UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR
THEORETISCHE INFORMATIK

Aus dem Institut für Theoretische Informatik
der Universität zu Lübeck
Direktor: Prof. Dr. math. K. Rüdiger Reischuk

Sicherheit und Anonymität in der Vorratsdatenspeicherung

Inauguraldissertation
zur
Erlangung der Doktorwürde
der Universität zu Lübeck
– Aus den Sektionen Informatik/Technik und Naturwissenschaften –

vorgelegt von
Dipl.-Inf. Markus Hinkelmann
aus Neumünster

Lübeck 2011

1. Berichterstatter:	PD Dr. rer. nat. Andreas Jakoby
2. Berichterstatter:	Prof. Dr. rer. nat. habil. Andreas Brandstädt
Tag der mündlichen Prüfung:	25. Februar 2011
Zum Druck genehmigt. Lübeck, den:	3. November 2011

Zusammenfassung

Heutzutage werden an und durch viele Stellen persönliche Daten gesammelt und gespeichert. Jeder Besuch auf einer Webseite kann über die Netzwerkadresse protokolliert werden, jeder Eintrag in einem sozialen Netzwerk kann mit anderen Daten vernetzt werden, jeder Einkauf mit einer digitalen Kundenkarte kann ein Kundenprofil erweitern. Durch die Leistungsfähigkeit aktueller Computersysteme können die gesammelten Daten effizient gefiltert, verknüpft und analysiert werden.

Das Sammeln von Daten erfolgt häufig aus guten Gründen. Da sich auch Kriminelle im Internet bewegen, setzen Sicherheitsbehörden auf die Untersuchung von gespeicherten Kommunikationsdaten. Durch Kundenprofile können Angebote aufeinander abgestimmt und optimiert werden. Zudem können die Benutzer, deren Daten gespeichert werden, von Vergünstigungen und Rabatten profitieren. Die Benutzer haben bei der Vorratsdatenspeicherung jedoch auch ein gegenläufiges Interesse: den Schutz der persönlichen Daten. Der Missbrauch und der Diebstahl von Vorratsdaten muss verhindert werden. Um die Daten zu schützen, ist die Erstellung eines tragfähigen Sicherheitskonzeptes notwendig. Dies ist mit Aufwand verbunden und bietet dem Datensammler wenige Vorteile. Bisherige Strategien zur Vorratsdatenspeicherung lassen einen umfassenden Schutz der Daten vermissen.

In dieser Arbeit werden zum Schutz von persönlichen Daten Konzepte und Verfahren entwickelt, die die Interessen an der Vorratsdatenspeicherung ausgleichen. Der Missbrauch von Vorratsdaten wird verhindert, der Zugriff eindeutig geregelt und die Daten anonym gehalten. Zunächst werden allgemeine Sicherheitsanforderungen an eine Vorratsdatenspeicherung ermittelt. Die Anforderungen werden für eine Ausführung der Vorratsdatenspeicherung weiterentwickelt, bei der der Zugriff auf die Daten nur dann erlaubt wird, wenn die Anzahl von verdächtigen Aktionen eines Benutzers innerhalb eines festgelegten Vorhaltezeitraums einen Schwellwert überschreitet. Es wird ein Schema für diese Art der Vorratsdatenspeicherung entwickelt, das beweisbar die Sicherheit der Vorratsdaten und die Anonymität der entsprechenden Benutzer gewährleistet. Im Falle eines Zugriffs können nur die Daten geöffnet werden, die innerhalb des Vorhaltezeitraums liegen. Zudem können die Vorratsdaten getrennt vom Aufzeichnungsort gespeichert werden, zum Beispiel direkt bei einer Sicherheitsbehörde.

Eine wichtige Eigenschaft eines Schemas für eine Vorratsdatenspeicherung ist die Begrenzung des Vorhaltezeitraums. Im Idealfall würden alte Daten gelöscht und die Kenntnis der Nutzungshistorie eines Benutzers wäre auf diesen Zeitraum begrenzt. Die Löschung von Daten ist jedoch schwer zu kontrollieren, insbesondere, wenn die Vorratsdaten bei einer nicht vertrauenswürdigen Partei gespeichert werden. Damit das Wissen über die Historie begrenzt bleibt, werden Protokolle zum Schutz der Historie entwickelt. Bei diesen Ansätzen beruht die Sicherheit auf graphentheoretischen Eigenschaften und probabilistischen Prozessen. Ein Ansatz ist, die Historien der Benutzer miteinander zu verflechten. Ein weiterer Ansatz basiert auf der zufälligen Auswahl von Knoten eines Graphen durch ein sicheres Protokoll. Der Zerfall des entstehenden Teilgraphen in mehrere Komponenten korrespondiert mit der Entkopplung der Historie.

Danksagung

An erster Stelle möchte ich mich bei Andreas Jakoby für die ausgezeichnete Betreuung in den letzten Jahren bedanken. Jederzeit war es möglich, seinen Rat einzuholen und über Probleme zu diskutieren.

Zudem möchte ich den Dank an mehrere Menschen richten, die mich während der Anfertigung dieser Arbeit unterstützt haben:

An Gisa Drebes und Lisanne Jakoby für die vielen Sonntagnachmittage, an denen sie Andreas und den Kuchen mit mir geteilt haben.

An Rüdiger Reischuk für die Möglichkeit, am Institut für Theoretische Informatik an meiner Dissertation arbeiten zu dürfen.

An alle Kollegen und Studenten des Instituts für Theoretische Informatik für die angenehme Atmosphäre. Insbesondere gilt mein Dank Johannes Textor und Till Tantau für die interessanten Diskussionen und hilfreichen Anregungen. Zudem bedanke ich mich bei Christoph Stockhusen für die praktische Umsetzung des Schemas zur Vorratsdatenspeicherung in seiner Studienarbeit.

An Ines Hinkelmann, Mark Busse und Nils Papenberg für das Korrekturlesen der Arbeit sowie an Juliane Hardt und Kai Trojahner für die gegenseitige Motivation und die vielen guten Gespräche.

An meine Eltern, die mich zu allen Zeiten unterstützt haben.

An meine liebe Freundin Anja Stackebrandt für die kritischen Korrekturen und Anmerkungen sowie ihr Verständnis und ihren Rückhalt.

