ParTCP-Dokumentation

Release 0.9

Martin Wandelt

06.12.2022

Inhalt:

1 Vorbemerkungen **1**
- 1.1 Begriffe . . . 1
- 1.2 Sequenzdiagramme . . . 2
- 1.3 Beispielnachrichten . . . 3

2 Einleitung **5**
- 2.1 Anforderungen . . . 5
- 2.2 Bausteine . . . 6
- 2.3 Nicht-geheime Abstimmungen . . . 8
- 2.4 Geheime Abstimmungen . . . 9

3 Nachrichten **11**

4 Vertrauenswürdige Maschinen **13**
- 4.1 Transparenz . . . 13
- 4.2 Abschottung . . . 14

5 Kryptosystem **15**
- 5.1 Universelle öffentliche Schlüssel . . . 15
- 5.2 Signaturen . . . 16
- 5.3 Verschlüsselung . . . 16

6 Wahlprotokoll **17**
- 6.1 Registrierung . . . 17
- 6.2 Vorbereitung einer Veranstaltung . . . 20
- 6.3 Durchführen einer Abstimmung . . . 24

7 DSGVO-Konformität **29**

8 Verzeichnisstruktur **31**

9 Abschottungsprozedur **33**
- 9.1 Klärung der Verantwortung . . . 33
- 9.2 Zufallsentscheide . . . 33
- 9.3 Auswahl und Aufstellen der Maschine . . . 34
- 9.4 Installation des Betriebssystems . . . 34
- 9.5 Installation und Test der Anwendung . . . 35
- 9.6 Abschottung und Versiegelung . . . 35

9.7 Bericht 36

10 Verzeichnisse und Tabellen **37**

KAPITEL 1

Vorbemerkungen

Dieses Dokument beschreibt die technischen Einzelheiten der ParTCP-Plattform und insbesondere der Abstimmungsfunktionen. Es soll interessierten Personen ermöglichen, die Sicherheit der Plattform zu beurteilen, ohne sich mit den Implentierungsdetails zu befassen.

1.1 Begriffe

Im Folgenden sind die Begriffe erklärt, die im Zusammenhang mit ParTCP eine besondere Bedeutung haben oder in einem engeren Sinne verwendet werden als im allgemeinen Sprachgebrauch üblich.

Eine **Gemeinschaft** (community) ist eine Gruppe von Menschen, die den Wunsch haben oder aus äußeren Gründen gezwungen sind, gemeinsame Entscheidungen auf demokratischer Grundlage zu treffen. Die Menschen, die zu einer bestimmten Gemeinschaft gehören, werden als deren **Mitglieder** (members) bezeichnet. Mitglieder, die an einem ParTCP-Server registriert sind, heißen **Teilnehmer** (participants).

Eine Gemeinschaft kann in **Gruppen** gegliedert sein, die vorübergehend oder dauerhaft eigene Gemeinschaften bilden.

Eine **Mitgliederliste** ist eine Liste oder Datenbank, in der die Menschen verzeichnet sind, die Mitglied der Gemeinschaft sind oder zu einem früheren Zeitpunkt waren. Die Informationen, die in der Liste zu einem bestimmten Mitglied gespeichert sind, bilden den **Mitgliedsdatensatz**. Die Mitgliederliste ist nicht Teil der ParTCP-Plattform, sondern wird von einem externen System bereitgestellt. Sie muss nicht zwangsläufig in digitaler Form vorliegen, sondern es kann sich auch um Karteikarten, Aufzeichnungen auf Papier o. ä. handeln.

Ein **Schlüsselserver** ist ein zentraler Rechner, den eine Gemeinschaft betreibt, um die Identitäten ihrer Mitglieder zu verwalten. Eine Gemeinschaft kann auch mehrere Schlüsselserver betreiben, wenn es in der Mitgliederliste eindeutige und unveränderliche Kriterien gibt, aus denen abgeleitet werden kann, welcher Schlüsselserver für ein bestimmtes Mitglied zuständig ist.

Eine **Identität** (identity) ist eine Kombination aus einer Teilnehmerkennung und einem damit verknüpften öffentlichen Schlüssel. Auf einem bestimmten Schlüsselserver kann es nicht mehrere Identitäten mit derselben Teilnehmerkennung geben. Zu unterscheiden sind **offene** (public) Identitäten, die mehr oder weniger nachvollziehbar mit einer realen Person verknüpft sind, und **anonyme** (anonymous) Identitäten, die so erstellt werden, dass eine solche Verknüpfung nicht möglich ist.

Eine **Registrierung** (registration) bezeichnet den Vorgang, bei dem für ein Mitglied einer Gemeinschaft eine offene Identität auf dem Schlüsselserver erstellt wird.

Ein **Administrator** ist ein Mitglied der Gemeinschaft, das berechtigt ist, die Mitgliederliste einzusehen und Registrierungen durchzuführen.

Ein **Abstimmungsserver** ist ein zentraler Rechner, der zur Abwicklung von Veranstaltungen und Abstimmungen einer Gemeinschaft dient. Ein Abstimmungsserver kann zugleich Schlüsselserver sein, und umgekehrt.

Eine **Veranstaltung** (event) ist ein zeitlich befristeter organisatorischer Rahmen mit festem Teilnehmerkreis, innerhalb dessen eine oder mehrere Abstimmungen stattfinden. Alle Teilnehmer einer Veranstaltung sind grundsätzlich berechtigt, an allen Abstimmungen dieser Veranstaltung teilzunehmen. Werden abgestufte Berechtigungen benötigt, müssen für die einzelnen Teilnehmerkreise jeweils eigene Veranstaltungen definiert werden.

Jede Veranstaltung hat mindestens einen **Veranstaltungsleiter** (event manager), der die Berechtigung hat, Abstimmungen zu definieren, zu starten und zu beenden, Teilnehmer einzuladen etc.

Eine **Abstimmung** (voting) ist ein Vorgang, bei dem die Teilnehmer einer Veranstaltung eine Liste von Optionen vorgelegt bekommen und diese mit einer Bewertung versehen. Der **Abstimmungstyp** (voting type) bestimmt darüber, welche Bewertungsmöglichkeiten zur Verfügung stehen und wie aus den Einzelbewertungen am Ende das Abstimmungsergebnis berechnet wird. Das heißt, auch Wahlen und Konsensierungen zählen im ParTCP-Jargon zu den Abstimmungen.

Ein **Client** ist ein Computerprogramm, mit dessen Hilfe ein Mitglied mit Schlüssel- und Abstimmungsservern kommunizieren kann. Der Client stellt Funktionen zur Verfügung, um Schlüsselpaare zu erzeugen und zu verwalten, Nachrichten zu verschlüsseln und zu signieren und digitale Stimmzettel auszufüllen. Ein Client kann außerdem spezielle Funktionen für Administratoren und Veranstaltungsleiter bereitstellen.

Eine **Nachricht** ist eine Zeichenkette im YAML-Format, die zum Datenaustausch zwischen Clients und Servern dient. Einzelheiten hierzu sind im Abschnit „Nachrichten“ zu finden.

Eine **vertrauenswürdige Maschine** ist ein Computersystem, das von vertrauenswürdigen Personen nach festgelegten Regeln eingerichtet und gegen nachträgliche Veränderungen abgeschottet wurde (siehe den Abschnitt „Vertrauenswürdige Maschinen“).

1.2 Sequenzdiagramme

Dieses Dokument nutzt Sequenzdiagramme, um Prozesse zu visualisieren. Dabei werden die beteiligten Akteure (Personen oder Computer) als Spalten dargestellt und die Daten, die zwischen ihnen ausgetauscht werden als Pfeile zwischen den Spalten. Durchgezogene Pfeile bedeuten, dass es sich um einen Datenaustausch innerhalb der ParTCP-Plattform handelt; gestrichelte Pfeile bezeichnen einen nicht näher definierten Kommunikationsweg.

Wenn der Datenaustausch in Form einer ParTCP-Nachricht stattfindet, ist der Nachrichtentyp in Fettschrift angegeben, und darunter werden in Klammern die wichtigsten Nachrichtenelemente aufgezählt.

Wenn in der Spalte „Mitglied“ der weiße Aktivitätsbalken durch einen grauen Aktivitätsbalken überlagert wird, bedeutet dies, dass das Mitglied Nachrichten nicht unter seiner offenen Identität versendet, sondern eine anonyme Identität nutzt.

Eine allgemeine Einführung in diese Art der Visualisierung gibt der Wikipedia-Artikel zum Stichwort „Sequenzdiagramm“.

1.3 Beispielnachrichten

Im Anschluss an die Sequenzdiagramme werden jeweils die wichtigsten der darin verwendeten Nachrichtentypen in Form von Beispielnachrichten wiedergegeben. Aus Gründen der Lesbarkeit sind hier die Signaturen und Zeitstempel weggelassen und die Inhalte mancher Nachrichtenelemente gekürzt wiedergeben. Die gekürzten Stellen sind jeweils durch drei Auslassungspunkte (…) gekennzeichnet.

KAPITEL 2

Einleitung

Dieses Kapitel gibt einen Überblick über das Wahlprotokoll, das in ParTCP eingesetzt wird. Der erste Abschnitt geht der Frage nach, welche Anforderungen erfüllt sein müssen, damit ein Wahlprotokoll als sicher und vertrauenswürdig eingestuft werden kann. Der zweite Abschnitt beschreibt die einzelnen Komponenten, auf die bei der Umsetzung des Protokolls zurückgegriffen wird, bevor dann der letzte Abschnitt den Ablauf einer Wahl überblicksartig beschreibt. Technische Einzelheiten finden sich weiter hinten im Abschnitt „Wahlprotokoll“.

2.1 Anforderungen

Unabhängig davon, ob eine Abstimmung durch Urnengang, per Briefwahl oder auf elektronischem Weg durchgeführt wird, muss das Verfahren einige grundsätzliche Anforderungen erfüllen:

- Identifizierbarkeit Es muss sichergestellt sein, dass nur berechtigte Personen eine Stimme abgeben können und dass eine bestimmte Person dies auch nur einmal tun kann. Dazu gehört in der Regel, dass nachweisbar sein muss, ob eine bestimmte Person an der Abstimmung teilgenommen hat, oder nicht.
- Anonymität Es darf niemandem möglich sein, herauszufinden, *wie* eine bestimmte Person abgestimmt hat. Dies gilt für Außenstehende ebenso wie für Behörden und Einrichtungen, die an der Durchführung der Wahl beteiligt sind. Und es gilt auch für den Abstimmenden selbst: Er darf Dritten gegenüber nicht belegen können, wie er abgestimmt hat, da sonst nicht auszuschließen ist, dass seine Stimmabgabe durch Anreize oder Drohungen beeinflusst wird.
- Manipulationssicherheit Es muss sichergestellt sein, dass jede abgegebene Stimme ins Abstimmungsergebnis einfließt. Ein nachträgliches Ändern oder Unterdrücken von Stimmen muss ebenso ausgeschlossen sein wie ein unbemerktes Hinzufügen zusätzlicher Stimmen.
- Nachprüfbarkeit Wenn es Anhaltspunkte für Fehler gibt, muss es möglich sein, die Richtigkeit des Abstimmungsergebnisses zu überprüfen.
- Zugänglichkeit und Stabilität Die Abstimmung muss für alle stimmberechtigten Personen gleichermaßen zugänglich sein, also auch für Menschen mit körperlichen Beeinträchtigungen. Und es muss weitestgehend sichergestellt sein, dass der Abstimmungsverlauf nicht durch Angriffe von außen schwerwiegend beeinträchtigt werden kann.
- Transparenz Jeder Beteiligte muss die Möglichkeit haben, alle wesentlichen Schritte der Abstimmung zu beobachten und ohne besondere Sachkenntnis zu verstehen.

