht am 06. 09. 2017) (siehe Seite 89).

[169] Latanya Sweeney, Ji Su Yoo und Jinyan Zang. „Voter Identity Theft: Submitting Changes to Voter Registrations Online to Disrupt Elections". In: *Technology Science* (6. Sep. 2017). URL: https://techscience.org/a/2017090601 (siehe Seite 91).

[170] Wolfgang Schreiber. *BWahlG: Kommentar zum Bundeswahlgesetz*. 10. Auflage. Carl Heymanns, 2017. ISBN: 978-3-452-28738-0 (siehe Seite 91).

[171] Lina Timm. „Briefwahl-Skandal: Bleiben Tausende Wähler ohne Stimme?" In: *Focus* (21. Sep. 2013). URL: http://www.focus.de/politik/deutschland/bundestagswahl-2013/tid-33649/stimmzettel-verschwunden-briefwahl-skandal-bleiben-tausende-waehler-ohne-stimme_aid_1107824.html (besucht am 07. 09. 2017) (siehe Seiten 91, 94).

[172] Jan Thomsen. „64.000 Wahlscheine beantragt: Verfassungsrechtler kritisieren Briefwahl". In: *Berliner Zeitung* (18. Aug. 2013). URL: http://www.berliner-zeitung.de/4464084 (besucht am 05. 09. 2017) (siehe Seiten 91, 94).

[173] Raphael Moritz. „Pannen bei der Briefwahl". In: *Handelsblatt* (9. Okt. 2013). URL: http://www.handelsblatt.com/8908552.html (besucht am 05. 09. 2017) (siehe Seiten 91, 94).

[174] Jörg Thoma. „Wahlbetrug leicht gemacht". In: *Golem.de* (5. Juli 2013). URL: https://www.golem.de/news/hacking-wahlbetrug-leicht-gemacht-1307-100234.html (besucht am 07. 09. 2017) (siehe Seite 92).

[175] *Erste Verordnung zur Änderung der Bundeswahlordnung*. Nov. 1989 (siehe Seite 92).

[176] William Clarkson et al. „Fingerprinting Blank Paper Using Commodity Scanners". In: *Proceedings of the IEEE Symposium on Security and Privacy*. Mai 2009 (siehe Seite 92).

[177] Landeshauptstadt München - Kreisverwaltungsreferat. *Antrag für einen Wahlschein mit Briefwahlunterlagen für die Bundestagswahl am 24. September 2017 für wahlberechtigte Münchnerinnen und Münchner*. URL: https://briefwahl-muenchen.de (besucht am 01. 08. 2017) (siehe Seite 93).

[178] Fabian Wahl. „Tausende Briefe und Pakete kommen nie an". In: *Die Welt* (22. Aug. 2011). URL: https://www.welt.de/wirtschaft/article13558939/Tausende-Briefe-und-Pakete-kommen-nie-an.html (besucht am 27. 02. 2018) (siehe Seite 94).

[179] Thomas Kutschbach. „Ärger mit der Post: Briefe, die nie ankommen". In: *Berliner Zeitung* (5. März 2014). URL: https://www.berliner-zeitung.de/panorama/aerger-mit-der-post-briefe--die-nie-ankommen-3219438 (besucht am 27. 02. 2018) (siehe Seite 94).

[180] Ronghui Gu et al. „CertiKOS: An Extensible Architecture for Building Certified Concurrent OS Kernels". In: *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*. OSDI'16. Savannah, GA, USA: USENIX Association, 2016, Seiten 653–669 (siehe Seite 96).

[181] Mihir Bellare und Phillip Rogaway. „Optimal asymmetric encryption". In: *Advances in Cryptology — EUROCRYPT'94: Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May 9–12, 1994 Proceedings*. Herausgegeben von Alfredo De Santis. Berlin, Heidelberg: Springer, 1995, Seiten 92–111. ISBN: 978-3-540-44717-7 (siehe Seite 111).

[182] Uwe Schöning. *Kryptologie-Kompendium*. Berlin: Lehmanns Media, 2013, 136 Seiten. ISBN: 978-3-86541-515-8 (siehe Seiten 112, 114).

[183] C. P. Schnorr. „Efficient Identification and Signatures for Smart Cards“. In: *Advances in Cryptology — CRYPTO' 89 Proceedings*. Herausgegeben von Gilles Brassard. New York, NY: Springer New York, 1990, Seiten 239–252. ISBN: 978-0-387-34805-6 (siehe Seiten 112, 113).

[184] David Chaum und Torben Pryds Pedersen. „Wallet Databases with Observers“. In: *Advances in Cryptology — CRYPTO' 92: 12th Annual International Cryptology Conference Santa Barbara, California, USA August 16–20, 1992 Proceedings*. Herausgegeben von Ernest F. Brickell. Berlin, Heidelberg: Springer, 1993, Seiten 89–105. ISBN: 978-3-540-48071-6 (siehe Seiten 112, 113).

[185] Amos Fiat und Adi Shamir. „How To Prove Yourself: Practical Solutions to Identification and Signature Problems“. In: *Advances in Cryptology — CRYPTO' 86: Proceedings*. Herausgegeben von Andrew M. Odlyzko. Berlin, Heidelberg: Springer, 1987, Seiten 186–194. ISBN: 978-3-540-47721-1 (siehe Seite 113).

[186] Felix Brandt. „Efficient Cryptographic Protocol Design Based on Distributed El Gamal Encryption“. In: *Information Security and Cryptology - ICISC 2005: 8th International Conference, Seoul, Korea, December 1-2, 2005, Revised Selected Papers*. Herausgegeben von Dong Ho Won und Seungjoo Kim. Berlin, Heidelberg: Springer, 2006, Seiten 32–47. ISBN: 978-3-540-33355-5 (siehe Seite 115).

A ANHANG

A.1 KRYPTOGRAFISCHE GRUNDLAGEN

In diesem Anhang werden für interessierte Leser zum besseren Verständnis die kryptografischen Grundlagen vorgestellt, welche für einige der Wahlverfahren benötigt werden.

A.1.1 RSA-OAEP

OAEP steht für Optimal Asymmetric Encryption Padding [181]. Da RSA ein deterministisches Verschlüsselungsverfahren ist, ist das Ziel des Paddings, das Chiffrat zu randomisieren. Sei k_0 ein Sicherheitsparameter, welcher so dimensioniert ist, dass ein Angreifer unmöglich 2^{k_0} Rechenschritte ausführen kann. Seien G und H kryptografische Hashfunktionen, sodass G einen Output von n Bits produziert, und H eine Hashfunktion, welche einen Output von k_0 Bits erzeugt, wobei n die Größe des RSA-Schlüssels ist. Des Weiteren sei m die zu verschlüsselnde Nachricht. Zur **Verschlüsselung** wird eine Zufallszahl $r \in_R \{1, 2, \ldots 2^{k_0}\}$ gewählt und

$$X = (m \oplus G(r)) \text{ und } Y = (r \oplus H(m \oplus G(r))$$

berechnet. Anschließend wird $c = RSA_{enc}(X||Y)$ berechnet. Zur **Entschlüsselung** wird nun $X||Y = RSA_{dec}(c)$ berechnet, sodass aus X und Y wieder m rekonstruiert werden kann, indem

$$r = (Y \oplus H(X)) \text{ und } m = (X \oplus G(r))$$

berechnet wird. Jetzt ist es noch wichtig zu zeigen, dass dieses Verfahren auch funktioniert. Dies gilt, da:

$$\begin{aligned} Y \oplus H(X) =& r \oplus H(m \oplus G(r)) \oplus H(X) \\ =& r \oplus H(m \oplus G(r)) \oplus H((m \oplus G(r)) \\ =& r \end{aligned}$$

und nachdem r gegeben ist, gilt für den zweiten Term:

$$\begin{aligned} X \oplus G(r) =& m \oplus G(r) \oplus G(r) \\ =& m \end{aligned}$$

A.1.2 Zero-Knowledge-Proofs

Zero-Knowledge-Beweise werden von einem Beweiser P dazu benutzt, um einem Verifizierer V die Gültigkeit einer Aussage A zu beweisen, ohne dass V etwas über A lernt. Zero-Knowledge-Beweise laufen interaktiv als sogenanntes (Sigma)$\sum$-Protokoll ab. Das bedeutet, ein Zero-Knowledge-Protokoll besteht aus den drei Phasen Commitment, Challenge und Response. Während der Commitment-Phase legt sich P gegenüber V auf einen bestimmten Wert fest, welcher nachträglich nicht mehr geändert werden kann. In der anschließenden Challenge-Phase stellt V an P eine Aufgabe, welche ohne Kenntnis von A schwer zu lösen ist. In der Response-Phase überzeugt sich V davon, dass P die gegebene Aufgabe richtig gelöst hat. In manchen Protokollen wird dieses Vorgehen mehrfach wiederholt, bis V sich

mit nahezu hundertprozentiger Wahrscheinlichkeit sicher sein kann, dass der Beweis korrekt ist [16, 182].