Inhaltsverzeichnis

1 Einleitung

Aus der EU-Richtlinie 2006/24/EG:

> „Mit dieser Richtlinie sollen die Vorschriften der Mitgliedstaaten über die Pflichten von Anbietern öffentlich zugänglicher elektronischer Kommunikationsdienste oder Betreibern eines öffentlichen Kommunikationsnetzes im Zusammenhang mit der Vorratsspeicherung bestimmter Daten[1], die von ihnen erzeugt oder verarbeitet werden, harmonisiert werden, um sicherzustellen, dass die Daten zum Zwecke der Ermittlung, Feststellung und Verfolgung von schweren Straftaten, wie sie von jedem Mitgliedstaat in seinem nationalen Recht bestimmt werden, zur Verfügung stehen.[2] [...]
>
> Die auf Vorrat gespeicherten Daten sind von der gleichen Qualität und unterliegen der gleichen Sicherheit und dem gleichen Schutz wie die im Netz vorhandenen Daten, in Bezug auf die Daten werden geeignete technische und organisatorische Maßnahmen getroffen, um die Daten gegen zufällige oder unrechtmäßige Zerstörung, zufälligen Verlust oder zufällige Änderung, unberechtigte oder unrechtmäßige Speicherung, Verarbeitung, Zugänglichmachung oder Verbreitung zu schützen, [... und] um sicherzustellen, dass der Zugang zu den Daten ausschließlich besonders ermächtigten Personen vorbehalten ist,[3] [...].“

Die vorliegende Arbeit beschäftigt sich mit Verfahren zur sicheren und anonymen Speicherung von Vorratsdaten und mit der Frage, welche Informationen aus solchen gespeicherten Daten gewonnen werden können. Bei den meisten Anbietern elektronischer Dienste fallen Benutzerdaten an, die zur weiteren Verarbeitung gespeichert werden. Hierbei steht häufig der aus den gespeicherten Daten zu erwartende Wissensgewinn im Vordergrund, während der Schutz dieser Daten vernachlässigt wird. Aktuell ist bei den Datenschützern sogar ein

[1] Benutzerkennungen, Rufnummern, Name und Anschrift der Teilnehmer, Standortkennungen, siehe [33] Artikel 5. Fußnoten nicht im Original.

[2] siehe [33] Artikel 1 Absatz 1

[3] siehe [33] Artikel 7 Buchstaben a und b

Zuwachs der Anzahl an Beschwerden für den nicht öffentlichen Bereich zu beobachten [107].

Ein prominentes Beispiel für den Mangel des Datenschutzes in der Vorratsdatenspeicherung ist die Umsetzung der zitierten „Richtlinie 2006/24/EG des Europäischen Parlaments und des Rates vom 15. März 2006 über die Vorratsspeicherung von Daten, die bei der Bereitstellung öffentlich zugänglicher elektronischer Kommunikationsdienste oder öffentlicher Kommunikationsnetze erzeugt oder verarbeitet werden, und zur Änderung der Richtlinie 2002/58/EG“ [33] in deutsches Recht. Sie wurde zur Verbesserung der Abwehr und der Aufklärung von schweren Straftaten erlassen. Die Provider von Telekommunikationsdiensten müssen demnach Kommunikationsdaten ihrer Benutzer beziehungsweise Kunden speichern. Die Speicherung beinhaltet die Quelle und das Ziel der Kommunikation sowie ihren Typ, ihren Zeitpunkt und ihre Dauer. Diese Daten müssen für mindestens ein halbes Jahr vorgehalten werden.

Diese Richtlinie wurde im Frühjahr 2009 durch das „Gesetz zur Neuregelung der Telekommunikationsüberwachung und anderer verdeckter Ermittlungsmaßnahmen sowie zur Umsetzung der Richtlinie 2006/24/EG“ [1] in nationales Recht umgesetzt. Aufgrund schwerwiegender Verstöße gegen die grundgesetzlich geschützte Privatsphäre hat das Bundesverfassungsgericht in seiner Entscheidung [16] vom Frühjahr 2010 dieses Gesetz für nichtig erklärt. In dem Urteil heißt es:

> „Die Ausgestaltung einer vorsorglichen Telekommunikationsverkehrsdatenspeicherung, wie sie in § 113a TKG [Telekommunikationsgesetz] vorgesehen ist, unterliegt besonderen verfassungsrechtlichen Anforderungen insbesondere hinsichtlich der Datensicherheit, des Umfangs der Datenverwendung, der Transparenz und des Rechtsschutzes. Nur wenn diesbezüglich hinreichend anspruchsvolle und normenklare Regelungen getroffen sind, ist der in einer solchen Speicherung liegende Eingriff verhältnismäßig im engeren Sinne.[4] [...]
>
> Angesichts des Umfangs und der potentiellen Aussagekraft der mit einer solchen Speicherung geschaffenen Datenbestände ist die Datensicherheit für die Verhältnismäßigkeit der angegriffenen Vorschriften von großer Bedeutung. Dieses gilt besonders, weil die Daten bei privaten Diensteanbietern gespeichert werden, die unter den Bedingungen von Wirtschaftlichkeit und Kostendruck handeln und dabei nur begrenzte Anreize zur Gewährleistung von Datensicherheit haben. Sie handeln grundsätzlich privatnützig und sind nicht durch spezifische Amtspflichten gebunden. Zugleich ist die Gefahr eines illegalen Zugriffs auf die Daten groß, denn angesichts ihrer vielseitigen Aussagekraft können diese für verschiedenste Akteure attraktiv sein. Geboten ist daher ein besonders hoher Sicherheitsstandard, der über das allgemein verfassungsrechtlich gebotene Maß für die Aufbewahrung von Daten der Telekommunikation hinausgeht. Solche Anforderungen der Datensicherheit gelten dabei sowohl für die Aufbewahrung der Daten als auch für deren Übermittlung; ebenso bedarf es effektiver Sicherungen zur Gewährleistung der Löschung der Daten.[5] [...]
>
> Die angegriffenen Vorschriften genügen diesen Anforderungen nicht.[6]“

[4] siehe [16] Absatz 220
[5] siehe [16] Absatz 222
[6] siehe [16] Absatz 269

Der Schutz der Daten wird bei bisherigen Verfahren vernachlässigt. In dieser Arbeit stehen der Datenschutz und die Anonymität der teilnehmenden Personen im Zentrum des Entwurfs von Verfahren zur Vorratsdatenspeicherung, die trotzdem den Zugriff auf die vollständigen Nutzungsdaten in berechtigten, klar geregelten Fällen zulassen.

1.1 Vorratsdatenspeicherung in der Literatur

Die Einführung der Datenspeicherung und Überwachung im Internet hat einige Aufmerksamkeit in der Informatik, in der Sozialwissenschaft und in der Rechtswissenschaft auf sich gezogen. Marx [71] identifiziert vier Bedingungen, unter denen sich bei der Speicherung von Daten ethische Bedenken ergeben:

1. Beim Sammeln von Daten wird physisches oder psychisches Leid zugefügt.
2. Aus der Sammlung von Daten werden ungültige Ergebnisse oder Schlüsse gezogen.
3. Die Datensammlung überschreitet persönliche Grenzen ohne den Verletzten davon in Kenntnis zu setzen.
4. Das Sammeln von Daten verletzt Vertrauen.

Marx legt zudem dar, dass die Versuchung des Missbrauchs von gesammelten Daten groß sei, wenn man Zugriff darauf habe. Daher sollte bei der Datenspeicherung generell vorsichtig vorgegangen und die Risiken des Missbrauchs minimiert werden.