Es gibt kein digitales oder analoges Abstimmungsverfahren, dass all diese Anforderungen vollständig erfüllt, und auch ParTCP erhebt keinen solchen Anspruch. Das Ziel ist eher ein Relatives, nämlich in jedem Punkt mindestens genauso gut, möglichst aber besser abzuschneiden als eine herkömmliche Briefwahl.

2.2 Bausteine

Das Wahlprotokoll, das in ParTCP umgesetzt wird, beruht auf einem pragmatischen Ansatz. Die Innovation liegt nicht in ausgeklügelten kryptografischen Verfahren, sondern in einer neuartigen Kombination bewährter Bausteine, insbesondere asymmetrischer Verschlüsselung, Einwegverschlüsselung (Hashing) und digitaler Signaturen. Um das Wahlprotokoll nachvollziehen zu können, ist ein allgemeines Verständnis dieser Komponenten erforderlich, darum werden diese im Folgenden kurz dargestellt. Zu beachten ist, dass es sich um vereinfachte bzw. idealisierte Darstellungen handelt, die lediglich die Grundprinzipien deutlich machen sollen.

In den folgenden Abschnitten bedeutet die Formulierung „es ist nicht möglich“, dass es nach dem Stand der Technik eines unverhältnismäßig hohen Aufwandes an Zeit und Rechnerkapazitäten bedürfte, um den betreffenden Sachverhalt zu ermöglichen.

2.2.1 Asymmetrische Verschlüsselung

Unter Verschlüsselung versteht man allgemein ein Verfahren, bei dem eine Nachricht („Klartext“) mit Hilfe eines geheimen Schlüssels so verändert („verschlüsselt“) wird, dass sie nicht mehr lesbar ist („Geheimtext“). Der verschlüsselte Inhalt kann nur unter Zuhilfename des Schlüssels wiedergewonnen („entschlüsselt“) werden.

Lange Zeit hindurch kannte die Menschheit nur sogenannte „symmetrische“ Verschlüsselungsverfahren, bei denen für das Ver- und Entschlüsseln derselbe Schlüssel verwendet wird. Erst in neuerer Zeit wurden „asymmetrische“ Verfahren entwickelt, bei denen zwei verschiedene Schlüssel zum Einsatz kommen: einer für das Ver- und einer für das Entschlüsseln. Diese Verfahren umgehen das Problem, das mit jeder symmetrischen Verschlüsselung verbunden ist, nämlich dass man erst einmal einen sicheren Übertragungsweg braucht, um den geheimen Schlüssel zwischen den Beteiligten auszutauschen. Hinzu kommt, dass die geheimen Schlüssel notgedrungen mehreren Personen bekannt sein müssen, was die Gefahr des Aufdeckens erhöht.

Bei der asymmetrischen Verschlüsselung verbleibt einer der beiden Schlüssel bei der Person, die das Schlüsselpaar erstellt hat. Dieser „private“ Schlüssel (private key), manchmal auch „geheimer“ Schlüssel (secret key) genannt, wird niemals mit anderen Personen ausgetauscht. Der zweite Schlüssel kann dagegen völlig offen und ungeschützt verteilt werden, weshalb er auch als der „öffentliche“ Schlüssel (public key) bezeichnet wird. Wer eine vertrauliche Nachricht senden will, braucht diese nur mit dem öffentlichen Schlüssel der Zielperson zu verschlüsseln und kann sicher sein, dass niemand außer dieser Person den Inhalt lesen kann, da nur sie den für das Entschlüsseln nötigen geheimen Schlüssel besitzt.

Asymmetrische Verschlüsselungsverfahren gelten als sicher, sind weit verbreitet und spielen eine zentrale Rolle in der Informationstechnik, insbesondere beim Online-Banking und beim E-Commerce. Alle modernen Betriebssysteme liefern die Programmbibliotheken mit, die Anwendungsprogramme brauchen, um asymmetrische Verschlüsselungsverfahren nutzen zu können. Für Webbrowser, E-Mail-Clients und viele andere Programme ist es seit vielen Jahren selbstverständlich, ihren Anwendern solche Verfahren zugänglich zu machen. Das kleine Schlosssymbol im Fenster des Webbrowsers, das auf eine (asymmetrisch) verschlüsselte Verbindung hinweist, ist für die meisten Internetbenutzer ein vertrauter Anblick.

Ein asymmetrisches Verschlüsselungsverfahren muss folgende Eigenschaften aufweisen, wenn es für das ParTCP-Wahlprotokoll einsetzbar sein soll:

- Eine Nachricht, die mit dem öffentlichen Schlüssel eines Schlüsselpaars verschlüsselt wurde, lässt sich mit dem zugehörigen privaten Schlüssel entschlüsseln.
- Es ist nicht möglich, eine Nachricht, die mit dem öffentlichen Schlüssel eines Schlüsselpaars verschlüsselt wurde, ohne Zugriff auf den zugehörigen privaten Schlüssel zu entschlüsseln.

- Es ist nicht möglich, aus einem öffentlichen Schlüssel den zugehörigen privaten Schlüssel zu ermitteln.

2.2.2 Einwegverschlüsselung (Hashing)

Neben der asymmetrischen Verschlüsselung spielt die sogenannte Einwegverschlüsselung (auch als Hashing oder Hashfunktion bezeichnet) eine wichtige Rolle im ParTCP-Wahlprotokoll. Hashverfahren sind in der Informationstechnik noch älter und weiter verbreitet als asymmetrische Verschlüsselungsverfahren. Sie erzeugen aus einer Nachricht mit Hilfe eines komplexen mathematischen Verfahrens einen sogenannten Streuwert (Hash), der überprüfbar ist in dem Sinne, dass dieselbe Nachricht immer zum selben Streuwert führt. Es ist aber nicht möglich, aus dem Streuwert wieder die Nachricht zu rekonstruieren.

Ein wichtiges Einsatzgebiet für die Einwegverschlüsselung ist das Speichern von Passwörtern in Datenbanken. Wenn sich eine Person bei einem Webserver registriert und dafür einen Benutzernamen und ein Passwort wählt, wird das Passwort auf dem Server nicht im Klartext gespeichert, sondern als Streuwert. Ein späterer Anmeldeversuch gelingt nur, wenn der Streuwert, der aus dem gerade eingegebenen Passwort errechnet wird, mit dem ursprünglich gespeicherten Streuwert übereinstimmt. Auf diese Weise ist ein Passwortvergleich möglich, ohne dass ein Administrator herausfinden kann, welches Passwort für eine bestimmte Person auf dem Server gespeichert ist.

Ein Einwegverschlüsselungsverfahren muss folgende Eigenschaften aufweisen, wenn es für das ParTCP-Wahlprotokoll einsetzbar sein soll:

- Aus einer bestimmten Nachricht wird stets derselbe Streuwert errechnet.
- Die Länge des Streuwerts ist stets gleich und unabhängig von der Länge der Nachricht.
- Es ist nicht möglich, aus dem Streuwert Rückschlüsse auf den Inhalt der Nachricht zu ziehen.
- Es ist extrem unwahrscheinlich, dass bei zwei verschiedenen Nachrichten derselben Streuwert errechnet wird und somit praktisch unmöglich, zwei unterschiedliche Nachrichten zu finden, die denselben Streuwert ergeben.

Eine wichtige Rolle im ParTCP-Konzept spielen sogenannnte *private Streuwerte*, zu deren Erzeugung neben der eigentlichen Nachricht auch der private Schlüssel des Servers herangezogen wird. Diese Streuwerte sind von Außenstehenden nicht nachvollziehbar, das heißt, sie können selbst dann, wenn der Inhalt einer Nachricht bekannt ist, nicht herausfinden, welcher Streuwert zu dieser Nachricht gehört.

2.2.3 Digitale Signaturen

Digitale Signaturverfahren arbeiten wie asymmetrische Verschlüsselungsverfahren mit Schlüsselpaaren. Auch hier besitzt jede Person einen privaten und einen öffentlichen Schlüssel, aber diese werden nicht benutzt, um eine Nachricht zu ver- und entschlüsseln, sondern um den Streuwert einer Nachricht (die „digitale Signatur“) zu erzeugen und zu verifizieren. Der Sender der Nachricht erzeugt mit Hilfe seines geheimen Schlüssels die digitale Signatur, und der Empfänger kann mit Hilfe des zugehörigen öffentlichen Schlüssels eine Verfizierung durchführen, die zweifelsfrei feststellt, ob die Signatur tatsächlich von dem Sender stammt, oder nicht. Diese Überprüfung schlägt fehl, wenn eine korrekt signierte Nachricht im Nachhinein inhaltlich verändert wurde, so dass das digitale Signieren auch einen Schutz vor Datenmanipulationen darstellt.

Digitale Signaturen, auch elektronische Unterschriften genannt, sind in der Informationstechnik ähnlich lange bekannt und ähnlich verbreitet wie asymmetrische Verschlüsselungsverfahren. Sie dienen insbesondere dazu, die Urheberschaft von E-Mails und Dateien zu dokumentieren bzw. überprüfbar zu machen. Mit ihrer Hilfe lässt sich zum Beispiel feststellen, ob eine bestimmte Software, die man sich aus dem Internet heruntergeladen hat, aus einer vertrauenswürdigen Quelle stammt.

Ein digitales Signaturverfahren muss folgende Eigenschaften aufweisen, wenn es für das ParTCP-Wahlprotokoll einsetzbar sein soll:

- Eine Signatur, die mit einem bestimmten geheimen Schlüssel erzeugt wurde, lässt sich mit dem zugehörigen öffentlichen Schlüssel verifizieren.

- Es ist extrem unwahrscheinlich, dass für zwei verschiedene Nachrichten mit demselben privaten Schlüssel dieselbe Signatur erzeugt wird. Das heißt, dass eine Signatur nicht mehr erfolgreich verifiziert werden kann, wenn die ursprüngliche Nachricht verändert wurde.
- Es ist nicht möglich, aus einem öffentlichen Schlüssel den zugehörigen privaten Schlüssel zu ermitteln.
- Ohne Zugriff auf den zugehörigen privaten Schlüssel ist es nicht möglich, eine digitale Signatur zu erzeugen, die mit einem bestimmten öffentlichen Schlüssel verifizierbar ist.