Schnorrs Zero-Knowledge-Protokoll

Das Zero-Knowledge-Protokoll von Schnorr [183] dient dazu, die Kenntnis eines diskreten Logarithmus zu beweisen, ohne diesen preiszugeben. Bei Internetwahlverfahren wird das Protokoll zusammen mit der ElGamal-Verschlüsselung oft dazu verwendet, um die Kenntnis über den privaten Schlüssel zu beweisen. Sei dazu p eine Primzahl und $g \in \mathbb{Z}_p$ ein Generator, angenommen der Beweiser P kennt $y = g^x$ und will dem Verifizierer V zeigen, dass er das x kennt. Dies können P und V mit dem im Folgenden beschriebenen Σ-Protokoll realisieren.

1. P wählt $a \in_R \mathbb{Z}_p$ ung sendet $i = g^a$ an V (Commitment),
2. V wählt eine Zufallszahl $r \in_R \mathbb{Z}_p$ und sendet diese an P (Challenge),
3. P berechnet $s = rx + k$ und sendet dies an V (Response),
4. V kann nun prüfen, ob $i = g^s \cdot y^{-r}$ ist. Falls ja, kann V mit hoher Wahrscheinlichkeit davon ausgehen, dass V das x zum y kennt.

Siehe dazu auch Abbildung 27.

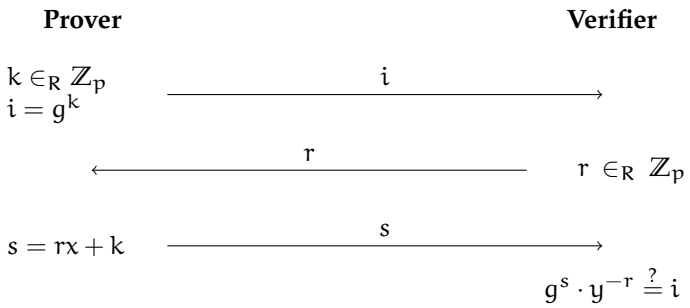


Abbildung 27: Schnorrs Zero-Knowledge-Protokoll in Anlehnung an [183].

Chaum-Pedersen-Protokoll

Das Chaum-Pedersen-Protokoll [184] beweist Logarithmengleichheit. Das bedeutet der Beweiser P kennt $y = g^x$ sowie $z = m^x$ und will dem Verifizierer P beweisen, dass $\log_g(y) = \log_m(z)$ ist. Bei Internetwahlverfahren wird dies im Zusammenhang mit der ElGamal-Verschlüsselung oft dazu verwendet, um die Kenntnis eines Klartextes zu beweisen. Sei dazu p eine Primzahl und $g \in \mathbb{Z}_p$ ein Generator, dann läuft das entsprechende $\sum$-Protokoll folgendermaßen ab (siehe Abbildung 28):

1. P wählt $a \in_R \mathbb{Z}_p$ und sendet $i_1 = g^a$ und $i_2 = m^a$ an V (Commitment),
2. V wählt eine Zufallszahl $r \in_R \mathbb{Z}_p$ und sendet diese an P (Challenge),
3. P berechnet $s = rx + k$ und sendet dies an V (Response),
4. V kann nun prüfen, ob $g^s = i_1 y^r$ und $m^s = i_2 z^r$ gilt. Fall ja, kann V mit hoher Wahrscheinlichkeit davon ausgehen, dass $\log_g(y) = \log_m(z)$ gilt und dass V das entsprechende x zu y und z kennt.

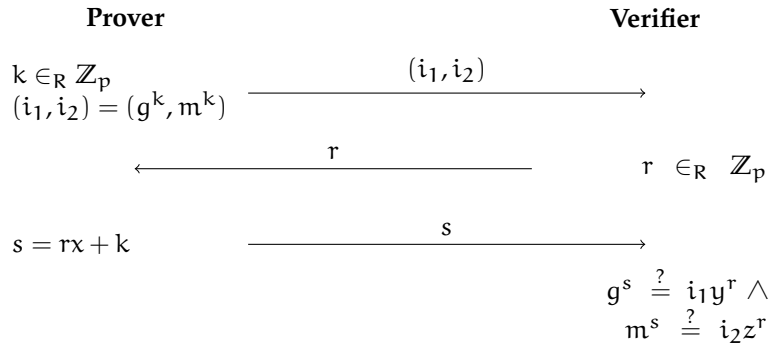


Abbildung 28: Chaum-Pedersen-Protokoll in Anlehnung an [184].

Fiat-Shamir-Heuristik

Bei Internetwahlsystemen ist es gewünscht, dass Zero-Knowledge-Beweise keine Interaktion zwischen dem Beweiser P und dem Verifizierer V voraussetzen. Dies ist besonders dann wichtig, wenn ein Beweis von jedem nachvollzogen werden können soll. Die Fiat-Shamir-Heuristik stellt dazu ein Vorgehen bereit, welches jedes beliebige interaktive Zero-Knowledge-Protokoll in ein nicht-interaktives Zero-Knowledge-Protokoll transformiert. Die Idee dahinter ist es, die Challenge nicht von V wählen zu lassen, sondern durch ein Random Oracle[1] zu erzeugen. Allerdings ist bewiesen, dass es keine Random Oracles gibt, weswegen in der Praxis stattdessen kryptografische Hashfunktionen verwendet werden. In Schnorrs Zero-Knowledge-Protokoll kann dies beispielsweise bewerkstelligt werden, indem $r = h(i)$ gewählt wird. Die Überprüfung erfolgt, indem kontrolliert wird, ob $h(g^s \cdot y^{-r}) = r$ gilt. Dies funktioniert, da:

$$\begin{aligned} r = h(i) = h(g^k) &\stackrel{?}{=} h(g^s \cdot y^{-r}) \\ &\stackrel{?}{=} h(g^{rx+k} \cdot g^{-rx}) \\ &\stackrel{?}{=} h(g^{rx+k-rx}) \\ &\stackrel{?}{=} h(g^k) \end{aligned}$$

Vorausgesetzt die Hashfunktion ist kryptografisch sicher, kann das nur dann der Fall sein, wenn P das x kennt [16, 183, 185].

A.1.3 ElGamal-Verschlüsselung

Das ElGamal-Kryptosystem ist eines der ersten Public-Key-Kryptosysteme. Es ist ein nicht-deterministisches Verfahren, was bedeutet, dass die gleiche Nachricht auf unterschiedliche Chiffrate abgebildet werden kann. Das ElGamal-Verfahren basiert dabei auf dem Problem des diskreten Logarithmus, welches besagt, dass über einer zyklischen Gruppe G für ein Element $c = a^x \in G$ der Wert x schwer zu berechnen ist. Dazu sei p eine „sichere Primzahl" d. h. $p = 2q + 1$, wobei q ebenfalls eine Primzahl ist[2] und sei g ein Erzeuger der Gruppe $\mathbb{Z}_p^*$. Der geheime Schlüssel x wird zufällig gewählt, sodass $x \in_R \{2, 3, \ldots, p-1\}$. Der öffentliche Schlüssel ist $y = g^x$. Da alle Berechnungen über der Gruppe $\mathbb{Z}_p^*$ durchgeführt werden, werden sie modulo p ausgeführt. In der vorliegenden Arbeit wird aus Gründen der Übersichtlichkeit jedoch darauf verzichtet, da dies aus dem Kontext hervorgehen sollte. Zur Verschlüsselung der Nachricht m wird nun eine Zufallszahl $r \in_R \{2, 3, \ldots p-1\}$ gewählt und mittels des

1 Ein Random Oracle ist eine Funktion, die eine Eingabe x auf einen echt zufälligen Wert abbildet, wobei gleiche Eingaben auf den gleichen Zufall abgebildet werden.

2 Die Eigenschaft wird benötigt, um effizient einen Generator g von $\mathbb{Z}_p^*$ finden zu können.

öffentlichen Schlüssels y die Berechnungen $B = y^r m$ und $A = g^r$ durchgeführt. Das Chiffrat besteht dann aus dem Tupel $c = (A, B)$. Zur Entschlüsselung kann nun mittels des privaten Schlüssels x aus dem Tupel (A, B) wieder m hergestellt werden, indem $(A^x)^{-1}B$ berechnet wird. Dies ist möglich da,

$$(A^x)^{-1}B = ((g^r)^x)^{-1}(g^x)^r m = g^{-rx}g^{rx}m = g^{-rx+rx} = g^0 m = m$$

Es sollte auffallen, dass durch Kenntnis von r das Chiffrat c ebenfalls entschlüsselt werden kann, da

$$(y^r)^{-1}B = ((g^x)^r)^{-1}(g^x)^r m = g^{-rx}g^{xr}m = m$$

gilt.