Blanchette und Johnson [8, 9] argumentieren, dass mit der digitalen Speicherung und Analyse von Daten die soziale Vergesslichkeit verloren geht. Auch wenn später die gespeicherten Daten gelöscht werden, so ist es nahezu unmöglich, die analytisch gewonnenen und verknüpften Daten komplett aus dem Internet zu tilgen. Als Konsequenz lassen sich persönliche Daten zurückverfolgen, aber nur schwer aus dem Netz entfernen. Der Einsatz von kryptographischen Primitiven wie Pseudonyme [19, 21] versprechen laut Blanchette und Johnson einige Hoffnung für die Sicherheit bei einer möglichen technischen Umsetzung.

Landau [69] spricht bei der Überwachung und Datenspeicherung im Internet von einem konstruierten Bruch der Sicherheit. Zur Umsetzung müssen Sicherheits- und Kommunikationsprotokolle für die Überwachung geöffnet bleiben. Es bestehe im Gegensatz zur analogen Überwachung die Gefahr, dass Dritte über diese Öffnung das System und die Daten angreifen. Daher sei es nötig, den Einsatz solcher Techniken nur in entsprechend geeigneten Szenarien zuzulassen.

Pfitzmann und Köpsell [87] sehen die Bruchstelle der Sicherheit bei bisherigen Verfahren zur Vorratsdatenspeicherung in der Komplexität des Systems zur Sicherung, Speicherung und zum Abruf der Daten sowie in der damit verbundenen Fehleranfälligkeit. Aufgezeichnete Daten sind noch vor der Speicherung zu verschlüsseln. Dabei sei nach heutigem Kenntnisstand asymmetrische Kryptographie einzusetzen. Es sei wünschenswert, dass die Freigabe der Schlüssel nur durch mehrere Personen gemeinsam erfolgen kann. Nach Ansicht von Pfitzmann und Köpsell besteht bei der Vorratsspeicherung der kritische Punkt darin, wie der Zugriff auf die Entschlüsselungsschlüssel und deren Verwendung beschränkt wird. Der Gesetzgeber hat in jedem Fall Sicherheitslücken entgegenzuwirken, die durch die Vorratsdatenspeicherung entstehen.

Roßnagel et al. [98] kritisieren die völlig unzureichende Absicherung der Vorratsdaten unter Anwendung der gesetzlichen Vorratsdatenspeicherung nach [1]. Insbesondere mahnen

sie die fehlende Trennung von Datensammlung, Datenspeicherung und Zugriffsmöglichkeit auf die Daten an. Dadurch ist der bei einer Kompromittierung entstandene Schaden nicht ausreichend begrenzt. Die physikalische Trennung von Speicherung und Sammlung ermöglicht zudem unterschiedliche Löschfristen der Vorratsdaten am Sammlungs- und am Speicherort. Darüber hinaus gehen Roßnagel et al. weiter auf die Problematik der Löschung der Vorratsdaten ein. Es müsse garantiert sein, dass alle Informationen aus der Datensammlung gelöscht und vor Wiederverwendbarkeit geschützt werden. Der Datenabruf müsse unbedingt geregelt und nachvollziehbar sein. Das Bundesverfassungsgericht teilt diese Bedenken hinsichtlich mangelnder Verschlüsselung, nicht getrennter Speicherung und problematischer Zugriffsregelung [16]. Darüber hinaus hält das Gericht die Speicherung bei privaten Unternehmern für unsicher und rügt die mangelnde Transparenz und Informationspflicht gegenüber den Bürgern.

Einige Veröffentlichungen beschäftigen sich in letzter Zeit mit der technischen Umsetzung der Vorratsdatenspeicherung [65, 108, 112] oder der effizienten Analyse der Daten [85]. Allerdings lassen nach meinem Wissen bisherige Arbeiten zur technischen Umsetzung eine ausreichende Behandlung des Datenschutzes vermissen.

1.2 Einsatzbereiche der Vorratsdatenspeicherung

Wir betrachten nun einige Beispiele der Vorratsdatenspeicherung. Dabei zielt die Speicherung in allen Fällen auf den Gewinn von bestimmtem Wissen ab.

1.2.1 Kriminalitätsbekämpfung

Seit einiger Zeit sind viele Regierungen bestrebt, ein höheres Maß an Überwachung durchzusetzen. Dazu gehören die Rasterfahndung, die automatische Übermittlung von Flugpassagier- und Banktransferdaten, die Videoüberwachung von öffentlichen Plätzen oder die Speicherung von Telekommunikationsverbindungsdaten auf Vorrat. Begründet werden diese Eingriffe in die Privatsphäre durch die wachsende Terrorgefahr und die zunehmende technologische Aufrüstung von Kriminellen. Ziel ist es, sowohl die Kommunikationsverbindungen von erkannten Tätern zu untersuchen als auch aus dem Datenbestand Individuen zu identifizieren, die viele verdächtige Aktionen ausführen. Zudem ist die Aufzeichnung und die Analyse von Vorratsdaten mittlerweile technologisch durchführbar. Nicht zuletzt wegen der Anschläge in London im Juli 2005 hat die Europäische Union die oben bereits diskutierte Richtlinie 2006/24/EG zur Vorratsdatenspeicherung von Kommunikationsdaten erlassen. Durch den Zugriff auf diese Daten erhoffen sich die Strafverfolgungsbehörden eine bessere Beweissicherung und die Identifizierung von verdächtigen Personen.

1.2.2 Verhinderung von illegalen Finanztransaktionen

Bei Finanztransaktionen gilt es, die Einführung von Geld aus illegalen Quellen in den regulären Finanzkreislauf, die sogenannte Geldwäsche, zu stoppen. Schon seit den siebziger Jahren werden in den Vereinigten Staaten Währungstransaktionen in der Summe von mehr als 10.000$ je Tag der Finanzaufsicht zur Speicherung auf Vorrat gemeldet. In Deutschland ist bei solchen Bewegungen ab 15.000€ die Identität des Einführers festzustellen und zu speichern. Bei Verdachtsmomenten, zum Beispiel wenn von einer Person eine bestimmte

Schwelle von Transaktionen erreicht wird, ist die Finanzaufsicht darüber zu informieren. Im Allgemeinen werden die Daten für fünf Jahre aufbewahrt.

1.2.3 Verwaltung des Verkehrszentralregisters

In Deutschland werden die Verfehlungen eines Verkehrsteilnehmers beim Kraftfahrt-Bundesamt in Flensburg gespeichert. Die Verfehlungen werden dort in Form von Punkten ausgedrückt. Für Ordnungswidrigkeiten gibt es einen bis vier Punkte, Straftaten werden mit mehr Punkten belegt. Die Punkte verfallen nach einer bestimmten Zeit. Beispielsweise werden Ordnungswidrigkeiten spätestens nach fünf Jahren aus dem Zentralregister getilgt. Erreicht ein Verkehrsteilnehmer 18 Punkte, wird ihm seine Fahrerlaubnis entzogen. Er kann diese erst nach einer medizinisch-psychologischen Untersuchung zurückerhalten.