2.2.4 Zufallszahlengenerator

Der vierte zentrale Baustein für das ParTCP-Wahlprotokoll ist ein Mechanismus für das Generieren von Zufallszahlen. Dieser wird an verschiedenen Stellen verwendet, um nicht-vorhersagbare Zeichenketten zu erzeugen. Entscheidend sind hier die folgenden Eigenschaften:

- Bei einer großen Zahl an Ziehungen müssen alle möglichen Werte gleich häufig auftreten (Gleichverteilung).
- Nach der Ziehung eines bestimmten Werts muss der darauffolgende Wert ebenfalls gleich häufig verteilt vorkommen (Unvorhersagbarkeit).

2.3 Nicht-geheime Abstimmungen

Um den Ablauf einer geheimen Abstimmung zu verstehen, ist es hilfreich, sich zunächst vor Augen zu führen, wie eine nicht-geheime (namentliche) Abstimmung mit den oben beschriebenen Komponenten aussehen könnte:

1. Für jeden Teilnehmer wird auf dem Schlüsselserver ein Ordner mit einer eindeutigen Kennung (Teilnehmerkennung) angelegt. Diese Kennung wird zusammen mit einem zufällig erzeugten Berechtigungscode an den Teilnehmer gesendet. Der Streuwert des Berechtigungscodes wird im Teilnehmerordner gespeichert.
2. Der Teilnehmer erzeugt ein Schlüsselpaar und sendet seinen öffentlichen Schlüssel zusammen mit seiner Teilnehmerkennung und seinem Berechtigungscode als signierte Nachricht an den Schlüsselserver. Der Server prüft a) anhand der Signatur, ob der Absender tatsächlich im Besitz des privaten Schlüssels ist, der zu dem öffentlichen Schlüssel gehört, und b) anhand des hinterlegten Streuwerts, ob es sich um den korrekten Berechtigungscode handelt. Wenn beides zutrifft, wird der Schlüssel im Teilnehmerordner abgespeichert und der Berechtigungscode ungültig gemacht, damit keine weitere Schlüsselhinterlegung möglich ist.
3. Steht eine Abstimmung an, füllt der Teilnehmer den digitalen Stimmzettel aus, signiert diesen mit Hilfe seines privaten Schlüssels und sendet ihn an den Abstimmungsserver. Dieser überprüft anhand der Teilnehmerkennung, ob es sich um einen stimmberechtigten Teilnehmer handelt. Außerdem verifiziert er die Signatur anhand des öffentlichen Schlüssels, der auf dem Schlüsselserver hinterlegt ist.
4. Nachdem alle Teilnehmer ihre digitalen Stimmzettel abgegeben haben, werden die Stimmen ausgezählt und die Ergebnisse veröffentlicht.

Durch den Einsatz der Teilnehmerkennungen und Signaturen lassen sich gefälschte Stimmzettel ohne weiteres erkennen und aussondern. Und da es sich um gläserne Server handelt, kann jeder Beteiligte jeden abgegebenen Stimmzettel auf seine Richtigkeit überprüfen.

2.4 Geheime Abstimmungen

In den meisten Fällen darf nicht nachvollziehbar sein, welcher Teilnehmer wie abgestimmt hat, darum ist das eben beschriebene Verfahren nicht praxistauglich. Aber es ist lediglich ein zusätzlichen Schritt erforderlich, um geheime Abstimmungen zu ermöglichen. Dieser zusätzliche Schritt besteht darin, dass für jede Abstimmung ein Pool an anonymen Teilnehmerkennungen erzeugt und jedem Stimmberechtigten nach dem Zufallsprinzip eine dieser Pseudo-Identitäten zugelost wird. Alle weiteren Schritte laufen dann so ab wie oben beschrieben, das heißt, der Teilnehmer hinterlegt (incognito) seinen öffentlichen Schlüssel und sendet seinen digitalen signierten Stimmzettel (ebenfalls incognito, aber unverschlüsselt) an den Abstimmungsserver. Der Abstimmungsserver kann anhand der Teilnehmerkennung feststellen, dass es sich um einen stimmberechtigten Teilnehmer handelt, und er kann auch die Signatur anhand des öffentlichen Schlüssels verifizieren, aber er hat keine Möglichkeit festzustellen, welche Person sich hinter dieser anonymen Identität verbirgt.

Damit dieses Verfahren die weiter obene beschriebenen Anforderungen erfüllt, müssen zwei Voraussetzungen erfüllt sein:

1. Es darf keine Möglichkeit geben herauszufinden, welchem Teilnehmer welche anonyme Abstimmungskennung zugelost wurde.
2. Es muss sichergestellt sein, dass nur stimmberechtigte Teilnehmer eine anonyme Abstimmungskennung erhalten.

Damit die Zuordnung einer anonymen Abstimmungskennung zu einem bestimmten Teilnehmer nicht nachvollziehbar ist, wird jede dieser Kennungen mit einer zufällig erzeugten „Losnummer“ verknüpft, die an den Abstimmungsleiter ausgehändigt, aber auf dem Server nur als privater Streuwert gespeichert wird. Das heißt, der Server kann zu einer bestimmten Losnummer die zugehörige anonyme Abstimmungskennung ermitteln, aber niemand sonst.

Damit serverseitig keine Verbinung zwischen einem bestimmten Mitglied und einer Losnummer möglich ist, findet die Verteilung mit Hilfe eines externen Systems statt, das autonom arbeitet. Der Server liefert lediglich eine Liste mit den Losnummern aus, und der Versammlungsleiter verteilt diese nach dem Zufallsprinzip unter den Teilnehmern (siehe „Übermittlung der Losnummern“ im Abschnitt „Wahlprotokoll > Vorbereitung einer Veranstaltung“).

KAPITEL 3

Nachrichten

Im ParTCP-Kontext bezeichnet der Begriff „Nachricht“ ein serialisiertes Datenobjekt im YAML-Format, das zwischen Client und Server ausgetauscht wird. Die Attribute dieses Datenobjekts werden als „Nachrichtenelemente“ oder kurz „Elemente“ bezeichnet. Damit ein YAML-String als Nachricht angesehen wird, muss er ein Element mit dem Namen *Message-Type* enthalten, das angibt, um welche Art von Nachricht es sich handelt. Hiervon hängt ab, welche weitere Elemente erwartet werden und wie die Nachricht verarbeitet wird.

Der Name eines Nachrichtenelements wird stets mit großen Anfangsbuchstaben geschrieben. Besteht er aus mehreren Wörtern, werden diese durch Bindestriche miteinander verbunden und jeweils mit großen Anfangsbuchstaben geschrieben. Enthält ein Element ein Datenobjekt, werden dessen Attribute in Kleinbuchstaben und mit Unterstrichen als Worttrenner geschrieben.

Bestimmte Nachrichtenelemente sind unabhängig vom Nachrichtentyp für bestimmte Zwecke reserviert:

Message-Type Typ der Nachricht

To Adressat (Empfänger) der Nachricht

From Absender der Nachricht

Signature digitale Signatur der Nachricht

Date Zeitstempel der Nachricht

Public-Key universeller öffentlicher Schlüssel des Absenders

Original-Message die ursprüngliche Nachricht als String (bei Antworten)

Ist der Inhalt eines Elements verschlüsselt, wird dem Namen eine Tilde (~) angehängt. Enthält eine Nachricht dasselbe Element in verschlüsselter und unverschlüsselter Form, wird das unverschlüsselte Element ignoriert, es sei denn, das verschlüsselte Element ist nicht entschlüsselbar.

Eine Nachricht, die ein Client an einen Server sendet, wird von diesem wiederum mit einer Nachricht beantwortet, wobei die ursprüngliche Nachricht im Normalfall in die Antwort eingebunden wird (Element *Original-Message*). Hat die ursprüngliche Nachricht die Änderung eines Datenobjekts ausgelöst, wird das neue Objekt vollständig in die Antwort eingebunden, und die Antwort wird auf dem Server abgelegt.

Nachrichten werden grundsätzlich in UTF-8-Kodierung erstellt. Zeilenumbrüche müssen durch Linefeed-Zeichen (LF, ASCII-Wert 10) repräsentiert werden. Carriage-Return-Zeichen (CR, ASCII-Wert 13) sind nicht zulässig, auch nicht

in der Kombination CR+LF.

KAPITEL 4

Vertrauenswürdige Maschinen

Das ParTCP-Sicherheitskonzept setzt voraus, dass es sich bei den eingesetzten Servern um „vertrauenswürdige Maschinen“ handelt. Vertrauenswürdig bedeutet in diesem Zusammenhang:

1. Die Maschine muss für alle Mitglieder einer Gemeinschaft transparent („gläsern“) sein. Das heißt, jedes Mitglied muss jederzeit in der Lage sein, das gesamte Dateisystem in Augenschein zu nehmen, um sich selbst ein Bild von der installierten Software und den auf der Maschine gespeicherten Daten zu machen.
2. Außer über das Absetzen von Nachrichten darf es keine Möglichkeit geben, Änderungen am Software- oder Datenbestand der Maschine vorzunehmen. Das heißt, es müssen hard- und softwareseitig Vorkehrungen getroffen werden, um Rootzugriffe und Manipulationen am Dateisystem zu verhindern. Dies wird als „Versiegeln“ oder „Abschotten“ bezeichnet.
3. Ein besonders hohes Maß an Vertrauenswürdigkeit lässt sich erreichen, wenn alle relevanten Daten auf einem WORM-Laufwerk (write once, read multiple) gespeichert werden, das heißt auf einem Laufwerk, das physikalisch nur das Erstellen von Dateien und Verzeichnissen erlaubt, aber nicht das nachträgliche Ändern. Leider weisen solche Laufwerke in der Regel nur geringe Schreibgeschwindigkeiten auf, so dass sie zwar für Archivierungszwecke, aber nicht für den Echtzeiteinsatz geeignet sind. Ungeachtet dessen bildet der Grundsatz, dass einmal geschriebene Daten nicht mehr verändert werden, einen Kern des ParTCP-Konzepts, so dass künftige technische Fortschritte bei den Speichermedien unmittelbar genutzt werden können.

4.1 Transparenz

Die Forderung nach Transparenz wird durch Einhaltung der folgenden Bedingungen erfüllt:

- Der Server wird mit einem quelloffenen Betriebssystem ausgestattet, dessen Quellcode unter einer öffentlich zugänglichen Versionsverwaltung steht. Der installierte Systemstand muss jederzeit mit offiziellen Paketquellen abgeglichen werden können.
- Das Dateisystem des Servers kann von jedem Mitglied der Gemeinschaft in Augenschein genommen werden, z. B. durch einen SSH-Zugang, der Lesezugriff auf alle Dateien und Verzeichnisse bietet. Unzugänglich sind nur die privaten Schlüssel des Servers. Um eine übermäßige Belastung des Servers zu vermeiden, darf der Lesezugriff zeitlich und/oder mengenmäßig begrenzt werden.