HOMOMORPHIE Eine weitere nützliche Eigenschaft des ElGamal-Verfahrens ist die Homomorphie-Eigenschaft bezüglich der komponentenweisen Multiplikation. Das bedeutet, dass die Multiplikation zweier Chiffrate c_1, c_2 der Multiplikation der entsprechenden Klartexte m_1, m_2 entspricht, welche mit dem Zuffall $r_1 + r_2$ verschlüsselt sind. Dass das ElGamal-Verfahren diese Eigenschaft erfüllt, ist leicht einzusehen. Sei $c_1 = (g^{r_1}, y^{r_1}m_1)$ und $c_2 = (g^{r_2}, y^{r_2}m_2)$ dann entspricht die Multiplikation von c_1 und c_2:

$$c_1c_2 = (g^{r_1}g^{r_2}, y^{r_1}m_1y^{r_2}m_2) = (g^{r_1+r_2}, y^{r_1+r_2}m_1m_2)$$

Bei der Entschlüsselung erhält man [16, 182]:

$$\begin{aligned}(g^{r_1+r_2})^{-x}(g^x)^{r_1+r_2}m_1m_2 &= g^{-x(r_1+r_2)}g^{x(r_1+r_2)}m_1m_2\\ &= g^{-(xr_1+xr_2)}g^{xr_1+xr_2}m_1m_2\\ &= g^{xr_1+xr_2-(xr_1+xr_2)}m_1m_2\\ &= m_1m_2\end{aligned}$$

Exponentielles ElGamal

Im letzten Abschnitt wurde die Homomorphie-Eigenschaft von ElGamal bezüglich der Multiplikation gezeigt. Allerdings wird für das Auszählen von Ergebnissen bei Wahlen ein Verschlüsselungsverfahren benötigt, welches homomorph bezüglich der Addition ist. Auch ein solches Verfahren lässt sich mit einer kleinen Veränderung des ElGamal-Verschlüsselungsverfahrens realisieren, indem nicht die Nachricht m, sondern g^m verschlüsselt wird. Das bedeutet, dass $B = y^r g^m$ und $A = g^r$ ist. Folglich entsteht das Chiffrat

$$c = (A, B) = (g^r, y^g g^m).$$

Für die Entschlüsselung berechnet man jetzt:

$$(A^x)^{-1}B = g^{-rx}g^{rx}g^m = g^m$$

Folglich lautet der Klartext nun $m = \log_g(g^m)$. Dies ist prinzipiell schwer zu berechnen, allerdings ist eine solche Berechnung möglich, wenn die mögliche Klartextmenge gering ist. Bleibt noch zu zeigen, dass exponentielles ElGamal die Homomorphie-Eigenschaft bezüglich der Addition erfüllt. Seien dazu $c_1 = (g^{r_1}, y^{r_1}g^{m_1})$ und $c_2 = (g^{r_2}, y^{r_2}g^{m_2})$, dann gilt:

$$c_1c_2 = (g^{r_1}g^{r_2}, y^{r_1+r_2}g^{m_1+m_2})$$

Offensichtlich gilt bei der Entschlüsselung [16]:

$$\begin{aligned}\log_g\left((g^{r_1+r_2})^{-x}(g^x)^{r_1+r_2}g^{m_1+m_2}\right) &= \log_g\left(g^{-x(r_1+r_2)}g^{x(r_1+r_2)}g^{m_1+m_2}\right)\\ &= \log_g\left(g^{m_1+m_2}\right)\\ &= m_1 + m_2.\end{aligned}$$

Verteiltes ElGamal

Im folgenden Abschnitt soll ein verteiltes Verschlüsselungsverfahren auf Basis von ElGamal vorgestellt werden. Sei dazu p eine sichere Primzahl und g ein Erzeuger der Gruppe $\mathbb{Z}_p^*$. Seien $p_1, p_2 \ldots, p_n$ die Teilnehmer an dem Protokoll [186].

VERTEILTE SCHLÜSSELGENERIERUNG Zur Schlüsselgenerierung wählt jeder Teilnehmer $i \in 1, 2, \ldots, n$ einen zufälligen Wert $x_i \in_R \{2, 3, \ldots, p-1\}$. Das x_i stellt nun den privaten Schlüssel des Teilnehmers p_i dar. Jeder der Teilnehmer p_i kann nun aus seinem privaten Schlüssel seinen persönlichen öffentlichen Schlüssel $y_i = g^{x_i}$ berechnen und ihn veröffentlichen. Der öffentliche Schlüssel ist $y = \prod_{i=1}^{n} y_i$, welcher von jedem unter der Verwendung von y_i berechnet werden kann. Um zu verhindern, dass einer der Teilnehmer einen Wert für y_i wählt, ohne den Wert von x_i zu kennen, ist jeder Teilnehmer dazu verpflichtet, einen Zero-Knowledge-Proof über die Kenntnis des diskreten Logarithmus von y_i zur Basis g zu erstellen. Dies kann mit Schnorrs Protokoll (siehe Unterabschnitt A.1.2) durchgeführt werden [186].

VERTEILTE ENTSCHLÜSSELUNG Offensichtlich funktioniert die Verschlüsselung analog zu ElGamal. Folglich erhalten die Teilnehmer eine verschlüsselte Nachricht $c = (A, B) = (g^r, y^r m)$ wobei r der beim Verschlüsseln gewählte Zufall ist. Dann lässt sich c entschlüsseln, indem jeder Teilnehmer i, $A_i = A^{x_i}$ veröffentlicht. Jeder kann nun den Klartext m berechnen, indem er

$$\begin{aligned}\left(\prod_{i=1}^{n} A_i\right)^{-1} B &= \left(\prod_{i=1}^{n} g^{r x_i}\right)^{-1} \left(\prod_{i=1}^{n} g^{x_i}\right)^{r} m \\ &= \prod_{i=1}^{n} g^{-r x_i} \prod_{i=1}^{n} g^{x_i r} m \\ &= \prod_{i=1}^{n} g^{-r x_i + x_i r} m \\ &= m\end{aligned}$$

berechnet. Um sicherzustellen, dass keiner der Teilnehmer ein falsches A_i veröffentlicht, führt jeder Teilnehmer i einen Zero-Knowledge-Proof durch, welcher zeigt, dass $y_i^r = g^{x_i r}$ gleich ist wie $A_i = g^{r x_i}$. Dies kann mit Hilfe des Chaum-Pedersen-Protokolls (siehe Abbildung A.1.2) erfolgen [186].

A.1.4 Verifizierbares Mischen

Beim verifizierbaren Mischen (Verifiable Shuffle) geht es darum, die Herkunft einer Nachricht zu verschleiern. Die Idee dazu wird von Chaum [54] beschrieben. Seither gibt es viele weitere Verfahren, welche oft im Zusammenhang mit elektronischen Wahlen in Verbindung stehen. Einige Beispiele sind die Verfahren von Jakobsson [62] und Abe [61]. Um verifizierbares Mischen durchzuführen, kommen sogenannte Mix-Nets (manchmal auch nur „Mix“ genannt) zum Einsatz. Ein solches Mix-Net besteht dabei aus einem oder mehreren Servern. Jeder Server nimmt mehrere Nachrichten entgegen und gibt diese in zufälliger oder pseudozufälliger permutierter Reihenfolge wieder aus. Dabei ist es nicht möglich, eine Eingabenachricht seiner entsprechenden Ausgabenachricht zuzuordnen. Um den Vorgang verifizierbar zu gestalten, gilt es nun folgende Dinge zu beweisen:

- jeder einzelne Mix wurde korrekt ausgeführt,
- keine Nachricht wurde durch eine andere neue Nachricht ausgetauscht sowie
- keine Nachricht wurde gelöscht und dafür eine andere Nachricht mehrmals verwendet.

Dazu gibt es mehrere Techniken welche im Folgenden beschrieben werden [16].