1.2.4 Intrusion-Detection

Bei der Erkennung von Einbruchsversuchen in ein Computersystem werden in den meisten Fällen Profile von Kommunikationspartnern anhand von IP-Adressen erstellt und gespeichert. Anhand der Profile kann ein Intrusion-Detection-System Aktionen der Kommunikationspartner klassifizieren, gegebenenfalls einen Einbruch erkennen und den Angriff abwehren, zum Beispiel durch Blockieren und Ausschließen der Kommunikation. Dabei werden die Profile häufig statistisch oder durch Techniken des Data-Mining analysiert oder mit Expertenwissen verknüpft. Ein einfaches und oft angewandtes Mittel zur Erkennung eines Angriffs ist die Bestimmung von bösartigem Verhalten mittels eines Schwellwertes. Überschreitet ein Kommunikationspartner mit seiner Aktion einen Schwellwert, so wird dieser als auffällig oder verdächtig eingestuft. Die Intrusion-Detection ähnelt vom Prinzip her der Verwaltung des Verkehrszentralregisters. Doch während sich das Verkehrszentralregister noch per Hand verwalten lässt, ist dies bei der Intrusion-Detection aufgrund der hohen Anzahl von Transaktionen pro Zeit unmöglich. Daher muss die Verwaltung von einem elektronischen System übernommen werden.

1.2.5 Erstellung von Kundenprofilen im Marketing

Unternehmen mit vielen Kunden sind daran interessiert, das Verhalten ihrer Kunden kennenzulernen. Insbesondere beim Handel mit Konsumgütern helfen Kundenprofile, Werbung gezielt auf die Kunden abzustimmen und damit den Umsatz zu steigern. Das Marketing kennt zur Erstellung von Kundenprofilen viele Techniken. Personalisierte Kunden- und Bonuskarten geben Aufschluss über jeden getätigten Kauf. Durch das Verwenden von Konten in Online-Stores kann zusätzlich jede Bewegung vor einem Kauf aufgezeichnet werden. Die Verknüpfung dieser Daten mit den Kundenprofilen anderer Unternehmen steigert darüber hinaus den möglichen Erkenntnisgewinn über die Kunden. Die einmal gewonnenen und mit einer Person verknüpften Daten stehen dauerhaft zur Verfügung. Die Kontrolle des Zugriffs, die Begrenzung der Verwendung oder die Löschung dieser Daten sind derzeit kaum oder gar nicht geregelt.

1.3 Ziele der Arbeit

Die vorliegende Arbeit widmet sich der Entwicklung eines Schemas zu einer Variante der Vorratsdatenspeicherung, bei der der Zugriff auf die Vorratsdaten von einem Schwellwert

abhängt. Nur wenn ein Benutzer innerhalb eines bestimmten Intervalls (der *Vorhaltezeit*) einen *Schwellwert* an sogenannten *kritischen Aktionen* überschreitet, sollen die folgende Punkte zutreffen:

Identifizierung:
Der Benutzer wird identifiziert.

Öffnung:
Alle Daten des Benutzers, die innerhalb der Vorhaltezeit liegen, können geöffnet werden.

Für einen Benutzer, der innerhalb der Vorhaltezeit den Schwellwert an kritischen Aktionen nicht überschritten hat, soll gelten:

Schutz:
Seine Daten sind vor unbefugtem Zugriff geschützt.

Anonymität:
Dem Benutzer können seine Daten nicht zugeordnet werden. Sie sind anonym.

Als weitere Ziele sind zu nennen:

Trennung:
Die Daten werden getrennt von ihrem Aufzeichnungsort gespeichert.

Löschung:
Daten außerhalb der Vorhaltezeit werden gelöscht oder sind unbrauchbar.

Einige der zuvor genannten Einsatzgebiete können auch als Beispiel für eine Vorratsdatenspeicherung mit Schwellwertbedingung angesehen werden. Dabei entsprechen zum Beispiel Verkehrsdelikte, Finanztransaktionen und verdächtige Kommunikation den kritischen Aktionen.

Die Sicherheit des Schemas soll hohen Sicherheitsstandards genügen, das heißt, es soll kryptographisch sicher und beweisbar anonym sein. Weiterhin soll es sich mit vertretbarem Aufwand auf gängigen Systemen implementieren lassen.

1.4 Aufbau und Ergebnisse der Arbeit

Diese Arbeit beschäftigt sich nach meinem Wissen als erste mit der Umsetzung der Vorratsdatenspeicherung mit kryptographisch beweisbaren Eigenschaften bezüglich der Sicherheit der gespeicherten Daten und der Anonymität der Benutzer.

Die folgenden zwei Kapitel geben einen allgemeinen Überblick über die Grundlagen der Kryptographie. In Kapitel 2 werden die benötigten Begriffe aus der Wahrscheinlichkeitstheorie sowie die kryptographische Sicherheit von Protokollen beschrieben. Das zweite Grundlagenkapitel, Kapitel 3, geht auf die für diese Arbeit wichtigen kryptographischen Primitive und Protokolle ein.

Den Einstieg in die private Vorratsdatenspeicherung bildet Kapitel 4. Zunächst befassen wir uns mit den allgemeinen Sicherheitsanforderungen an die Vorratsdatenspeicherung. Anschließend wird ein effizientes Schema zur sicheren Vorratsdatenspeicherung mit Schwellwertbedingung vorgestellt. Dieses erfüllt die oben genannten Ziele in den Punkten Identifizierung, Öffnung, Schutz, Anonymität und Trennung. An dem beschriebenen Verfahren nehmen

drei Arten von Parteien teil: Benutzer führen Aktionen aus, die von Providern aufgezeichnet und verschlüsselt werden. Die verschlüsselten Daten werden an die sogenannte Sammelstelle zur zentralen Speicherung übertragen. Nur wenn eine zuvor definierte Bedingung für die Identifizierung und Öffnung erfüllt ist, kann die Sammelstelle die entsprechenden Daten entschlüsseln. Dies geschieht ohne weitere Kommunikation mit einer anderen Partei. Durch die Speicherung der Daten bei der Sammelstelle und nicht bei den Providern ist das Prinzip der Trennung erfüllt. Gleichzeitig erfüllt das Schema die Eigenschaften Schutz und Anonymität. Es wird gezeigt, dass ein Angreifer mit nicht mehr als vernachlässigbarer Wahrscheinlichkeit unbefugt Daten entschlüsseln oder einem Benutzer zuordnen kann. Bei der Analyse der Sicherheit wird davon ausgegangen, dass die Provider vertrauenswürdig sind, da nur die Sammelstelle Interesse an den Vorratsdaten hat. Die Sammelstelle und die Benutzer werden dagegen als potentielle Angreifer betrachtet.