- Alle für die demokratischen Entscheidungsprozesse relevanten Daten müssen in Form von Klartextdateien im Dateisystem abgelegt sein. Der Einsatz von binär-codierten Dateien, insbesondere von Datenbanktabellen, ist nur für Indizierungs- oder Caching-Zwecke zulässig, um Datenzugriffe zu beschleunigen. In diesem Fall müssen die Routinen, die aus den Klartextdateien die Tabellen erzeugen bzw. füllen, in Form von lesbaren Skripten vorliegen (siehe nächsten Punkt).
- Die Anwendungslogik muss durch Skripte festgelegt sein, die in einer gängigen Skriptsprache verfasst sind, unter öffentlich zugänglicher Versionsverwaltung stehen und auf definierte und dokumentierte Weise zum Server übertragen werden. Es muss jederzeit nachvollziehbar sein, welchem Versionsstand der Server entspricht und ob bzw. inwieweit ein Skript auf der Maschine gegenüber der Originalquelle verändert wurde.
- Programme und Bibliotheken, die ausschließlich als Maschinencode vorliegen und nicht zum Betriebssystem gehören, dürfen nur eingesetzt werden, wenn der zugehörige Quellcode veröffentlicht ist, unter Versionskontrolle steht und direkt auf der Maschine kompiliert werden kann. Der Kompilierungsvorgang ist in diesem Fall zu dokumentieren. Updates dürfen nur durch Aktualisieren und Rekompilieren des Quellcodes eingespielt werden.

4.2 Abschottung

Die Forderung nach Abschottung wird durch Einhaltung folgender Bedingungen erfüllt:

- Alle äußeren Schnittstellen, die für eine Dateneingabe in Frage kommen, aber nicht für den Nachrichtenaustausch benötigt werden, sind ausgebaut oder unbenutzbar gemacht. Dies betrifft insbesondere serielle Schnittstellen für den Anschluss von Tastaturen und Datenträgern, aber zum Beispiel auch Bluetooth-, WLAN- und HDMI-Adapter.
- Das Servergehäuse ist so zu verschließen und zu versiegeln, dass es sich nicht mehr öffnen lässt, ohne Spuren zu hinterlassen. Es muss jederzeit zweifelsfrei belegbar sein, dass das Innere des Gehäuses dem physischen Zustand entspricht, in dem es sich zum Zeitpunkt der Abschottung befunden hat.
- Das Betriebssystem ist in einen Zustand zu versetzen, der das Lesen der privaten Schlüssel und das Überschreiben relevanter Daten an der Rechteprüfung der ParTCP-Software vorbei unmöglich macht. Welche Schritte dazu im Einzelnen gehören, hängt vom Betriebssystem und von der ParTCP-Implementierung ab. Beispiele wären:
  - Sperrung aller Ports, außer denen, die für den Nachrichtenaustausch (z. B. HTTP) und die Lesezugriffe (z. B. SSH) benötigt werden
  - Deaktivieren des Root-Benutzerkontos
  - Deaktivieren des Bootloaders
  - Deaktivieren des Single-User-Modus‘
  - Löschen von Treibern, die das Hochfahren des Servers von einer lokalen Konsole erlauben
  - Sicherstellen, dass der Systembenutzer, in dessen Namen die ParTCP-Software ausgeführt wird (und der allein die Leserechte für die privaten Schlüssel sowie die Schreibrechte für das ParTCP-Datenverzeichnis besitzt), keine interaktiven Sitzungen starten oder zur Ausführung von Fremdprogrammen veranlasst werden kann

Die Abschottung muss von mindestens zwei, besser drei Personen durchgeführt und schriftlich dokumentiert werden. Die Dokumentation ist mit einer eidesstattlichen Versicherung der beteiligten Personen zu versehen, in der diese „nach bestem Wissen und Gewissen“ bescheinigen, dass es keine Möglichkeit mehr gibt, unbemerkt Änderungen an der Maschine vorzunehmen. Die Dokumentation ist so zu veröffentlichen, dass alle Mitglieder der Gemeinschaft darauf zugreifen können. Weitere Einzelheiten finden sich im Abschnitt *Abschottungsprozedur*.

KAPITEL 5

Kryptosystem

Herzstück der ParTCP-Plattform ist ein asymmetrisches Kryptosystem, das für das Signieren von Nachrichten und das Verschlüsseln von Nachrichtenelementen verwendet wird. ParTCP 1.0 spezifiziert folgende Verfahren für die Umsetzung dieses Kryptosystems:

- Signieren und Verifizieren mittels Ed25519 bzw. Curve25519 (RFC 7748)
- Schlüsselaustausch mittels X25519
- Verschlüsselung mittels AES-256 (CTR-Modus)

5.1 Universelle öffentliche Schlüssel

Um den Umgang mit Teilnehmeridentitäten zu erleichtern, setzt ParTCP das Vorhandensein von universell einsetzbaren öffentlichen Schlüsseln voraus. Das heißt, auch wenn das verwendete Kryptosystem (wie in ParTCP 1.0 der Fall) unterschiedliche Schlüsselpaare für das Signieren und das Verschlüsseln von Daten erfordert, tritt nach außen nur ein einzelner Schlüssel in Erscheinung.

Um aus den zwei öffentlichen Schlüsseln einen zu machen, wird in ParTCP 1.0 folgende (in Pseudocode ausgedrückte) Formel verwendet:

pubKeyUniversal = base64_encode(concat(pubKeySign, pubKeyCrypt))

Das heißt, die öffentlichen Schlüssel für das Signieren und Verschlüsseln werden als binäre Bytefolgen aneinandergehängt und dann mittels base64-Kodierung in eine ASCII-Zeichenfolge umgewandelt.

Ein Trennzeichen zwischen den beiden Schlüsseln ist nicht erforderlich, da die Längen der Schlüssel feststehen (32 Byte). Um aus dem zusammengefassten Schüssel die beidenTeilschlüssel zu ermitteln, muss die Bytefolge nach der base64-Dekodierung an der entsprechenden Byteposition getrennt werden.

5.2 Signaturen

Um eine elektronische Signatur in eine Nachricht einzubinden, wird die binäre Bytefolge mittels base64-Kodierung in eine ASCII-Zeichenfolge umgewandelt und dann als eigene Zeile mit dem Präfix *Signature:* am Anfang der Nachricht eingefügt.

Um eine signierte Nachricht zu verfizieren, wird die erste Zeile entfernt, um die ursprüngliche Nachricht zu erhalten. Aus der entfernen Zeile wird das Präfix entfernt und der übrige Teil mittels base64-Dekodierung in eine binäre Bytefolge umgewandelt.

Vor dem Erstellen und Verifizieren einer Signatur ist sicherzustellen, dass alle Zeilenumbrüche innerhalb der Nachricht durch einfache Linefeed-Zeichen (ASCII-Wert 10) repräsentiert werden und sich keine Carriage-Return-Zeichen (ASCII-Wert 13) in der Nachricht befinden.

5.3 Verschlüsselung

ParTCP setzt beim Umgang mit verschlüsselten Daten voraus, dass diese für sich selbst stehen und für die Entschlüsselung keine Informationen erforderlich sind außer der Absenderangabe bzw. des öffentlichen Schlüssels, der für die Verschlüsselung verwendet wurde. Da eine sichere AES-Verschlüsselung die Übermittlung eines zufällig erzeugten Initialisierungsvektors erfordert, definiert ParTCP 1.0 folgende Formel, um den Initialisierungsvektor in die verschlüsselten Daten einzubinden:

```
encryptedFinal = concat(
        base64_encode( iniVector ),
        ':',
        base64_encode( encryptedRaw )
)
```

Um die endgültige verschlüsselte Zeichenkette zu erzeugen, werden die binären Bytefolgen des Initialisierungsvektors und der rohen verschlüsselten Daten jeweils mittels base64-Kodierung in ASCII-Zeichenfolgen umgewandelt und dann mit einem Doppelpunkt als Trennzeichen aneinandergehängt.

KAPITEL 6

Wahlprotokoll

Das Wahlprotokoll ist in drei Phasen gegliedert: die Registrierungs-, die Anmelde- und die Abstimmungsphase. Die Registrierung ist pro Teilnehmer nur einmal erforderlich; die Anmeldephase wird einmal pro Veranstaltung durchgeführt, die Abstimmungsphase unter Umständen mehrmals pro Veranstaltung.

6.1 Registrierung

Bei einer Registrierung wird für ein bestimmtes Mitglied der Gemeinschaft eine offene Identität auf einem Schlüsselserver erstellt. Dieser Schritt ist für das Durchführen von Abstimmungen nicht zwingend erforderlich, da Teilnehmer an einer Abstimmung hierfür ohnehin mit einer temporären, anonymen Identität ausgestattet werden. Das dafür erforderliche Zustellen von Losnummern wird allerdings erleichtert, wenn die Teilnehmer über eine offene Identität verfügen (siehe weiter unten den Abschnitt „Übermittlung der Losnummern“ in „Vorbereitung einer Veranstaltung“).

Die Registrierung besteht aus zwei Schritten: dem Vergeben einer Teilnehmerkennung und dem Hinterlegen eines öffentlichen Schlüssels.