Randomized Partial Checking (RPC)

Randomized Partial Checking (RPC) ist eine sehr effiziente Methode, um die Korrektheit eines Re-Encryption-Mix zu zeigen. Der Grund dafür ist, dass diese Methode keinen Gebrauch von Zero-Knowledge-Beweisen macht. Die Idee dabei ist sehr simpel. Angenommen der Mix-Server erhält als Eingabe eine Liste von n Chiffraten $C = \{c_1, \dots c_n\}$ und berechnet als Ausgabe eine Liste $C' = \{c'_1, \dots c'_n\}$. Dazu führt der Mixed-Server folgende Schritte aus:

1. Er generiert zwei Zufallspermutationen π_l und π_r.
2. Der Server berechnet $c''_{\pi_l(i)} = \text{enc}(c_i)$ für alle $i \in \{1 \dots n\}$, wobei $\text{enc}(c_i)$ eine Re-Encryption darstellt. Daraus wird anschließend der sogenannte „Shadow Mix" $C'' = \{c''_1, \dots, c''_n\}$ generiert.
3. Aus C'' wird $c'_{\pi_r(i)} = \text{enc}(c''_i)$ für alle $i \in \{1 \dots n\}$ berechnet und die Werte in der Ausgabeliste C' gespeichert. Danach werden C' und C'' veröffentlicht.
4. Um die Korrektheit zu zeigen, bekommt der Server nun eine Zufallsbitfolge $\{b_1, \dots, b_n\}$ als Challenge vorgegeben. Die Challenge kann z. B. durch die Fiat-Shamir-Heuristik generiert werden.
5. Der Mixed-Server zeigt die Korrektheit, indem er für alle $b_i \in \{b_1, \dots, b_n\}$
   - $\pi_l^{-1}(i)$ offenlegt und zeigt, dass $c''_i = \text{enc}(c_{\pi_l^{-1}(i)})$, falls $b_i = 0$. Oder
   - $\pi_r(i)$ offenlegt und zeigt, dass $c'_i = \text{enc}(c''_{\pi_r(i)})$, falls $b_i = 1$.

Der Vorgang wird in Abbildung 29 verdeutlicht.

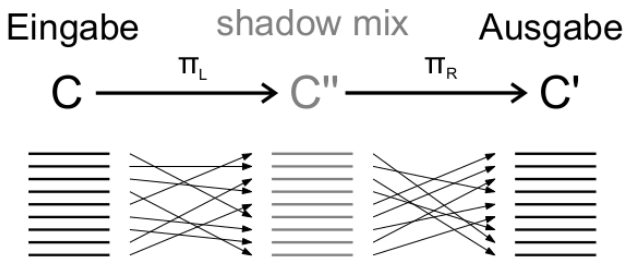


Abbildung 29: Schematische Darstellung RPC [16].

RPC hat den Vorteil, dass es sehr effizient ist, allerdings ist die hier beschriebene Variante bei einer kleinen Eingabeliste unsicher. Da entweder π_l oder π_r aufgedeckt wird, kann ungefähr die Hälfte[3] aller Zuordnungen zwischen Ein- und Ausgabe ausgeschlossen werden. Dieses Problem lässt sich durch mehrfache Anwendung des Mix-Verfahrens eindämmen, da bei jeder Ausführung statistisch gesehen mehr Eingaben für eine feste Ausgabe in Frage kommen [16].

Re-Encryption-Mix mit Shadow-Mixes

Eine weitere Möglichkeit, das korrekte Mischen zu beweisen, ist ein Zero-Knowledge-Beweis. Dieses Verfahren ist zwar wesentlich weniger effizient als RPC, dafür lässt sich der Deanonymisierungsangriff ausschließen. Angenommen der Mix-Server erhält als Eingabe eine Liste von n Chiffraten $C = \{c_1, \dots c_n\}$ und berechnet als Ausgabe eine Liste $C' = \{c'_1, \dots c'_n\}$. Dazu geht der Server wie folgt vor (siehe dazu auch Abbildung 30):

1. Er generiert eine Permutation π und k Permutationspaare $(\pi_{l,j}, \pi_{r,j})$ sodass $\pi = \pi_{l,j} \circ \pi_{r,j}$.

3 Abhängig von der Anzahl an Nullen bzw. Einsen in der Challenge.

2. Dann berechnet er für jedes der k Permutationspaare einen Shadow-Mix wie in Unterabschnitt A.1.4 Schritt 2 beschrieben. Dadurch entstehen k Shadow Mixes $C''_1, \ldots, C''_n$.

3. Für jedes C''_i wird wie in Unterabschnitt A.1.4 Schritt 3 das C' berechnet. Hier kann die Re-Encryption allerdings nur für C''_1 zufällig gewählt werden. Für C''_2 bis C_k berechnet sie sich aus dem entsprechenden Wert für C''_1 und dem Wert für die Re-Encryption aus der vorherigen Re-Encryption. Dies ist nötig, damit alle Shadow-Mixes zur selben Ausgabe führen. Anschließend werden C' sowie die Shadow-Mixes $C''_1, \ldots, C''_n$ veröffentlicht.

4. Um die Korrektheit zu zeigen, bekommt der Server nun eine Zufallsbitfolge $\{b_1, \ldots, b_n\}$ als Challenge vorgegeben. Die Challenge kann z. B. durch die Fiat-Shamir-Heuristik generiert werden.

5. Der Mixed-Server zeigt die Korrektheit, indem er für alle $b_i \in \{b_1, \ldots, b_n\}$ folgendes ausführt:
   - π_l offenlegt und zeigt, dass $c''_{\pi_{L,j}(i),j} = \text{enc}(c_{i,j})$ (für $0 < i \leqslant k$), falls $b_i = 0$.
   - π_r offenlegt und zeigt, dass $c'_{\pi_{r,j}(i),j} = \text{enc}(c''_{i,j})$ (für $0 < i \leqslant k$), falls $b_i = 0$

Der Mixed-Server beweist das korrekte Mischen mittels Zero-Knowledge-Proof. Wichtig ist dabei, dass die Permutation π geheimgehalten werden muss. Um das Ganze etwas effizienter Umzusetzen wie oben beschrieben, wird in der regel erst mitels der Permutation π die Ausgabe erstellt und erst für den Beweis die Shadowmixes generiert [16].

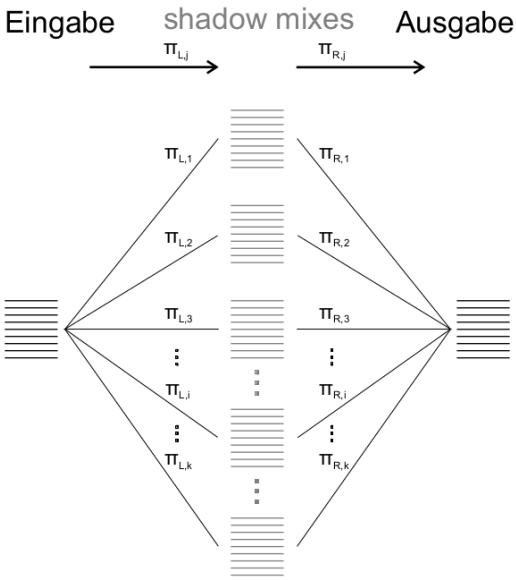


Abbildung 30: Schematische Darstellung Re-Encryption-Mix [16].

A.2 UNTERSTÜTZENDE UNTERLAGEN ZUR BEWERTUNG

A.2.1 Zusammenfassung Bewertungskriterien

Individuelle Verifizierbarkeit

IV.1 **Innere individuelle Verifizierbarkeit:** Der Wähler kann überprüfen, dass seine Stimme auf dem Bulletin Board veröffentlicht wurde und dass diese Stimme den korrekten Kandidaten enthält.

IV.2 **Äußere individuelle Verifizierbarkeit:** Der Wähler kann überprüfen, ob seine Stimme auf dem Bulletin Board veröffentlicht wurde, er kann allerdings nicht verifizieren, ob die Stimme den korrekten Kandidaten enthält.

Es sollte klar sein, dass die Eigenschaft IV.1 als höherwertig anzusehen ist, als die Eigenschaft IV.2. Erfüllt ein Wahlprotokoll die Eigenschaft IV.1, kann der Wähler sicherstellen, dass seine Stimme nicht gelöscht oder geändert wurde. Das bedeutet, dass die Integrität der Stimme sichergestellt werden kann. Wird von einem Protokoll nur die Eigenschaft IV.2 erfüllt, so kann der Wähler zwar überprüfen, ob seine Stimme auf dem Bulletin Board veröffentlicht wurde, er kann allerdings nicht verifizieren, ob die Stimme geändert wurde.

IV.x.1 **Individuelle Verifizierbarkeit nach der Auszählung:** Der Wähler kann überprüfen, ob seine Stimme korrekt in das Ergebnis eingeflossen ist.

IV.x.2 **Individuelle Verifizierbarkeit vor der Auszählung:** Der Wähler kann überprüfen, ob seine Stimme korrekt beim Bulletin Board bzw. Urnenserver abgegeben wurde.