Die Behandlung der Sammelstelle als Angreifer stellt bei der Gewährleistung der Löschung ein Problem dar. Sie kann nicht dazu gezwungen werden, Daten außerhalb der Vorhaltezeit zu löschen. Allerdings können Daten quasi als gelöscht betrachtet werden, wenn die Sammelstelle aus ihnen kein weiteres Wissen gewinnen kann, diese Daten also unbrauchbar geworden sind. Durch die Erfüllung der Eigenschaft Schutz könnte man meinen, dass die Sammelstelle kein Wissen aus Daten gewinnen kann, die außerhalb der Vorhaltezeit liegen. In Kapitel 5 untersuchen wir daher, welches Wissen aus nicht gelöschten Daten gewonnen werden kann. Es zeigt sich, dass die Kenntnis über die Zeitpunkte des Auftretens von Vorratsdaten ausreicht, um Wissen über die *Historie* eines Benutzers zu sammeln. Die Historie spiegelt das Verhalten im zeitlichen Verlauf wider. Würde die Sammelstelle Daten löschen, könnte sie nur begrenzte Information über die Historien der Benutzer sammeln. Um die Begrenzung des Wissens über die Historie eines Benutzers zu erreichen, werden in den Kapiteln 5, 6 und 7 Protokolle vorgestellt, die das Wissen einer bösartigen Sammelstelle über die Historien begrenzen. In dieser Arbeit wird in diesem Zusammenhang auch von der *Entkopplung der Historie* gesprochen.

Im ersten vorgestellten Verfahren in Kapitel 5 wird die Entkopplung der Historie eines Benutzers erreicht, wenn der Benutzer für eine bestimmte Zeit keine kritischen Aktionen durchführt. Dieses Verfahren ist jedoch angreifbar, wenn wenige Benutzer zu derselben Zeit kritische Aktionen ausführen. Bei den weiteren Verfahren dieses Kapitels zeigt sich, dass die Entkopplung durch Trade-Offs erlangt werden kann. Durch den Einsatz des Verfahrens aus Abschnitt 5.4 können wir eine Entkopplung der Historie durch die Lockerung der Schwellwertbedingung erreichen. Vermischen wir die Historien von verschiedenen Benutzern untereinander, so zeigt die Analyse in Abschnitt 5.5, dass eine verbesserte Entkopplung der Historie mit einer höheren Laufzeit zur Erkennung des Schwellwertübertritts verbunden ist.

In Kapitel 6 untersuchen wir, wie Vorratsdaten eines Benutzers über eine Graphenstruktur verflochten werden können und unter Verwendung eines randomisierten Protokolls die Entkopplung der Historie erreicht werden kann. Die Entkopplung korrespondiert hier mit einem Zerfall der Graphenstruktur in mehrere Zusammenhangskomponenten. Wir können garantieren, dass der Schwellwert d mit Sicherheit $1-\varepsilon$ eingehalten werden kann. Gleichzeitig beträgt die Wahrscheinlichkeit höchstens e^{-c}, dass eine Entkopplung erst nach $\mathcal{O}(c \cdot \frac{d^2}{\varepsilon^2} \cdot \log(\frac{d}{\varepsilon}))$ Nachrichten auftritt. In den Abschnitten 6.7 bis 6.9 untersuchen wir die Sicherheit dieses Verfahrens bezüglich verschiedener Angreifertypen und erweitern das Verfahren, so dass es auch gegenüber aktiven Angreifern robust ist. Die Nutzung dieser Graphenstrukturen eignet sich nicht nur zum Schutz der Historie in der klassischen Vorratsdatenspeicherung, sondern auch speziell bei der privaten Erstellung von Kundenprofilen

im Marketing. Die Kunden fordern bei der Datenerhebung ein hohes Maß an Datenschutz. Insbesondere sollen die Profile anonym sein, also nicht direkt mit der Identität des Kunden verknüpft werden können. Zudem soll erreicht werden, dass die Profile in ihrer Größe beschränkt bleiben. Damit wird der potentielle Wissensgewinn aus der Kombination der Profile mit anderen Datenquellen beschränkt und die Identität des Kunden geschützt. Auf der anderen Seite interessieren sich die Unternehmen als Profilersteller für das typische Kundenverhalten und benötigen eine bestimmte Mindestgröße der Profile. Über entsprechende Protokoll- und Sicherheitsparameter können die Anonymität der Kunden und die Größe der Kundenprofile angepasst werden. In Kapitel 7 betrachten wir eine erweiterte Graphenstruktur mit ähnlichen Sicherheitseigenschaften. Durch diese Struktur kann gegenüber dem Verfahren aus Kapitel 6 die Anzahl der benötigten Zufallsbits etwa auf den $\ln(d)$-ten Bruchteil gesenkt und die Kommunikationskomplexität um die Hälfte reduziert werden. Dafür kann eine Entkopplung mit großer Sicherheit erst nach mindestens $\Omega(d^3)$ Nachrichten gewährleistet werden.

Die Arbeit endet mit einem tabellarischen Vergleich der verschiedenen Protokolle zur Entkopplung der Historie und einem Ausblick auf weiterführende Forschungsarbeiten.

2

Grundlagen: Wahrscheinlichkeitstheorie, Berechnungsmodelle in der Kryptographie, Sicherheit von Protokollen

Im Verlauf dieses Kapitels werden wir auf wichtige Begriffe und Grundlagen dieser Arbeit eingehen. Die Basis der meisten der folgenden Definitionen bildet die Wahrscheinlichkeitstheorie. Daher werden wir einige grundlegende Definitionen und Ergebnisse aus diesem Bereich zuerst vorstellen. Wir orientieren uns dabei an [32, 41, 97, 80, 51]. Bei der anschließenden Einführung in die Bereiche der Sicherheit und der kryptographischen Protokolle beziehen wir uns insbesondere auf die Bücher von Oded Goldreich [41, 42, 39]. Diese und weitere seiner Arbeiten bilden einen guten Einstieg in die Theorie der Kryptologie.

2.1 Notationen

In dieser Arbeit ist ein *String* eine endlich lange Zeichenkette über dem Alphabet $\{0,1\}$ (*Binärstring*). Für einen String w gibt $|w|$ seine Länge an. Für einen String $w = w_1 \dots w_a$ und eine Menge $B = \{b_1, \dots, b_k\} \subseteq \{1, \dots, a\}$ mit $b_1 < b_2 < \dots < b_k$ sei $w_B = w_{b_1} \dots w_{b_k}$ der *Substring* von w bezüglich B. Der Operator $\circ$ bezeichnet die *Konkatenation* oder Verkettung von Strings, das heißt, für Strings $v = v_1 \dots v_a$ und $w = w_1 \dots w_b$ gilt $v \circ w = v_1 \dots v_a w_1 \dots w_b$. Wir verwenden das Symbol $\circ$ auch für die *Verkettung (Hintereinanderausführung) von Funktionen*. Für zwei Funktionen $f\colon A \to B$ und $g\colon B \to C$ und $x \in A$ sei $(f \circ g)(x) = g(f(x))$. Aus dem Zusammenhang wird immer deutlich sein, welche Verkettungsoperation gemeint ist.