6.1.1 Teilnehmerkennung vergeben

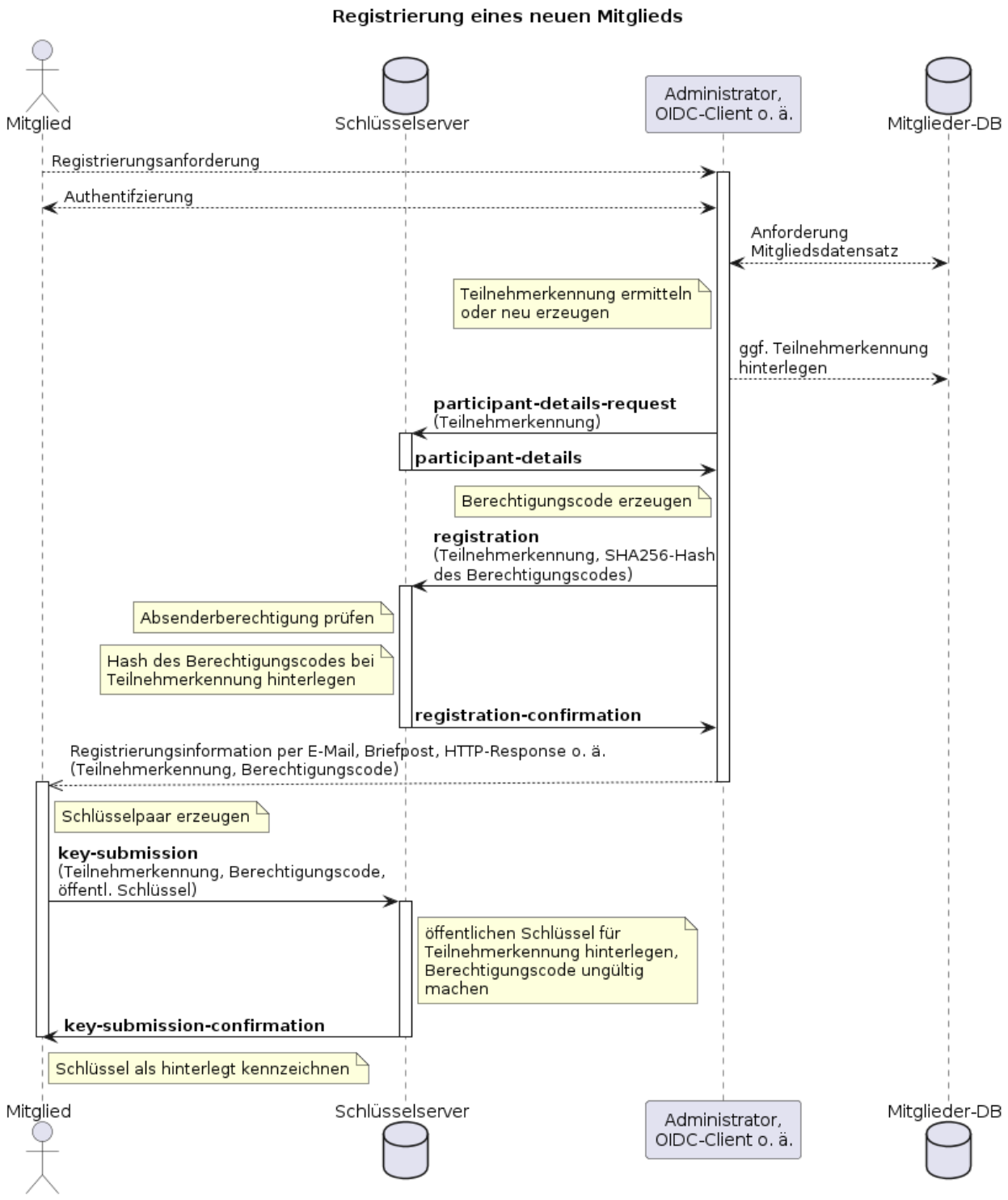
Das Vergeben der Teilnehmerkennung geschieht durch einen Administrator, der die Mitgliederliste einsehen kann und auf dem Schlüsselserver über die Berechtigung verfügt, neue Identitäten anzulegen. Dabei kann es sich um einen Menschen handeln oder um ein Computerprogramm, das auf einem zentralen Rechner läuft und ein maschinelles Authentifizierungsverfahren nutzt. In beiden Fällen muss es eine Regel geben, die festlegt, wie aus einem bestimmten Mitgliederdatensatz eine Teilnehmerkennung gebildet wird. Und diese Regel muss so gewählt sein, dass zu einem Datensatz stets dieselbe Kennung gebildet wird, auch wenn sich bestimmte Mitgliedsdaten zwischenzeitlich geändert haben. Es muss sichergestellt sein, dass ein Mitglied zu keinem Zeitpunkt mehr als eine Identität erhalten kann.

Wenn der Administrator eine Registrierung startet, erzeugt er nach dem Zufallsprinzip einen Berechtigungscode und hinterlegt dessen SHA256-Hash zusammen mit der Teilnehmerkennung auf dem Schlüsselserver. Teilnehmerkennung und Berechtigungscode werden an das betreffende Mitglied übermittelt, wobei durch die Art der Übermittlung und der vorgeschalteten Prüfung sichergestellt sein muss, dass diese Daten tatsächlich an das richtige Mitglied gelangen. Dies kann zum Beispiel durch einen Versand per Post an die im Mitgliederdatensatz hinterlegte Anschrift oder einen digitalen Versand an die E-Mail-Adresse des Mitglieds geschehen.

6.1.2 Öffentlichen Schlüssel hinterlegen

Nachdem das Mitglied seine Teilnehmerkennung und seinen Berechtigungscode erhalten hat, sendet es seinen öffentlichen Schlüssel zusammen mit Kennung und Code an den Schlüsselserver. Der überprüft, ob der Hash des übermittelten Berechtigungscodes mit dem Hash übereinstimmt, der für die betreffende Teilnehmerkennung hinterlegt ist, und speichert dann den Schlüssel zu der Teilnehmerkennung ab. Der Berechtigungscode verliert damit seine Gültigkeit.

Sollte irgendwann die Notwendigkeit bestehen, einen neuen öffentlichen Schlüssel zu hinterlegen, zum Beispiel weil der alte kompromittiert wurde oder der zugehörige private Schlüssel verlorengegangen ist, muss das Mitglied beim Administrator einen neuen Berechtigungscode anfordern. Bei der Zusendung ist wiederum sicherzustellen, dass der Code tatsächlich an das richtige Mitglied gelangt.




```
Message-Type: registration
From: admin01@partcp.example.org
To: partcp.example.org
Participant-Id: p003283
Credential: 6ca13d52ca70c883e0f0bb101e425a89e8624de51db2d2392593af6a84118090
```

```
Message-Type: registration-confirmation
From: partcp.example.org
To: admin01@partcp.example.org
Original-Message: |
  ...
Participant-Data:
  id: p003283
  credential: 6ca13d52ca70c883e0f0bb101e425a89e8624de51db2d2392593af6a84118090
```

```
Message-Type: key-submission
From: p003283@partcp.example.org
To: partcp.example.org
Credential~: "lFpidrMI5KMrFP7Vnb02Uw==:wZHfct/V"
Public-Key: lqLGtV1EtR8ClvIPAYyiWeeU3Ug9t9agY1jyjZWJxe6XPB3rnpZTKB67gctR4n4p...
```

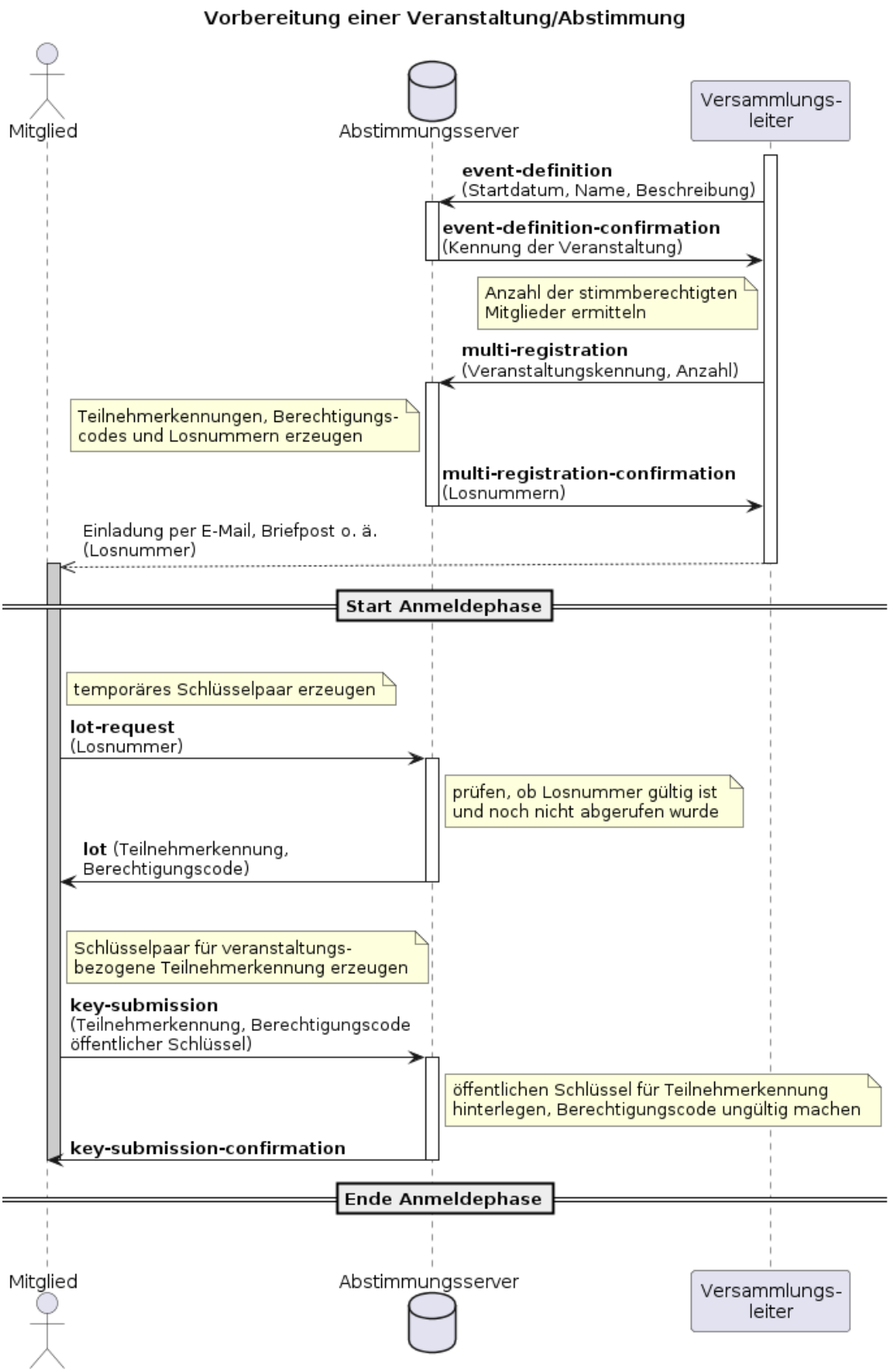
```
Message-Type: key-submission-confirmation
From: partcp.example.org
To: p003283@partcp.example.org
Original-Message: |
  ...
Participant-Data:
  id: p003283
  public_key: lqLGtV1EtR8ClvIPAYyiWeeU3Ug9t9agY1jyjZWJxe6XPB3rnpZikjTKB67IgXd...
```

6.2 Vorbereitung einer Veranstaltung

Um eine oder mehrere Abstimmungen durchzuführen, muss zunächst eine Veranstaltung definiert und der Teilnehmerkreis festgelegt werden. Außerdem müssen anonyme Identitäten erstellt und unter den Teilnehmer „verlost“ werden. Die Schritte im Einzelnen:

1. Der Veranstaltungsleiter legt Titel und Beschreibung der Veranstaltung fest und teilt dem Abstimmungsserver mit, wie viele Mitglieder stimmberechtigt sind.
2. Der Abstimmungsserver erzeugt die gewünschte Anzahl an Teilnehmerkennungen mit Berechtigungscodes und speichert diese verschlüsselt auf der Festplatte. Zu jeder Teilnehmerkennung wird nach dem Zufallsprinzip eine Losnummer erzeugt, deren SHA256-Hash ebenfalls auf dem Server gespeichert wird und als Index für den Zugriff auf Teilnehmerkennung und Berechtigungscode dient. Die erzeugten Losnummern werden in zufälliger Reihenfolge an den Veranstaltungsleiter zurückgeliefert.
3. Der Veranstaltungsleiter teilt jedem stimmberechtigten Mitglied eine Losnummer zu und sorgt dafür, dass sie dem betreffenden Mitglied – und nur diesem – zugesendet bzw. bereitgestellt wird. Wie dieser Schritt praktisch ausgestaltet werden kann, ist unten im Abschnitt „Übermittlung der Losnummern“ beschrieben. Sollten Losnummern übrig bleiben, werden diese vom Veranstaltungsleiter an den Abstimmungsserver übermittelt und für ungültig erklärt.
4. Der Veranstaltungsleiter erklärt die Anmeldephase für eröffnet (oder der vordefinierte Zeitpunkt des Anmeldestarts ist erreicht).
5. Der Abstimmungsserver blockiert den öffentlichen Lesezugriff auf das Verzeichnis, in dem die Veranstaltungsdaten gespeichert sind.
6. Das einzelne Mitglied ruft mit Hilfe seines Clients die Teilnehmerkennung und den Berechtigungscode ab, die auf dem Abstimmungsserver mit seiner Losnummer verknüpft sind. Falls mit der betreffenden Losnummer bereits zuvor ein Abruf erfolgte, wird die Anfrage zurückgewiesen. Abrufe werden ohne Zeitstempel protokolliert.