Universelle Verifizierbarkeit

WV.1 **Bedingungslose Wahlberechtigungs-Verifizierbarkeit:** Jeder kann verifizieren (ohne einer in den Prozess involvierten Partei zu vertrauen), dass ausschließlich berechtigte Wähler ihre Stimme abgegeben haben.

WV.2 **Bedingte Wahlberechtigungs-Verifizierbarkeit:** Jeder kann verifizieren, dass ausschließlich berechtigte Wähler Stimmen abgegeben haben. Dazu ist es nötig, dass der Verifizierer bestimmten Parteien vertraut, welche in den Authentifizierungsprozess eingebunden sind.

EV.1 **Bedingungslose Einmaligkeits-Verifizierbarkeit:** Jeder kann verifizieren (ohne einer in den Prozess involvierten Partei zu vertrauen), dass alle Wähler ausschließlich eine Stimme abgegeben haben.

EV.2 **Bedingte Einmaligkeits-Verifizierbarkeit:** Jeder kann verifizieren, dass alle Wähler ausschließlich eine Stimme abgegeben haben. Dazu ist es nötig, dass der Verifizierer bestimmten Parteien vertraut, welche in den Authentifizierungs- und Wahlprozess eingebunden sind.

KV.1 **Kontinuierliche Korrektheits-Verifizierbarkeit:** Jeder kann verifizieren, dass während des kompletten Auszählungsprozesses keine Fehler gemacht wurden.[4]

KV.2 **Diskrete Korrektheits-Verifizierbarkeit:** Jeder kann verifizieren, dass während eines Teils des Auszählungsprozesses keine Fehler gemacht wurden.[5]

Es gilt, dass WV.1, EV.1 und KV.1 die jeweils dazugehörigen Eigenschaften WV.2, EV.2 und KV.2 implizieren. Folglich ist WV.1 höher anzusehen als WV.2, EV.1 höher anzusehen als EV.2 und KV.1 höher anzusehen als KV.2.

4 Es wird sichergestellt, dass während der gesamten Auszählungsphase keine Stimme geändert, gelöscht oder vervielfältigt wird.
5 Es wird sichergestellt, dass während einem Teil der Auszählungsphase keine Stimme geändert, gelöscht oder vervielfältigt wird.

Wahlgeheimnis und Quittungsfreiheit

UL.1 **Unverknüpfbarkeit zwischen Wähleridentität und Wahlentscheidung:** Es ist (dem Angreifer) nicht möglich, eine Verbindung zwischen Wähleridentität und Wahlentscheidung herzustellen.

UL.2 **Unbeweisbarkeit der Verknüpfung zwischen Wähleridentität und Wahlentscheidung:** Es ist (dem Angreifer) möglich, eine Verbindung zwischen Wähleridentität und Wahlentscheidung herzustellen. Diese Verbindung kann (der Angreifer) gegenüber Dritten jedoch nicht beweisen.

QF.1 **Quittungsfreiheit:** Der Wähler kann gegenüber dem Angreifer (Erpresser) eine falsche Wahlentscheidung vorgeben, ohne dass der Angreifer feststellen kann, ob der Wähler gelogen oder die Wahrheit gesagt.

QF.2 **Unmöglichkeit von Stimmenkauf:** Es ist (dem Angreifer) selbst mit der Hilfe des Wählers nicht möglich, eine verifizierbare Verbindung zwischen Wähleridentität und Wahlentscheidung herzustellen.

Mit verifizierbarer Verbindung ist gemeint, dass der Angreifer die Informationen, welche er vom Wähler erhält, überprüfen kann, sodass er sicher sein kann, dass der Wähler nicht gelogen hat. Offensichtlich gilt QF.1 impliziert QF.2 und UL.1 impliziert UL.2. Für die Bewertung wird die Quittungsfreiheit als stärkere Definition des Wahlgeheimnisses betrachtet. Der Grund dafür ist, dass die Eigenschaft der Quittungsfreiheit das Wahlgeheimniss impliziert, da der Angreifer bei der Quittungsfreiheit die Informationen des Wählers in seinen Angriff einbeziehen kann. Zusammengefasst ergibt sich also folgende Reihenfolge: QF.1 impliziert QF.2, QF.2 impliziert UL.1, und UL.1 impliziert UL.2.

Nicht-Erpressbarkeit

AA.1 Es existiert eine Strategie zur Abwehr von Abwesenheitsangriffen, sodass der Angreifer nicht unterscheiden kann, ob der Wähler seinen Anweisungen gefolgt ist.

AA.2 Der Angreifer kann anhand der Informationen, die das System bereitstellt, nicht entscheiden, ob der Wähler gewählt hat.

RA.1 Es existiert eine Strategie zur Abwehr von Randomisierungsangriffen, sodass der Angreifer nicht unterscheiden kann, ob der Wähler seinen Anweisungen gefolgt ist.

SA.1 Es existiert eine Strategie zur Abwehr von Simulationsangriffen, sodass der Angreifer nicht unterscheiden kann, ob der Wähler seinen Anweisungen gefolgt ist.

AA.1 impliziert offensichtlich AA.2, da hier unterschieden wird, ob der Angreifer Informationen des Nutzers benötigt, um einen solchen Angriff abwehren zu können. Die restlichen Eigenschaften sind voneinander unabhängig. Ein Wahlsystem erfüllt also nur dann die Nicht-Erpressbarkeit, wenn es die Eigenschaften AA.1, RA.1 und SA.1 erfüllt. Ansonsten erfüllt das System nur die Eigenschaft der „teilweisen Nicht-Erpressbarkeit“.

Robustheit

RI.1 **Innere Robustheit:** Die Robustheit gegen Angriffe, welche die Auszählung verhindern, ohne die Verfügbarkeit des Wahlsystems anzugreifen.

Benutzbarkeit

BENUTZBARKEIT BEI DER WAHL

BW.1 **Klick-Voting:** Der Wahlvorgang gestaltet sich derart, dass der Wähler sich ausschließlich durch einen Wahlassistenten klicken muss.

BW.2 **Einfache Eingabe:** Der Wähler muss eine Zeichenfolge (z. B. einen Code) in den Wahlclient eingeben.

BW.3 **Selbst zusammengesetzte Eingabe:** Der Wähler muss seine Eingabe selbst zusammensetzen und diese eingeben.

Offensichtlich gilt BW.1 ist am effektivsten und BW.3 ist am wenigsten effektiv. Für die Effizienz wird zwischen einer Wahl mit und ohne Hilfsmittel sowie der einmaligen und mehrfachen Teilnahme am Wahlprotokoll unterschieden.

BW.x.1 **Einmalige Teilnahme ohne Hilfsmittel:** Der Wähler muss nur einmal aktiv am Wahlvorgang teilnehmen und benötigt dazu keine Hilfsmittel.

BW.x.2 **Einmalige Teilnahme mit Hilfsmittel:** Der Wähler muss nur einmal aktiv am Wahlvorgang teilnehmen, benötigt zur Wahl allerdings Hilfsmittel.

BW.x.3 **Mehrfache Teilnahme ohne Hilfsmittel:** Der Wähler muss mehrmals aktiv am Wahlvorgang teilnehmen, benötigt dazu jedoch keine Hilfsmittel.

BW.x.4 **Mehrfache Teilnahme mit Hilfsmittel:** Der Wähler muss mehrmals aktiv am Wahlvorgang teilnehmen und benötigt zur Wahl außerdem Hilfsmittel.

Auch hier sind die Eigenschaften absteigend nach der Effizienz geordnet. Mit der mehrfachen Teilnahme ist gemeint, dass der Wähler oder sein Wahlclient an unterschiedlichen Wahlphasen aktiv teilnehmen muss.

BENUTZBARKEIT BEI DER VERIFIZIERUNG

BV.1 **Vergleiche auf Basis von Zeichenketten:** Der Wähler muss zur individuellen Verifizierung vergleichen, ob die Zeichenkette, welche er beim Wahlvorgang erhalten hat, die gleiche ist, wie auf dem Bulletin Board bzw. in den postalisch übersandten Wahlunterlagen.

BV.2 **Überprüfen von Zero-Knowledge-Beweisen:** Der Wähler muss zur individuellen Verifizierung mindestens einen Zero-Knowledge-Beweis überprüfen.

Hierbei ist BV.1 die Eigenschaft, welche gegenüber BV.2 zu bevorzugen ist, da das Vergleichen zweier Strings offensichtlich einfacher durchzuführen ist, als die Überprüfung eines Zero-Knowledge-Beweises.