Da wir binäre Strings verwenden, interpretieren wir diese manchmal auch als Binärzahlen sowie Binärstrings der Länge 1 als logische Variablen. Der Operator $\oplus$ stellt das bitweise XOR auf Binärstrings dar.

Wir sagen ein Wert b ist *polynomiell* in einem Wert a, falls es ein Polynom p gibt, so dass $b < p(a)$. Sind a und b Funktionen, dann ist a polynomiell in b, wenn für ein Polynom p und alle Eingaben x gilt, dass $b(x) \leq p(a(x))$ ist. Ein String v heißt polynomiell lang in

einem String w, falls es ein Polynom p gibt, so dass $|v| < p(|w|)$.

In dieser Arbeit bezeichnen wir die Logarithmen zur Basis 2 und e mit log und ln.

Zur Vollständigkeit sei hier noch auf zwei Abschätzungen hingewiesen, die sich aus der bernoullischen Ungleichung herleiten lassen:

$$0 \leq \left(1-\frac{1}{n}\right)^n \leq e^{-1} \text{ für } n \geq 1 \text{ und } 1 \leq \left(1+\frac{1}{n}\right)^n \leq e \text{ für } n > 0.$$

2.2 Wahrscheinlichkeiten und Verteilungen

In der Kryptologie sind wir daran interessiert, wie viel Wissen wir aus einem verschlüsselten Text oder aus der Kommunikation zwischen zwei Parteien gewinnen können. Die Wahrscheinlichkeitstheorie bildet eine Grundlage, um über Wissen oder gewonnene Information zu diskutieren.

Wir betrachten eine beliebige nichtleere *Grundmenge* Ω. Diese Menge nennen wir auch *Ereignisraum*. Als ein *Ereignis* bezeichnen wir eine beliebige Teilmenge $A \subseteq \Omega$ des Ereignisraums. Ein Element $\omega \in \Omega$ heißt *Elementarereignis*.

Mit $\Pr[A] \in [0,1]$ bezeichnen wir die *Wahrscheinlichkeit* für das Ereignis A. Dabei muss gelten, dass $\Pr[\Omega] = 1$. Die Funktion Pr bestimmt die *Verteilung* der Ereignisse. Wir betrachten nur diskrete Wahrscheinlichkeitsverteilungen, das heißt, Ω ist endlich oder abzählbar. Die wichtigsten Verteilungen, die wir verwenden sind:

1. *uniforme* oder *Gleichverteilung*: Für alle $\omega \in \Omega$ gilt:
$$\Pr[\omega] = \frac{1}{|\Omega|}.$$
2. *Binomialverteilung* mit Parametern $n > 0$ und $p \in [0,1]$, das heißt, $\Omega = \{0,\ldots,n\}$ und
$$\Pr[\omega] = \binom{n}{\omega} \cdot p^{\omega} \cdot (1-p)^{n-\omega}.$$
3. *geometrische Verteilung* mit Parameter $p \in [0,1]$, das heißt, $\Omega = \mathbb{N} \setminus \{0\}$ und
$$\Pr[\omega] = (1-p)^{\omega-1} \cdot p.$$
4. *negative Binomialverteilung* mit Parametern $r > 0$ und $p \in [0,1]$, das heißt, $\Omega = \mathbb{N} \setminus \{0,\ldots,r-1\}$ und
$$\Pr[\omega] = \binom{\omega-1}{r-1} \cdot p^r \cdot (1-p)^{\omega-r}.$$

Für Ereignisse A und B ist die *bedingte Wahrscheinlichkeit von A gegeben B* definiert als

$$\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]},$$

wobei $\Pr[B] > 0$. Die Ereignisse $A_1,\ldots,A_n$ nennen wir *(stochastisch) unabhängig*, falls

$$\Pr\left[\bigcap_j A_j\right] = \prod_j \Pr[A_j].$$

In dieser Arbeit besteht der Ereignisraum Ω in den meisten Fällen aus der Menge der binären Zeichenketten (Strings) einer bestimmten Länge, zum Beispiel ℓ. Häufig wird aus dieser Menge ein Element mit uniformer Wahrscheinlichkeit gezogen, das heißt, im Falle von ℓ-Bit-Strings mit Wahrscheinlichkeit $\frac{1}{2^\ell}$.

2.3 Zufallsvariablen und Momente

Unter einer *Zufallsvariablen* X mit Werten in $\mathcal{X}$ verstehen wir eine Abbildung vom Ereignisraum Ω nach $\mathcal{X}$. Dabei sei $\Pr[X = x]$ die Wahrscheinlichkeit, dass X den Wert $x \in \mathcal{X}$ annimmt, also

$$\Pr[X = x] = \Pr[\{\omega \in \Omega \mid X(\omega) = x\}].$$

Die Wahrscheinlichkeit $\Pr[X = x]$ bestimmt die Verteilung von X. Entsprechend der vorherigen Definitionen erweitern wir den Begriff Wahrscheinlichkeit $\Pr[X \in B]$ auf Teilmengen $B \subseteq \mathcal{X}$. Wir nennen die Zufallsvariablen $X_1, \ldots, X_n$ *(stochastisch) unabhängig*, falls

$$\Pr[X_i \in B_i \text{ für } i \in \{1, \ldots, n\}] = \prod_{i=1}^{n} \Pr[X_i \in B_i],$$

wobei $X_i \colon \Omega \to \mathcal{X}_i$ und $B_i \subseteq \mathcal{X}_i$.