7. Der Client erzeugt ein Schlüsselpaar und überträgt den öffentlichen Schlüssel zusammen mit Teilnehmerkennung und Berechtigungscode zum Abstimmungsserver. Der Server prüft, ob der Hash des Berechtigungscodes mit dem gespeicherten Hash übereinstimmt und hinterlegt den öffentlichen Schlüssel dauerhaft für die betreffende Teilnehmerkennung. Der Berechtigungscode verliert damit seine Gültigkeit und ist nicht erneut verwendbar.
8. Der Veranstaltungsleiter erklärt die Anmeldephase für beendet (oder der vordefinierte Zeitpunkt des Anmeldeendes ist erreicht).
9. Der Abstimmungsserver gibt den öffentlichen Lesezugriff auf das Veranstaltungsverzeichnis wieder frei. Losnummerneinreichungen und Schlüsselhinterlegungen werden nun nicht mehr akzeptiert.


Vorbereitung einer Veranstaltung/Abstimmung
Mitglied
Abstimmungsserver
Versammlungs-
leiter
event-definition
(Startdatum, Name, Beschreibung)
event-definition-confirmation
(Kennung der Veranstaltung)
Anzahl der stimmberechtigten
Mitglieder ermitteln
multi-registration
(Veranstaltungskennung, Anzahl)
Teilnehmerkennungen, Berechtigungs-
codes und Losnummern erzeugen
multi-registration-confirmation
(Losnummern)
Einladung per E-Mail, Briefpost o. ä.
(Losnummer)
Start Anmeldephase
temporäres Schlüsselpaar erzeugen
lot-request
(Losnummer)
prüfen, ob Losnummer gültig ist
und noch nicht abgerufen wurde
lot (Teilnehmerkennung,
Berechtigungscode)
Schlüsselpaar für veranstaltungs-
bezogene Teilnehmerkennung erzeugen
key-submission
(Teilnehmerkennung, Berechtigungscode
öffentlicher Schlüssel)
öffentlichen Schlüssel für Teilnehmerkennung
hinterlegen, Berechtigungscode ungültig machen
key-submission-confirmation
Ende Anmeldephase
Mitglied
Abstimmungsserver
Versammlungs-
leiter

```
Message-Type: event-definition
From: admin@partcp.example.org
To: partcp.example.org
Event-Data:
  name: Bundesparteitag 2021
  date: 2021-11-27
  estimated_turnout: 10000
```

```
Message-Type: multi-registration
From: admin@partcp.example.org
To: partcp.example.org
Event-Id: 20211127-bundesparteitag-2021
Count: 7352
```

```
Message-Type: multi-registration-confirmation
From: partcp.example.org
To: admin@partcp.example.org
Original-Message: |
  ...
Lot-Codes~: >
  ntdXUPtFL3zWLJC7xFgfTw==:+g3cmDFlyNb1J50ANXoRo4XoJwAQ+0ShfYkjzm7cHnHSb/CdOFjyHAwCom...
```

```
Message-Type: lot-request
To: partcp.example.org
Event-Id: 20211127-bundesparteitag-2021
Lot-Code~: g3cmDFlyNb1J50ANXoR==:o4XoJwAQ+0ShfYkjzm7cHnHSb
Public-Key: qq/oN0xeBIluop53ezk47h3ax66ZGZHVckXK6hQ2zFQu6UDsurz7N5pfmFHoEwaICM1XfcDzN...
```

```
Message-Type: lot
From: partcp.example.org
Original-Message: |
  ...
Lot-Content~: >
  0ShfYkjzm7cHnHSbzUsa==:+b1J50ANXoRo4XoJwAQ+0ShfYkjzm7cHnHSb/CdOg3cmDFlyNFjyHAwCom...
```

6.2.1 Erzeugung der Losnummern

Eine Losnummer wird durch eine zufällige Auswahl von Buchstaben und Ziffern gebildet. Standardmäßig hat sie eine Länge von 16 Zeichen und besteht aus den Ziffern 1 bis 9 und den Großbuchstaben des lateinischen Alphabets, mit Ausnahme von „I“ und „O“. Die Ziffer 0 sowie die Buchstaben I und O bleiben unberücksichtigt, weil diese beim Lesen leicht verwechselt werden und daher zu Fehlern bei einer manuellen Übernahme in den Client führen können.

Nach diesen Standardvorgaben lassen sich 33^{16} oder $1{,}978 * 10^{24}$ verschiedene Losnummern bilden, so dass bei einer Veranstaltung mit einer Million Teilnehmern die Wahrscheinlichkeit, eine Nummer zu erraten, bei über $1 : 10^{18}$ liegt. Um bei kleineren Veranstaltungen mit kürzeren Losnummern arbeiten zu können, hat der Veranstaltungsleiter die Möglichkeit, eigene Vorgaben für die Nummernerzeugung festzulegen. Er kann die Gesamtlänge der Nummer und die zur Auswahl stehenden Zeichen vorgeben und, falls gewünscht, die Anzahl an Prüfziffern, die mit eingebunden werden sollen. Außerdem ist es möglich, ein Trennzeichen festzulegen, das nach einer bestimmten Anzahl von Zeichen in die Losnummer eingefügt wird, um die Lesbarkeit zu verbessern.

6.2.2 Übermittlung der Losnummern

Schritt 3 der Anmeldephase sieht vor, dass jedes stimmberechtigte Mitglied eine Losnummer zugeteilt und übermittelt bekommt. Wie dies geschieht, ist für das Wahlprotokoll unerheblich, solange sichergestellt ist, dass a) *jedes* stimmberechtigte Mitglied eine Losnummer erhält, b) kein *nicht*-stimmberechtigtes Mitglied eine Losnummer erhält und c) kein stimmberechtigtes Mitglied *mehr als eine* Losnummer erhält. Im Folgenden sollen verschiedene Wege vorgestellt werden, wie sich dies umsetzen lässt.

Ein einfacher Weg, der jedoch nur bei Präsenzveranstaltungen umsetzbar ist, besteht darin, die Losnummern auf Zettel zu drucken und jedes Mitglied nach Feststellung seiner Stimmberechtigung einen Zettel ziehen zu lassen. Mit einem zusätzlich aufgedruckten QR-Code kann die Übernahme der Losnummer in den Client erleichtert werden. Bei dieser Vorgehensweise müssen die Personen, die am Empfangstresen arbeiten, strikt darauf achten, dass jedes Mitglied tatsächlich nur ein Los zieht. Nach Abschluss der Teilnehmerregistrierung werden die übriggebliebenen Lose gezählt. Aus dieser Zahl wird die Zahl der ausgegebenen Lose ermittelt, und es wird geprüft, ob diese Zahl mit den auf der Teilnehmerliste als anwesend gekennzeichneten Mitgliedern übereinstimmt. Die übriggebliebenen Lose werden anschließend auf dem Abstimmungsserver für ungültig erklärt und vernichtet.

Bei Online-Veranstaltungen ist ein ähnlicher Weg gangbar, nur genügt es dann nicht, die Losnummern auf einfache Zettel zu drucken. Die Gefahr wäre zu groß, dass beim Eintüten und Versenden der Zettel Unbefugte Kenntnis der Losnummern erhalten. Daher ist in diesem Fall der Einsatz spezieller Sicherheitsdrucker erforderlich, bei denen die Losnummer durch eine abrubbelbare Schicht verdeckt oder im Innern eines fest verschlossenen Umschlags gedruckt wird. Dies erhöht die Sicherheit, aber auch den Kosten- und Zeitaufwand.