A.2.2 Zusammenfassung Angreifer-Fähigkeiten

Kommunikation

Angriffe auf die Kommunikation sind für den Angreifer teilweise relativ einfach durchzuführen. Er muss dazu nicht Teil des Wahlsystems sein, sondern kann die Nachrichten an einem zentralen Knotenpunkt abfangen. Umso wichtiger ist es deshalb, die Kommunikation durch geeignete Maßnahmen zu schützen.

K.1 Der Angreifer kann die Kommunikationskanäle passiv abhören.[6]

K.2 Der Angreifer kann die Kommunikationskanäle aktiv manipulieren.[7]

K.3 Der Angreifer kann dem Wähler ein bestimmtes Hilfsmittel zuordnen.

Hilfsmittel

Diese Kategorie ist nicht bei allen Wahlsystemen vertreten, weshalb diese Angriffe für viele Wahlsysteme keine Anwendung finden.

H.1 Der Angreifer kann Hilfsmittel außerhalb des Userspaces [8] unverändert kopieren.

H.2 Der Angreifer kann existierende Hilfsmittel außerhalb des Userspaces ohne Vervielfältigung manipulieren.

H.3 Der Angreifer kann eigene Hilfsmittel erstellen / fälschen und dem System zuführen.

Wahlkomponenten

Diese Fähigkeiten sind äußerst kritisch, da sie sich auf das Innere des Wahlverfahrens beziehen. S.2 und S.3 implizieren deshalb die Fähigkeiten K.1 und K.2 von allen angrenzenden Kanälen, wobei die Transportschichtsicherheit in diesem Fall keine Rolle mehr spielt. Für S.1 trifft dies nicht in jedem Fall zu, da das Einfügen von Nachrichten auf dem Bulletin Board abhängig von der eingesetzten Platform z. B. auch per Cross-Site-Scripting möglich sein könnte. Aufgrund des öffentlichen Charakters der auf dem Bulletin Board veröffentlichten Informationen wäre eine Kompromittierung der Vertraulichkeit ohnehin kein Problem. Eine Kompromittierung von Integrität und Authentizität, was das unberechtigte Einfügen von Nachrichten darstellt, allerdings durchaus.

S.1 Der Angreifer kann Nachrichten auf dem Bulletin Board einfügen.

S.2 Der Angreifer kontrolliert den Wahlcomputer.

S.3 Der Angreifer kontrolliert einige, jedoch nicht alle Wahlserver.

S.4 Der Angreifer kontrolliert alle Wahlserver.

Wahloffizielle

Diese Fähigkeit ist ebenfalls sehr kritisch, da der Angreifer dadurch an Schlüsselmaterial gelangen kann.

O.1 Der Angreifer kontrolliert einige, jedoch nicht alle Wahloffizielle.

---

6 Die Fähigkeit umfasst: Die Benutzung eines Kanals zu erkennen, zu entscheiden wer der Sender einer bestimmten Nachricht ist und die Nachricht abzuhören.

7 Die Fähigkeit umfasst: Blockieren, Einfügen sowie das Modifizieren von Nachrichten.

8 Zwei Fälle sind dabei denkbar: Der Angreifer erlangt Zugriff auf Hilfsmittel während der Produktion oder während des Transports. Aufgrund des gleichen Resultat werden diese beiden Fälle zusammengefasst.

A.2.3 Angreifermodelle für Wahlen unterschiedlicher Ordnung

Angreifermodell für Wahlen dritter Ordnung

Für Wahlen dritter Ordnung wird ein schwaches Angreifermodell gewählt. Es wird hierfür angenommen, dass der Angreifer nicht ins System eindringen kann. Folglich besitzt er lediglich die Fähigkeiten K.1 (passives Abhören von Kommunikationskanälen) und K.2 (aktives Manipulieren von Kommunikationskanälen).

Angreifermodell für Wahlen zweiter Ordnung

Für Wahlen zweiter Ordnung werden dem Angreifer ebenfalls die Fähigkeiten K.1 (passives Abhören von Kommunikationskanälen) und K.2 (aktives Manipulieren von Kommunikationskanälen) sowie zusätzlich S.1 (Einfügen von Nachrichten auf dem Bulletin Board) und S.3 (Kontrolle über einige, jedoch nicht alle Wahlserver) zugewiesen.

Angreifermodell für Wahlen erster Ordnung ohne Angriff auf Produktion von Hilfsmitteln

Bei Wahlen erster Ordnung wird unterschieden, ob der Angreifer Zugriff auf die Produktion der Hilfsmittel besitzt oder nicht. In der Praxis wäre diese Unterscheidung z. B. relevant, wenn der die Wahl durchführende Staat die Produktion der Hilfsmittel nicht überwachen lassen bzw. die Hilfsmittel sogar im Ausland und damit komplett außerhalb seiner Kontrolle einkaufen würde. Hat der Angreifer keinen Zugriff auf die Hilfsmittel, wird davon ausgegangen, dass der Angreifer die folgenden Fähigkeiten besitzt: K.1 (passives Abhören von Kommunikationskanälen), K.2 (aktives Manipulieren von Kommunikationskanälen), S.1 (Einfügen von Nachrichten auf dem Bulletin Board), S.2 (Kontrolle über den Wahlcomputer) , S.4 (Kontrolle über alle Server). Wobei jedoch weiterhin angenommen wird, dass Server und Wahlcomputer nicht zusammenarbeiten.

Angreifermodell für Wahlen erster Ordnung mit Angriff auf Produktion von Hilfsmitteln

Besitzt der Angreifer die Fähigkeit, die Hilfsmittel zu manipulieren, so kommen zu den oben genannten Fähigkeiten noch H.1 (unverändertes Kopieren von Hilfsmitteln außerhalb des Userspace), H.2 (Manipulation von Hilfsmitteln ohne Vervielfältigung außerhalb des Userspace), H.3 (Herstellung / Fälschung eigener Hilfsmittel und Zuführung zum System).

A.3 KORRESPONDENZ MIT DER STELLE DES BUNDESWAHLLEITERS BEIM STATISTISCHEN BUNDESAMT

Bestaetigungs-E-Mail - Bundeswahlleiter

Subject: Bestaetigungs-E-Mail - Bundeswahlleiter
From: [redacted]@destatis.de
Date: 21.08.2017 09:16
To: [redacted]@uni-ulm.de

Sehr geehrte(r) Herr Heinl,

vielen Dank fuer Ihre Nachricht. Wir werden Ihnen so schnell wie moeglich antworten.

Sie haben uns folgende Informationen uebermittelt:
Herr Michael Heinl
Schüler/in, Student/in

DEU
[redacted]

Ihre Nachricht:
Sehr geehrter [redacted]

im Rahmen einer Forschungsarbeit an der Universität Ulm beschäftigen uns folgende Fragen, die wir bislang leider nicht durch öffentlich zugängliche Quellen klären konnten:

- Welche Maßnahmen sind vorgesehen, damit sich an der Briefwahl teilnehmende Wahlberechtigte in örtlicher Abwesenheit versichern können, dass
    a) ihre postalisch versandten, ausgefüllten Briefwahlunterlagen tatsächlich bei der vorgesehenen Gemeindebehörde angekommen sind.
    b) ihr Stimmzettel tatsächlich in der Wahlurne gelandet ist und bei der Auszählung der Stimmen berücksichtigt wurde.

- Wie wird sichergestellt, dass per E-Mail beantragte Briefwahlunterlagen, deren Versandanschrift von der im Melderegister bzw. Wählerverzeichnis aufgeführten Adresse abweicht, tatsächlich von den entsprechenden Wahlberechtigten und nicht von unbefugten Dritten beantragt wurden?

- Wie ist der vorgesehene Ablauf, falls eine wahlberechtigte Person am Wahltag mit der Absicht zu wählen im Wahllokal erscheint und vom Wahlvorstand festgestellt wird, dass für diese Person bereits Briefwahlunterlagen beantragt wurden, die betreffende Person dies jedoch abstreitet?

- Wie wird die Amtlichkeit der eingegangenen Briefumschläge überprüft? Haben die Umschläge bestimmte Sicherheitsmerkmale?

Es würde uns sehr freuen, wenn Sie uns bei unserer Arbeit unterstützen und uns oben aufgeführten Fragen beantworten könnten.

Für Rückfragen stehen wir Ihnen selbstverständlich jederzeit zur Verfügung.

Mit freundlichen Grüßen
Simon Gölz und Michael Heinl

Bitte beachten Sie:
Dies ist eine automatisch versendete Nachricht. Bitte antworten Sie nicht auf dieses

Bestaetigungs-E-Mail - Bundeswahlleiter

Schreiben, da die E-Mail-Adresse lediglich zum Versenden der E-Mail-Bestaetigung
eingerichtet ist. Die Kontaktmoeglichkeiten finden Sie unter
https://www.bundeswahlleiter.de/info/kontakt.html.
Vielen Dank fuer Ihren Besuch auf unserer Homepage!