Wir werden in dieser Arbeit nur *diskrete Zufallsvariablen* verwenden, das heißt, eine Zufallsvariable nimmt nur diskrete Werte an. Kommt in einem Ausdruck ein Symbol für eine Zufallsvariable mehrfach vor, so beziehen wir uns stets auf dieselbe Zufallsvariable. Wenn beispielsweise $R(\cdot,\cdot)$ ein zweistelliges Prädikat ist, dann gibt $\Pr[R(X,X)]$ die Wahrscheinlichkeit an, dass $R(x,x)$ gilt, wobei x mit der Wahrscheinlichkeit $\Pr[X = x]$ gezogen ist. Das bedeutet:

$$\Pr[R(X,X)] = \sum_x \Pr[X = x] \cdot \mathbb{I}(R(x,x)),$$

wobei $\mathbb{I}$ die Indikatorfunktion ist, das heißt, $\mathbb{I}(R(a,b)) = 1$, falls $R(a,b)$ gilt und gleich 0 sonst. Für Ausdrücke $R(X_1, \ldots, X_n)$ von n unabhängigen Zufallsvariablen definieren wir die Wahrscheinlichkeit $\Pr[R(X_1, \ldots, X_n)]$ durch

$$\Pr[R(X_1, \ldots, X_n)] = \sum_{x_1, \ldots, x_n} \left(\prod_i \Pr[X_i = x_i] \right) \cdot \mathbb{I}(R(x_1, \ldots, x_n)).$$

Für eine diskrete Zufallsvariable X ist $\mathbb{E}(X)$ ihr *Erwartungswert* mit

$$\mathbb{E}(X) = \sum_x x \cdot \Pr[X = x].$$

Wir nennen den Wert $\mathbb{E}(X^i)$ das *i-te Moment von X*. Der Erwartungswert ist demzufolge das erste Moment. Die *momentgenerierende Funktion* einer Zufallsvariable X ist für alle reellen Werte t definiert über

$$\mathbb{M}(t) = \mathbb{E}\left(e^{tX}\right) = \sum_x e^{tx} \cdot \Pr[X = x].$$

Wir können alle Momente von X durch sukzessives Ableiten der momentgenerierenden Funktion erhalten:

$$\mathbb{M}^{(i)}(t) = \mathbb{E}(X^i e^{tX}) \quad \text{und} \quad \mathbb{M}^{(i)}(0) = \mathbb{E}(X^i).$$

Die *Varianz* von X lässt sich aus dem ersten und zweiten Moment berechnen:

$$\mathrm{Var}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)^2.$$

Die Eigenschaften der in dieser Arbeit wichtigen Verteilungen sind in Tabelle 2.1 aufgeführt.

Verteilung von X	$\Pr[X=x]$	$\mathbb{M}(t)$	$\mathbb{E}(X)$	$\mathrm{Var}(X)$
gleichverteilt über (a,b)	$\begin{cases} \frac{1}{\lvert b-a\rvert} & a<x<b \\ 0 & \text{sonst} \end{cases}$	$\frac{e^{tb}-e^{ta}}{t(b-a)}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
binomialverteilt	$\binom{n}{x}p^x(1-p)^{n-x}$	$(pe^t+1-p)^n$	np	$np(1-p)$
geometrisch verteilt	$(1-p)^{x-1}p$	$\frac{pe^t}{1-(1-p)e^t}$	$\frac{1}{p}$	$\frac{1-p}{p^2}$
negativ binomialverteilt	$\binom{x-1}{r-1}p^r(1-p)^{x-r}$	$\left(\frac{pe^t}{1-(1-p)e^t}\right)^r$	$\frac{r}{p}$	$\frac{r(1-p)}{p^2}$

Tabelle 2.1: Eigenschaften einer diskret verteilten Zufallsvariable X bei Gleichverteilung, Binomialverteilung, geometrischer Verteilung und negativer Binomialverteilung

2.4 Beziehung der negativen Binomialverteilung zu anderen Verteilungen

Seien $X_1, X_2, \ldots$ unabhängige 0-1-wertige Zufallsvariablen. Die Zufallsvariable X_i beschreibt für $X_i = 1$ den Erfolg im i-ten Versuch eines Experiments. Ein Versuch sei mit Wahrscheinlichkeit $p \in [0,1]$ erfolgreich, das heißt, für $i \geq 1$ gilt, dass $Pr[X_i = 1] = p$.

Sei $N_{r,p}$ eine Zufallsvariable, die für $N_{r,p} = n$ das Ereignis beschreibt, dass sich im n-ten Versuch der r-te Erfolg einstellt. Das heißt, unter in den Versuchen $X_1, \ldots, X_{n-1}$ gibt es genau $r-1$ Erfolge. Dann ist $N_{r,p}$ eine negativ binomialverteilte Zufallsvariable mit den Parametern r und p. Für $r = 1$ erhalten wir die geometrische Verteilung. Sie ist somit ein Spezialfall der negativen Binomialverteilung.

Sei $B_{n,p}$ eine Zufallsvariable, die für $B_{n,p} = r$ angibt, dass in den Versuchen $X_1, \ldots, X_n$ genau r Erfolge stattfinden. Dann ist $B_{n,p}$ eine binomialverteilte Zufallsvariable mit den Parametern n und p.

Falls $N_{r,p} > n$, dann findet der r-te Erfolg erst nach dem n-ten Versuch statt. Somit gibt es unter den ersten n Versuchen weniger als r Erfolge. Das Ereignis $B_{n,p} < r$ entspricht genau diesem Fall. Folglich gilt:

$$\Pr[N_{r,p} > n] = \Pr[B_{n,p} < r].$$

2.5 Wichtige Ungleichungen

Die folgenden wahrscheinlichkeitstheoretischen Ungleichungen werden wir an mehreren Stellen dieser Arbeit einsetzen. Die Abschätzungen beziehen sich auf reellwertige Zufallsvariablen.

2.5.1 Erste Bonferroni-Ungleichung

Die erste Bonferroni-Ungleichung wird auch Boolesche Ungleichung oder *Union-Bound* genannt. Für beliebige Ereignisse $A_1, \ldots, A_n$ gilt, dass

$$\Pr\left[\bigcup_{i=1}^{n} A_i\right] \leq \sum_{i=1}^{n} \Pr[A_i].$$

2.5.2 Markov-Ungleichung

Sei X eine nicht-negative Zufallsvariable und $a \in \mathbb{R}$. Dann gilt:

$$\Pr[X \geq a] \leq \frac{\mathbb{E}(X)}{a}.$$

Die *Markov-Ungleichung* wenden wir typischerweise an, wenn wir sehr wenig über die Verteilung wissen. Wir müssen lediglich den Erwartungswert von X kennen. Sind uns weitere strukturelle Eigenschaften der Verteilung der Zufallsvariable bekannt, dann gelangen wir über die Markov-Ungleichung zu weiteren Abschätzungen.

2.5.3 Chernoff-Schranken

Chernoff stellte 1952 diese wichtigen Schranken zur Abschätzung der Wahrscheinlichkeiten vor [24]. Weitere Darstellungen und spezielle Abschätzungen finden sich in [70, 97, 52, 80, 79, 110].