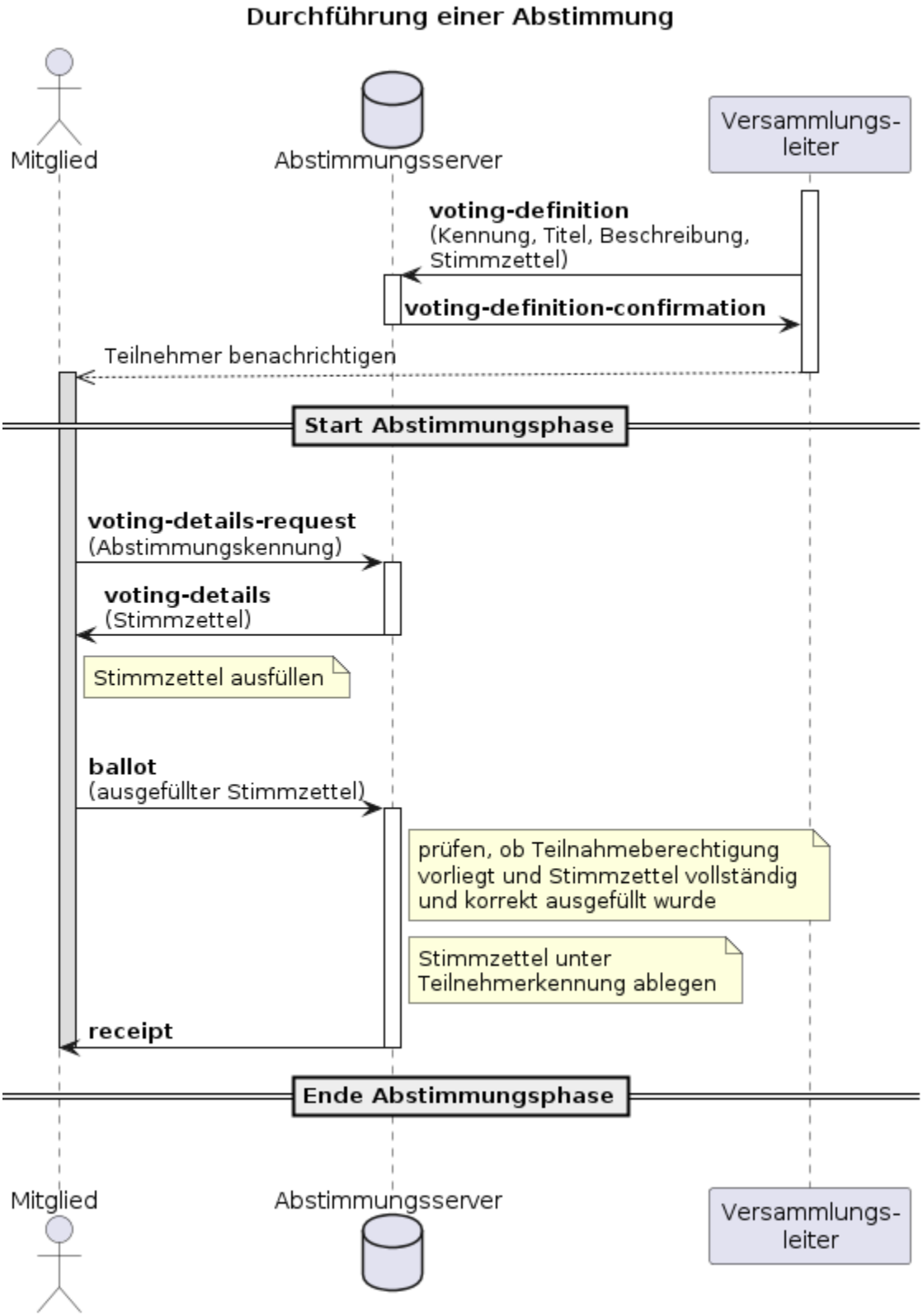
Ein dritter Weg, der sich für Präsenz- und Online-Veranstaltungen gleichermaßen eignet und der ebenso kostengünstig wie sicher ist, besteht darin, die Abstimmungsplattform selbst für die Übermittlung der Losnummern zu nutzen. Dies setzt voraus, dass sich alle Mitglieder zuvor auf der Plattform mit einer offenen Identität registriert haben (siehe oben den Abschnitt „Registrierung“). In diesem Fall kann der Veranstaltungsleiter für jedes stimmberechtigte Mitglied eine Losnummer mit dessen öffentlichem Schlüssel verschlüsseln und auf dem Abstimmungsserver hinterlegen. Der Client ruft die Losnummer ab und entschüsselt sie mit Hilfe des geheimen Schlüssels. Sie kann dann vom Client sofort für den Abruf der Teilnehmerdaten verwendet werden, ohne dass eine Nummer eingetippt oder ein Code gescannt werden muss.

6.3 Durchführen einer Abstimmung

Dies sind die Schritte, um im Rahmen einer Veranstaltung eine Abstimmung durchzuführen:

1. Der Veranstaltungsleiter übermittelt die Abstimmungsdetails einschließlich dem Musterstimmzettel an den Abstimmungsserver.
2. Der Veranstaltungsleiter erklärt die Abstimmung für eröffnet (oder der vordefinierte Startzeitpunkt wird erreicht).
3. Der Abstimmungsserver blockiert den öffentlichen Lesezugriff auf das Verzeichnis, in dem die Abstimmungsdaten gespeichert sind.
4. Das einzelne Mitglied ruft über seinen Client den Musterstimmzettel vom Abstimmungsserver ab und füllt ihn aus.
5. Der Client trägt die anonyme Teilnehmerkennung ein, siginiert das Ganze mit dem zugehörigen privaten Schlüssel und überträgt alles zum Abstimmungsserver.
6. Der Abstimmungsserver prüft, ob der Stimmzettel eine zugelassene Teilnehmerkennung trägt, ob die Signatur mit dem zu dieser Kennung hinterlegten öffentlichen Schlüssel verifiziert werden kann, ob der Stimmzettel vom Aufbau her dem hinterlegten Muster entspricht und ob alle Pflichtangaben vorhanden sind. Falls die Prüfung fehlschlägt, wird der Stimmzettel zurückgewiesen.
7. Der Abstimmungsserver speichert den Stimmzettel unter der betreffenden Teilnehmerkennung mit fortlaufender Nummer, aber ohne Zeitstempel, unverschlüsselt ab.

8. Der Veranstaltungsleiter erklärt die Abstimmung für beendet (oder der vordefinierte Endzeitpunkt wird erreicht).
9. Der Abstimmungsserver gibt den öffentlichen Lesezugriff auf das Abstimmungsverzeichnis wieder frei. Stimmzettel werden nun nicht mehr angenommen
10. Der Abstimmungsserver geht Teilnehmer für Teilnehmer durch, wertet jeweils den Stimmzettel mit der höchsten fortlaufenden Nummer aus und übernimmt die Bewertungen ins Gesamtergebnis. Das Gesamtergebnis wird bei den übrigen Abstimmungsdaten gespeichert.




```
Message-Type: voting-definition
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
From: admin@partcp.example.org
To: partcp.example.org
Event-Id: 20211127-bundesparteitag-2021
Voting-Data:
  name: satzaend-031
  title: Satzungsänderungsantrag Nr. 31
  description: ...
  type: consensus-10
  options:
    - { id: 1, name: Passivlösung (keine Änderung) }
    - { id: 2, name: Änderung wird vorgenommen }
```

```
Message-Type: voting-details-request
From: p000063.4@partcp.example.org
To: partcp.example.org
Event-Id: 20211127-bundesparteitag-2021
Voting-Id: satzaend-031
```

```
Message-Type: voting-details
From: partcp.example.org
To: p000063.4@partcp.example.org
Original-Message: >
  ...
Voting-Data:
  id: satzaend-031
  created_on: 2021-11-27 12:53:09
  created_by: admin@partcp.example.org
  modified_on: 2021-11-27 12:55:39
  modified_by: admin@partcp.example.org
  status: open
  name: satzaend-031
  title: Satzungsänderungsantrag Nr. 31
  type: consensus-10
  options:
    - { id: 1, name: Passivlösung (keine Änderung) }
    - { id: 2, name: Änderung wird vorgenommen }
```

```
Message-Type: ballot
From: p000063.4@partcp.example.org
To: partcp.example.org
Event-Id: 20211127-bundesparteitag-2021
Voting-Id: satzaend-031
Votes:
  - { id: 1, vote: 7 }
  - { id: 2, vote: 0 }
```

KAPITEL 7

DSGVO-Konformität

Da es eine eindeutige Verbindung zwischen einem Datensatz in der Mitgliederliste und einer ParTCP-Teilnehmerkennung gibt, müssen diese Kennungen als personenbezogene Daten im Sinne der DSGVO betrachtet werden. Daher müssen alle Teilnehmer über die Einzelheiten der Datenverarbeitung informiert werden und dieser Verarbeitung ausdrücklich zustimmen.

Das ParTCP-Konzept sieht die Möglichkeit vor, dass auf dem Schlüsselserver eine Datenschutzerklärung und der Wortlaut der Einwilligung hinterlegt wird, die von den Teilnehmern erwartet wird. Eine *registration*-Nachricht (siehe Abschnitt „Wahlprotokoll > Registrierung“) wird in diesem Fall nur dann verarbeitet, wenn sie das Attribut *Consent-Statement* enthält und der Wert dieses Attributs exakt mit dem Wortlaut der Einwilligung übereinstimmt, die auf dem Server hinterlegt ist.

Da alle *registration*-Nachrichten auf dem Server gespeichert bleiben, ist auf diese Weise für jeden Teilnehmer dauerhaft dokumentiert, dass er in die Verarbeitung seiner personenbezogenen Daten eingewilligt hat.

KAPITEL 8

Verzeichnisstruktur

Das folgende Listing zeigt die Verzeichnisstruktur einer beispielhaften Versammlung mit mehreren Abstimmungen. Die erste Abstimmung (*abstimmung_01*) ist bereits beendet. Man sieht, dass der Teilnehmer mit der Kennung *p000-2298* zweimal abgestimmt hat.

```
├── 20210915-example-event
│   ├── 20210915-150549-event-definition
│   ├── 20210918-130030-multi-registration
│   ├── lotcodes
│   │   ├── m20000138
│   │   ├── m20001581

...

│   ├── lots
│   │   ├── M2E0MGRDQ1I5NWEwOgwMjUzZGM2NzZkNDgwZ...ODhhMmQyM2Q1MTUyMzhmMDI3YjA1NTEzMw==
│   │   │   └── lot
│   │   ├── NzgyOWZTFiZTllNGFA0NTU2ZDgyOGFmMTIzN...NDJiNzcxN2U5ZDhmOThiZDEyYWJiYjMxMg==

...

│   ├── participants
│   │   ├── p000-0018
│   │   │   ├── 20210915-164133-registration
│   │   │   └── 20210916-093351-key-submission
│   │   ├── p000-0026

...

│   └── votings
│       ├── abstimmung_01
│       │   ├── 20210917-125309-voting-definition
│       │   ├── 20210918-103539-vote-count-request
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
│       │   └── ballots
│       │       ├── p000-2298
│       │       │   ├── ballot-0001
│       │       │   └── ballot-0002
│       │       ├── p000-3066

...

│       ├── abstimmung_02

...
```

KAPITEL 9

Abschottungsprozedur

Im Folgenden wird eine beispielhafte Prozedur beschrieben, mit der sich eine Maschine abschotten lässt, um als „vertrauenswürdig“ zu gelten.

9.1 Klärung der Verantwortung

Die Personen, die mit der Einrichtung und Abschottung einer Maschine betraut werden, müssen über entsprechende fachliche Qualifikationen verfügen und das Vertrauen der Gemeinschaft genießen, in deren Auftrag sie handeln. Die Einrichtung sollte von drei, muss aber mindestens von zwei Personen durchgeführt werden. Diese sollten aus einer größeren Gruppe von Personen, denen die Gemeinschaft per Abstimmung ihr Vertrauen ausgesprochen hat, per Losverfahren bestimmt werden.

Die beauftragten Personen müssen während der gesamten Dauer der nachfolgend beschriebenen Einrichtung anwesend sein und die im folgenden beschriebenen Schritte durchführen bzw. begleiten. Jede Person bestätigt am Ende durch eine eidesstattliche Versicherung, dass die Abschottung des Servers ordnungsgemäß erfolgte.

Neben den technisch verantwortlichen Personen sollte die Gemeinschaft mindestens eine weitere vertrauenswürdige Person als Beobachter bestimmen. Diese soll den ordnungsgemäßen Ablauf der Einrichtung überwachen und nach Möglichkeit auf Video dokumentieren.