Briefwahl, Der Bundeswahlleiter, GZ [redacted]

Subject: Briefwahl, Der Bundeswahlleiter, GZ [redacted]
From: [redacted]@bundeswahlleiter.de
Date: 25.08.2017 09:43
To: [redacted]@uni-ulm.de

Statistisches Bundesamt
Der Bundeswahlleiter

[redacted]
https://www.bundeswahlleiter.de/kontakt/

Sehr geehrter Herr Heinl,

vielen Dank für Ihre Mitteilung vom 21. August 2017.

Zu Ihren Fragen nehmen wir - wie folgt - Stellung:

- Welche Maßnahmen sind vorgesehen, damit sich an der Briefwahl teilnehmende Wahlberechtigte in örtlicher Abwesenheit versichern können, dass
a) ihre postalisch versandten, ausgefüllten Briefwahlunterlagen tatsächlich bei der vorgesehenen Gemeindebehörde angekommen sind.

Bundeswahlgesetz und Bundeswahlordnung enthalten keine Regelungen etwaig zu treffender Maßnahmen. "Die Verantwortung dafür, dass der Wahlbrief der zuständigen Stelle rechtzeitig zum Ende der Wahlzeit vorliegt, und das Risiko einer verspäteten Ankunft des Wahlbriefes, das bei einer Übermittlung per Post nie völlig auszuschließen ist, trägt mithin grundsätzlich der Wahlberechtigte, selbst wenn ihn kein persönliches Verschulden trifft" (W. Schreiber, Kommentar zum Bundeswahlgesetz, 10. Auflage, Rd.-Nr. 12 zu § 36).

b) ihr Stimmzettel tatsächlich in der Wahlurne gelandet ist und bei der Auszählung der Stimmen berücksichtigt wurde.

Die bei der zuständigen Stelle eingegangenen Wahlbriefe werden ungeöffnet gesammelt und unter Verschluss gehalten. Die weitere Behandlung der Wahlbriefe (siehe auch §§ 74 f BWO) ist auf folgender Internetseite schematisch dargestellt.

https://www.bundeswahlleiter.de/dam/jcr/3795d718-6094-48a6-831f-507ece60a8d8/btw_schaubild_briefwahl.pdf

---

[Anmerkung der Autoren: Aktuelle URL (Stand 24. September 2019): https://www.bundeswahlleiter.de/dam/jcr/3795d718-6094-48a6-831f-507ece60a8d8/btw_schaubild_briefwahl.png]

- Wie wird sichergestellt, dass per E-Mail beantragte Briefwahlunterlagen, deren Versandanschrift von der im Melderegister bzw. Wählerverzeichnis aufgeführten Adresse abweicht, tatsächlich von den entsprechenden Wahlberechtigten und nicht von unbefugten Dritten beantragt wurden?

Gemäß § 28 Absatz 4 BWO versenden in derartigen Fällen (abweichende Versandanschrift) die Gemeindebehörden eine entsprechende Mitteilung an die Wohnanschrift.

- Wie ist der vorgesehene Ablauf, falls eine wahlberechtigte Person am Wahltag mit der Absicht zu wählen im Wahllokal erscheint und vom Wahlvorstand festgestellt wird, dass für diese Person bereits Briefwahlunterlagen beantragt wurden, die betreffende Person dies jedoch abstreitet?

Briefwahl, Der Bundeswahlleiter, GZ [redacted]

Gemäß § 56 Absatz 6 Nr. 2 BWO wäre die Person zurückzuweisen.

- Wie wird die Amtlichkeit der eingegangenen Briefumschläge überprüft? Haben die Umschläge bestimmte Sicherheitsmerkmale?

Die Wahlbriefumschläge sollen gemäß § 45 Absatz 4 BWO eine bestimmte Größe und Farbe haben und nach dem Muster der Anlage 11 beschriftet sein (u.a. Wahlscheinnummer oder Wahlbezirk). Entsprechend diesen Vorgaben werden die Umschläge von den Kreiswahlleitungen beschafft.

Wir hoffen, dass wir Ihnen weiterhelfen konnten.

Mit freundlichen Grüßen
Im Auftrag

[redacted]

_____
Um Ihre Anfrage zügig beantworten zu können, haben wir Ihre Angaben wie Namen und Adresse elektronisch gespeichert.
Für weitere Anfragen halten wir diese Angaben intern - ausschließlich zum Zwecke der Kundenpflege - vor.
Falls Sie damit nicht einverstanden sind, bitten wir um Mitteilung.

Re: Briefwahl, Der Bundeswahlleiter, GZ [redacted]

Subject: Re: Briefwahl, Der Bundeswahlleiter, GZ [redacted]
From: [redacted]@uni-ulm.de
Date: 29.08.2017 14:30
To: [redacted]@bundeswahlleiter.de
CC: [redacted] @uni-ulm.de

[redacted]

vielen dank für Ihre schnelle und ausführliche Rückmeldung, die bereits einen Großteil unserer Fragen beantwortet. Nichtsdestotrotz gibt es noch ein paar Unklarheiten, die wir Ihnen im Folgenden gerne schildern und über deren Klarstellung wir uns sehr freuen würden.

1. Sie schreiben, dass es in der Verantwortung des Wählers liege, dass der Wahlbrief rechtzeitig zum Ende der Wahlzeit vorliegt. Zeitlichem Verzug kann der Wähler in der Tat eigenverantwortlich durch rechtzeitigs Absenden des Wahlbriefes entgegenwirken. Allerdings kann er durch frühzeitiges Absenden nicht automatisch sicher sein, dass der Wahlbrief aufgrund äußerer Widrigkeiten überhaupt ankommt (z. B. kann der Wahlbrief auf dem Postweg verloren gehen). Es würde uns deshalb interessieren, ob es für den Wähler grundsätzlich irgend eine gesetzlich vorgesehene Möglichkeit gibt, das Eintreffen des Briefes bei der Wahlbehörde (unabhängig von der zeitlichen Komponente) festzustellen (z. B. telefonisch o. ä.).

2. Vielen Dank für die Bereitstellung der schematischen Darstellung der Behandlung der Wahlbriefe [1]. Da es grundsätzlich dem Sinn der Briefwahl widerspräche, dass der Wähler selbst vor Ort ist, um das Prozedere zu überprüfen, würde uns vor allem interessieren, ob der Wähler nachträglich in der Lage ist, festzustellen, dass sein Stimme korrekt in der Urne gelandet ist. Gibt es hierfür einen Prozess, um dem Wähler nachträglich Einsicht in entsprechende Unterlagen zu gewähren, z. B. ein vom Wahlvorstand unterschriebenes Verzeichnis der Wähler, deren Wahlbriefe ausgewertet wurden, oder die archivierten Wahlscheine, woraus hervorgeht, dass seine Stimme tatsächlich berücksichtigt wurde?

3. Wie sieht eine solche Mitteilung gemäß § 28 Absatz 4 BWO bei abweichender Versandanschrift aus?
    a) Welche Informationen enthält diese Mitteilung an die Wohnanschrift? Enthält sie auch die abweichende Versandanschrift selbst?
    b) Gilt der doppelte Versand der Briefwahlunterlagen an Wohn- sowie abweichender Versandanschrift bereits als Mitteilung in diesem Sinne?
    c) Was geschieht, falls in dem in b) genannten Falle tatsächlich beide offiziell ausgestellten Wahlbriefe ausgefüllt wieder an die Gemeindebehörde zurückgeschickt werden?

4. Wie sieht der weitere Prozess aus, wenn ein Wähler eine Mitteilung an seine Wohnanschrift erhält, dass seine Briefwahlunterlagen an eine andere Versandanschrift versendet wurden, er diesen Versand aber selbst gar nicht veranlasst hat?

5. Gab es in der Vergangenheit Fälle, in denen Briefwahlunterlagen bzw. -umschläge versucht wurden, zu fälschen? Falls ja, was waren die Hintergründe? Gibt bzw. gab es Überlegungen, Briefwahlunterlagen bzw. -umschläge durch bestimmte Sicherheitsmerkmale, wie sie z. B. bei Geldnoten Verwendung finden, zusätzlich gegen Fälschung abzusichern?

Auf Ihre Antwort freuen wir uns.