9.2 Zufallsentscheide

Die nachfolgende Prozedur fordert an manchen Stellen, dass per Zufall oder Los eine von mehreren Möglichkeiten gewählt wird. Um dies umzusetzen, sollte jede beauftragte Person einen eigenen Würfel mitbringen. Das Losverfahren sieht dann folgendermaßen aus:

1. Die verfügbaren Möglichkeiten werden mit 0 (!) beginnend durchnummeriert.
2. Die Personen würfeln nacheinander so oft mit ihren jeweiligen Würfeln, wie es der Anzahl an Möglichkeiten entspricht.
3. Die Augenzahlen aller Würfe aller Personen werden aufaddiert.

4. Die Summe wird durch die Anzahl der Möglichkeiten dividiert; der dabei verbleibende Rest gibt an, welche Möglichkeit ausgelost wurde.

9.3 Auswahl und Aufstellen der Maschine

Wenn auf einer Maschine geheime Abstimmungen und Wahlen durchgeführt werden sollen, kommt für den Produktivbetrieb nur ein dedizierter Server in Frage, auf dem keine anderen Anwendungen laufen. Die Verwendung von virtuellen oder gemeinsam genutzten Servern ist lediglich für Entwicklungs- und Testzwecke zulässig.

Um auszuschließen, dass die Hardware oder das BIOS der Maschine manipuliert sind, muss es sich um ein handelsübliches Modell handeln, das fabrikneu und originalverpackt von den Einrichtern gemeinsam in Betrieb genommen wird. Idealerweise wird die Maschine aus einem Pool von mindestens drei gleichwertigen Modellen per Los ausgewählt.

Vor der Inbetriebnahme ist das Gehäuse zu öffnen und das Innenleben der Maschine in hoher Auflösung zu fotografieren. Die Fotos werden als Teil der Dokumentation archiviert.

Der Server muss über eine schnelle Internet-Anbindung und eine feste IP-Adresse verfügen, deshalb kommt als Aufstellort in der Regel nur ein Rechenzentrum in Frage. Dieses sollte sorgfältig ausgewählt werden und nach Möglichkeit über ein ISO-9001-zertifiziertes Qualitätsmanagement verfügen. Im Zweifelsfall muss es den Nachweis bringen können, dass sich während der gesamten Laufzeit niemand Unbefugtes physischen Zugang zu der Maschine verschaffen konnte.

9.4 Installation des Betriebssystems

Wenn die Maschine bereits ab Werk mit einem Betriebssystem ausgestattet wurde, ist die gesamte Festplatte zu löschen. Anschließend ist ein Betriebssystem zu installieren, das die technischen Voraussetzungen für den ParTCP-Einsatz erfüllt.

Die Installation wird nicht über das Netzwerk, sondern direkt an der Maschine von einem USB-Stick aus durchgeführt. Die Maschine ist dafür mit Monitor und Tastatur auszustatten, und zwar auf eine Weise, dass die Kabelverbindungen zwischen der Maschine und den angeschlossenen Geräten stets für alle beteiligten Personen auf einen Blick sichtbar sind. Per Losverfahren wird festgelegt, wer die Tastatur bedient. Die übrigen Einrichter stellen sich so auf, dass sie sehen können, was getippt und was auf dem Bildschirm angezeigt wird.

Die Einrichter haben sich davon zu überzeugen, dass die Maschine während der Installation noch nicht mit dem Netzwerk verbunden ist. Wenn im Laufe der Installationsprozedur das Rootpasswort für die Maschine einzugeben ist, tippt jeder Einrichter nacheinander eine nur ihm bekannte Zeichenfolge (mindestens acht Zeichen) ein. Dabei muss sichergestellt sein, dass keine andere Person und keine Überwachungskamera sehen kann, was getippt wird.

Der für die Installation verwendete USB-Stick muss von einem vertrauenswürigen Distributor produziert und originalverpackt angeliefert worden sein. Im Idealfall wird aus einem Pool von mindestens drei Sticks, die aus unterschiedlichen, aber durchweg vertrauenswürdigen Quellen stammen, einer ausgelost.

Die Einrichter ermitteln und notieren sich die Seriennummern der wichtigsten Hardware-Komponenten des Servers: Hauptplatine, Festplatte(n) und Netzwerkadapter.

Das Betriebssystem wird nach der Installation so angepasst, dass Wartungsupdates regelmäßig automatisch durchgeführt werden. Das heißt, es wird ein Prozess eingerichtet, der in bestimmten Zeitabständen eine nicht-interaktive Updateroutine mit Rootrechten ausführt. Ferner ist sicherzustellen, dass die üblichen Aufräumskripte des Systems automatisch ausgeführt werden.

Es wird ein Benutzerkonto mit dem Namen „public“ und dem Passwort „public“ angelegt und sichergestellt, dass über dieses Konto ein SSH-Zugang und ein Lesezugriff auf das gesamte Dateisystem möglich ist. Dieses Benutzerkonto darf über kein Home-Verzeichnis verfügen und auch sonst kein Schreibrecht für irgendein Verzeichnis haben.

9.5 Installation und Test der Anwendung

Nachdem das Betriebssystem installiert ist, werden Bildschirm und Tastatur entfernt, der Server wird mit dem Netzwerk verbunden und neu gestartet. Die weiteren Schritte erfolgen von einem virtuellen Terminal aus, wobei auch hier wieder ausgelost wird, wer die Tastatur bedient und wer beobachtet. Die Maschine muss bis zum Ende der Abschottung weiterhin im Sichtfeld der Einrichter bleiben.

Die Verbindung zum Server wird im Namen des Rootbenutzers mittels SSH hergestellt, wobei die Passworteingabe wieder geteilt erfolgt. Die Teilung stellt nicht nur sicher, dass keine Einzelperson mit Rootrechten auf den Server zugreifen kann. Das erfolgreiche Einloggen ist auch zugleich die Bestätigung dafür, dass die Anmeldung am „richtigen", das heißt an dem zuvor eingerichteten Server erfolgt. Als zusätzliche Absicherung werden auch noch einmal die Seriennummern der Hardware-Komponenten mit den zuvor notierten verglichen.

Durch entsprechende Shell-Kommandos werden die Anwendungen installiert, die für den Einsatz der ParTCP-Software erforderlich sind, insbesondere Apache/PHP und Git. Anschließend erfolgt die Installation der ParTCP-Software gemäß Dokumentation, das heißt durch Clonen aus dem Original-Repository. Die ParTCP-Konfigurationsdatei ist so anzupassen, dass Sie den Gegebenheiten auf der Maschine und den Wünschen der Gemeinschaft entspricht. Der Apache-Server ist so zu konfigurieren, dass er nur Dateien aus den vorgesehenen Verzeichnissen ausliefert, und die Inhalte dieser Verzeichnisse dürfen ohne Rootrechte nicht mehr veränderbar sein.

Nach der Installation sendet jeder Einrichter von einem persönlichen Rechner aus einige ParTCP-Ping-Nachrichten an den Server und testet dabei insbesondere auch die kryptografischen Funktionen. Anschließend wird das ParTCP-Schlüsselverzeichnis geleert (damit beim nächsten Aufruf neue Schlüssel erzeugt werden). Unmittelbar danach wird die Netzwerkverbindung getrennt.

9.6 Abschottung und Versiegelung

Nun wird der Server abgeschottet und versiegelt. Abschotten heißt, dass Zugriffe mit Rootrechten unterbunden werden, selbst für den Fall, dass jemand physischen Zugang zu der Maschine hat. Wie dies im Einzelnen geschieht, hängt vom verwendeten Betriebssystem ab und liegt in der Verantwortung der Einrichter. Die folgenden Schritte sind lediglich als Anregung zu verstehen:

- persönliche Benutzerkonten entfernen, sofern vorhanden
- sicherstellen, dass kein Benutzer Mitglied der *sudo*-Gruppe ist
- Root-Passwort entfernen bzw. durch ein Zufallspasswort ersetzen[1]
- Booten in Single-User-Mode durch Vergabe eines Zufallspassworts o. ä. unmöglich machen[2]
- Konsolenzugang sperren: physisch durch Unbrauchbarmachen der Schnittstellen (Sekundenkleber) und/oder softwartechnisch durch Löschen der erforderlichen Gerätetreiber

Nach dem Abschotten ist das Servergehäuse so mit Sicherheitsaufklebern und Plomben zu versehen, dass es nicht mehr möglich ist, das Gehäuse zu öffnen oder die unbrauchbar gemachten Schnittstellen zu nutzen, ohne irreversible Spuren zu hinterlassen.

[1]

https://www.tecmint.com/disable-root-login-in-linux
https://www.linuxcloudvps.com/blog/how-to-enable-and-disable-root-login-in-ubuntu
https://www.linuxfordevices.com/tutorials/linux/enable-disable-root-login-in-linux

[2]

https://www.techrepublic.com/article/how-to-password-protect-the-grub-boot-loader-in-ubuntu
https://kifarunix.com/how-to-protect-single-user-mode-with-password-in-ubuntu-18-04
https://www.linuxquestions.org/questions/linux-software-2/howto-disable-single-user-mode-544529

Die Maschine wird an ihren endgültigen Aufstellort gebracht, angeschlossen und hochgefahren. Jeder Einrichter sendet von einem persönlichen Rechner aus eine ParTCP-Ping-Nachricht an den Server und notiert sich den zurückgelieferten Public-Key. Anschließend loggt er sich mit dem öffentlichen SSH-Zugang auf dem Server ein, vergleicht noch einmal die Hardware-Seriennummern und prüft, ob der auf der Festplatte abgelegte Public-Key mit dem ausgelieferten übereinstimmt.

9.7 Bericht

Jeder Einrichter fertigt einen Bericht an, in dem er die durchgeführten Schritte im Detail dokumentiert. In dem Bericht sind die Hardware-Seriennummern und der Aufstellort der Maschine zu notieren, ferner die Sicherheitscodes der Siegelaufkleber und Plomben, die IP-Adresse, unter der die Maschine erreichbar ist, der Public-Key sowie Name und Passwort des öffentlichen SSH-Zugangs und das Credential, mit dem sich der Hauptadministrator auf dem Server registrieren kann. Auf der letzten Seite ist an Eides statt zu versichern, dass die Angaben in dem Bericht vollständig sind und der Wahrheit entsprechen und dass weitere Rootzugriffe auf den Server nach bestem Wissen und Gewissen ausgeschlossen sind.

Die Berichte werden dem Auftraggeber ausgehändigt, und jeder Einrichter legt eine Kopie seines eigenen Berichts dauerhaft in einem persönlichen Archiv ab. Der Auftraggeber veröffentlicht den Bericht auf eine Weise, die allen, auch künftig dazukommenden Mitgliedern der Gemeinschaft einen Zugriff auf den Inhalt erlaubt.

KAPITEL 10

Verzeichnisse und Tabellen

- genindex
- modindex
- search