Re: Briefwahl, Der Bundeswahlleiter, GZ [redacted]

Herzliche Grüße
Simon Gölz und Michael Heinl

[1] https://www.bundeswahlleiter.de/dam/jcr/3795d718-6094-48a6-831f-507ece60a8d8/btw_schaubild_briefwahl.pdf

Briefwahl, Der Bundeswahlleiter, GZ [redacted]

Subject: Briefwahl, Der Bundeswahlleiter, GZ [redacted]
From: [redacted]@bundeswahlleiter.de
Date: 20.10.2017 11:33
To: [redacted]@uni-ulm.de

Statistisches Bundesamt
Der Bundeswahlleiter

[redacted]
https://www.bundeswahlleiter.de/kontakt/

Sehr geehrter Herr Heinl,

vielen Dank für Ihre Anfrage vom 29. August 2017. Leider hat die große Zahl der Zuschriften unsere Kapazitäten gesprengt, sodass wir nicht zeitnah antworten konnten. Dafür entschuldigen wir uns.

Heute möchte ich Ihre Rückfragen beantworten, ich hoffe, dies ist Ihnen hilfreich.

"Es würde uns deshalb interessieren, ob es für den Wähler grundsätzlich irgend eine gesetzlich vorgesehene Möglichkeit gibt, das Eintreffen des Briefes bei der Wahlbehörde (unabhängig von der zeitlichen Komponente) festzustellen (z. B. telefonisch o. ä.)."

[redacted]

[Anmerkung der Autoren: Die vom Büro des Bundeswahlleiters am 27.09.2019 zur Veröffentlichung freigegebene, modifizierte Antwort auf diese Frage steht am Ende dieser Email.]

"würde uns vor allem interessieren, ob der Wähler nachträglich in der Lage ist, festzustellen, dass sein Stimme korrekt in der Urne gelandet ist. Gibt es hierfür einen Prozess, um dem Wähler nachträglich Einsicht in entsprechende Unterlagen zu gewähren, z. B. ein vom Wahlvorstand unterschriebenes Verzeichnis der Wähler, deren Wahlbriefe ausgewertet wurden, oder die archivierten Wahlscheine, woraus hervorgeht, dass seine Stimme tatsächlich berücksichtigt wurde?"

§ 89 Absatz 2 BWO bestimmt, dass Auskünfte aus Wählerverzeichnissen, Wahlscheinverzeichnissen und Verzeichnissen nach § 28 Abs. 8 Satz 2 (Anmerkung: Verzeichnis der für ungültig erklärten Wahlscheine) und § 29 Abs. 1 (Anmerkung: Verzeichnis der wahlberechtigten Personen aus der Gemeinde, die in besonderen Einrichtungen wählen) nur Behörden, Gerichten und sonstigen amtlichen Stellen des Wahlgebiets erteilt werden dürfen.

Ein Einsichtsrecht besteht für den/die Wahlberechtigte/n nur hinsichtlich des

Briefwahl, Der Bundeswahlleiter, GZ [redacted]

Wählerverzeichnisses.

"Wie sieht eine solche Mitteilung gemäß § 28 Absatz 4 BWO bei abweichender Versandanschrift aus?
a) Welche Informationen enthält diese Mitteilung an die Wohnanschrift? Enthält sie auch die abweichende Versandanschrift selbst?"

Enthalten muss die Mitteilung die Information, dass Briefwahlunterlagen an eine abweichende Anschrift gesendet worden sind. Ob in der Praxis in der Mitteilung auch die abweichende Anschrift genannt wird, ist hier nicht bekannt.

"b) Gilt der doppelte Versand der Briefwahlunterlagen an Wohn- sowie abweichender Versandanschrift bereits als Mitteilung in diesem Sinne?"

Zur Klarstellung: Briefwahlunterlagen werden nicht doppelt versandt, sondern entweder an die Wohnanschrift o d e r an eine abweichende Anschrift, wenn dies beantragt wird.

"c) Was geschieht, falls in dem in b) genannten Falle tatsächlich beide offiziell ausgestellten Wahlbriefe ausgefüllt wieder an die Gemeindebehörde zurückgeschickt werden?"

Siehe Frage b)

"Wie sieht der weitere Prozess aus, wenn ein Wähler eine Mitteilung an seine Wohnanschrift erhält, dass seine Briefwahlunterlagen an eine andere Versandanschrift versendet wurden, er diesen Versand aber selbst gar nicht veranlasst hat?"

Bei missbräuchlicher Beantragung durch einen Dritten und Versendung an eine andere Anschrift kann der Wahlberechtigte nach Erhalt der Kontrollmitteilung gegenüber der Gemeindebehörde nach § 28 Absatz 10 Satz 2 BWO glaubhaft machen, dass ihm der Wahlschein nicht zugegangen ist. Die Gemeinde kann ihm nach § 28 Absatz 10 Satz 2 BWO einen neuen Wahlschein erteilen. Der erste Wahlschein ist nach § 28 Absatz 8 Satz 1 BWO für ungültig zu erklären, so dass ein unberechtigter Dritter damit nicht wählen kann.

"Gab es in der Vergangenheit Fälle, in denen Briefwahlunterlagen bzw. -umschläge versucht wurden, zu fälschen? Falls ja, was waren die Hintergründe?"

Derartige Fälle sind hier nicht bekannt.

"Gibt bzw. gab es Überlegungen, Briefwahlunterlagen bzw. -umschläge durch bestimmte Sicherheitsmerkmale, wie sie z. B. bei Geldnoten Verwendung finden, zusätzlich gegen Fälschung abzusichern?"

Derartige Überlegungen gibt es nach unserer Einschätzung nicht. Es würde auch nicht ausreichen, den amtlichen Stimmzettel, den Stimmzettelumschlag und den Wahlbriefumschlag zu fälschen. Zudem müssten der Wahlschein gefälscht werden, und die Unterschrift des Wählers bei der Versicherung an Eides statt.

Sollten Sie noch Rückfragen haben, wollen wir sie gerne beantworten, allerdings bin ich erst am 6. November wieder im Hause.

Mit freundlichen Grüßen
Im Auftrag
[redacted]

Briefwahl, Der Bundeswahlleiter, GZ [redacted]

_____
Um Ihre Anfrage zügig beantworten zu können, haben wir Ihre Angaben wie Namen und Adresse elektronisch gespeichert.
Für weitere Anfragen halten wir diese Angaben intern - ausschließlich zum Zwecke der Kundenpflege - vor.
Falls Sie damit nicht einverstanden sind, bitten wir um Mitteilung.

[Anmerkung der Autoren: Die vom Büro des Bundeswahlleiters am 27.09.2019 zur Veröffentlichung freigegebene, modifizierte Antwort auf die erste Frage dieser Email:]

"Es würde uns deshalb interessieren, ob es für den Wähler grundsätzlich irgend eine gesetzlich vorgesehene Möglichkeit gibt, das Eintreffen des Briefes bei der Wahlbehörde (unabhängig von der zeitlichen Komponente) festzustellen (z. B. telefonisch o. ä.)."

Die wahlrechtlichen Vorschriften sehen nicht vor, dass Gemeinden Eingangsbestätigungen über eingegangene Wahlbriefe erteilen. Möchte ein Wähler/eine Wählerin Sicherheit über den Eingang des Wahlbriefs bei der Gemeindebehörde haben, hat er/sie verschiedene Möglichkeiten:

Wähler/-innen können den Wahlbrief in einer besonderen Versendungsform (z.B. als Einschreiben mit Rückschein) versenden, wenn sie eine Bestätigung über den Eingang des Wahlbriefs bei der Gemeindebehörde wünschen. Die Versendungskosten des Wahlbriefs muss der Wähler/die Wählerin in diesem Fall jedoch selbst tragen.

Selbstverständlich besteht auch die Möglichkeit, den Wahlbrief an der auf dem Wahlbriefumschlag angegebenen Adresse einzuwerfen bzw. ihn abzugeben.

Schließlich besteht häufig die Möglichkeit der Briefwahl an Ort und Stelle. Gemeinden sollen dem Wahlberechtigten, wenn er den Wahlschein (mit Briefwahlunterlagen) persönlich bei der Gemeindebehörde abholt, Gelegenheit geben, Briefwahl an Ort und Stelle auszuüben (§ 28 Absatz 5 Satz 1 Bundeswahlordnung). Der Wähler kennzeichnet den Stimmzettel in diesem Fall unbeobachtet vor Ort, legt ihn in den Stimmzettelumschlag, unterzeichnet die Versicherung an Eides statt zur Briefwahl und legt sie mit dem verschlossenen Stimmzettelumschlag in den Wahlbriefumschlag, den er zuklebt. Den Wahlbriefumschlag übergibt der Wähler/die Wählerin den Gemeindemitarbeitern oder wirft ihn gegebenenfalls in eine für diesen Zweck bereit gehaltene Wahlurne. Durch die Briefwahl an Ort und Stelle entfällt das Versendungsrisiko. Der Wähler/die Wählerin kann sicher sein, dass der Wahlbrief bei der Gemeindebehörde eingegangen ist.