Sei X eine Zufallsvariable und sei a eine positive reelle Zahl. Dann gilt:

$$\begin{aligned} \Pr[X \geq a] &\leq e^{-ta} \cdot \mathbb{M}(t) && \text{für alle } t > 0 \text{ und} \\ \Pr[X \leq a] &\leq e^{-ta} \cdot \mathbb{M}(t) && \text{für alle } t < 0. \end{aligned}$$

Für die Binomialverteilung betrachten wir diese Schranken noch etwas genauer. Seien $X_1, \ldots, X_n$ unabhängige 0-1-wertige Zufallsvariablen mit $\Pr[X_i = 1] = p$ für $i \in \{1, \ldots, n\}$. Dann ist $X = \sum_i X_i$ binomialverteilt mit $\mathbb{E}(X) = np$ und es gilt:

$$\begin{aligned} \Pr[X \geq (1+\delta) \cdot \mathbb{E}(X)] &\leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mathbb{E}(X)} && \text{für } \delta \geq 0 \text{ und} \\ \Pr[X \leq (1-\delta) \cdot \mathbb{E}(X)] &\leq \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mathbb{E}(X)} && \text{für } 1 > \delta \geq 0. \end{aligned}$$

Wir können diese Ungleichungen weiter vereinfachen. Für $\delta > 0$ erhalten wir

$$\Pr[X \geq (1+\delta) \cdot \mathbb{E}(X)] \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mathbb{E}(X)} \leq e^{-\min(\delta, \delta^2) \cdot \mathbb{E}(X)/3}$$

und für $1 > \delta \geq 0$ erhalten wir

$$\Pr[X \leq (1-\delta) \cdot \mathbb{E}(X)] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mathbb{E}(X)} \leq e^{-\delta^2 \cdot \mathbb{E}(X)/2} \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mathbb{E}(X)}.$$

2.6 Markov-Ketten

Wir betrachten eine Folge $X_0, X_1, \ldots$ von Zufallsvariablen, die Werte aus einer Menge $\mathcal{Z} = \{z_0, \ldots, z_n\}$ annehmen können. Wir nennen die Menge $\mathcal{Z}$ *Zustandsraum* und ihre Elemente *Zustände*. Die Zufallsvariable X_j interpretieren wir als den Zustand eines Systems im Schritt j. Die Folge $X_0, X_1, \ldots$ gibt uns eine natürliche Reihenfolge. Wir sagen X_{m+1} ist der *Nachfolger* von X_m. Die Folge der Zufallsvariablen heißt *Markov-Kette*, wenn die folgende Bedingung für alle m, i und j erfüllt ist: Befindet sich das System in Schritt m im Zustand z_i, so geht es mit einer Wahrscheinlichkeit von $p_{i,j}$ im Schritt $m+1$ in den Zustand z_j über. Das heißt, für alle m, $i_0, \ldots, i_{m-1}$, i und j gilt

$$\begin{aligned}\Pr[X_{m+1} = z_j \mid X_m = z_i, X_{m-1} = z_{i_{m-1}}, \ldots, X_0 = z_{i_0}] &= \Pr[X_{m+1} = z_j \mid X_m = z_i] \\ &= p_{i,j}\end{aligned}$$

2.6.1 Zustandsübergangmatrix

Die *Zustandsübergangs- oder Transitionswahrscheinlichkeiten* $p_{i,j}$ bilden die *Zustandsübergangs- oder Transitionsmatrix* P:

$$P = \begin{pmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,n} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,n} \\ \vdots & \vdots & & \vdots \\ p_{n,0} & p_{n,1} & \cdots & p_{n,n} \end{pmatrix}.$$

Sei $P^{(m)} = \overbrace{P \cdot P \cdot \ldots \cdot P}^{m\text{-mal}}$ und $p_{i,j}^{(m)}$ der (i,j)-te Eintrag von $P^{(m)}$. Dann gibt $p_{i,j}^{(m)}$ die Wahrscheinlichkeit an, dass sich das System beginnend im Zustand z_i nach m Schritten im Zustand z_j befindet. Somit gilt für $a \geq 0$:

$$p_{i,j}^{(m)} = \Pr[X_{a+m} = z_j \mid X_a = z_i].$$

2.6.2 Absorbierende Markov-Ketten

Ein *Zustand z_i einer Markov-Kette* heißt *absorbierend*, falls es unmöglich ist, diesen Zustand zu verlassen, das heißt, es gilt $p_{i,i} = 1$. Eine *Markov-Kette* heißt *absorbierend*, falls sie mindestens einen absorbierenden Zustand besitzt und falls es für jeden Zustand möglich ist, mit Wahrscheinlichkeit größer 0 in endlich vielen Schritten einen absorbierenden Zustand zu erreichen. Ein Zustand einer absorbierenden Markov-Kette, der nicht absorbierend ist, heißt *transient*.

Wir betrachten im Folgenden eine beliebige absorbierende Markov-Kette. Wir erhalten die *kanonische Form* der Markov-Kette, indem wir die Zustände so nummerieren, dass die transienten Zustände vor den absorbierenden Zuständen kommen. Demnach seien $z_0, \ldots, z_{t-1}$ die transienten und $z_t, \ldots, z_n$ die absorbierenden Zustände der Markov-Kette. Somit hat die Transitionsmatrix die kanonische Form

$$P = \begin{array}{c} \begin{array}{cc} \text{trans.} & \text{abs.} \end{array} \\ \left(\begin{array}{c|c} Q & B \\ \hline 0 & \mathrm{I} \end{array}\right) \end{array} \begin{array}{l} \\ \text{trans.} \\ \text{abs.} \end{array}.$$

Die Submatrizen beschreiben die Übergänge zwischen den transienten und absorbierenden Zuständen. Dabei ist I die $(n-t)\times(n-t)$-Einheitsmatrix, 0 eine $(n-t)\times t$-Nullmatrix, B eine $t\times(n-t)$-Nicht-Nullmatrix und Q eine $t\times t$-Matrix. Durch Potenzieren der Transitionsmatrix erhalten wir

$$P^{(m)} = \begin{array}{c} \begin{array}{cc} \text{trans.} & \text{abs.} \end{array} \\ \left(\begin{array}{c|c} Q^{(m)} & B' \\ \hline 0 & \mathrm{I} \end{array}\right) \end{array} \begin{array}{l} \\ \text{trans.} \\ \\ \text{abs.} \end{array} .$$

Die Submatrix $Q^{(m)}$ gibt die Übergangswahrscheinlichkeit an, im m-ten Schritt in einen entsprechenden transienten Zustand zu gelangen. Da wir aus jedem transienten Zustand mit positiver Wahrscheinlichkeit einen absorbierenden Zustand erreichen können, nähert sich jeder Eintrag von $Q^{(m)}$ mit zunehmendem m dem Wert 0 an.

Sei I die $t\times t$-Einheitsmatrix. Für eine absorbierende Markov-Kette gilt, dass die Matrix $I-Q$ invertierbar ist und

$$N = (I-Q)^{-1} = I + \sum_{i=1}^{\infty} Q^{(i)}.$$

Wir nennen N die *Fundamentalmatrix* der absorbierenden Markov-Kette. Der (i,j)-te Eintrag $n_{i,j}$ der Matrix N gibt die erwartete Anzahl von Schritten an, nach denen sich die Markov-Kette im Zustand z_j befindet, wenn sie in Zustand z_i beginnt. Sei nun s_i die erwartete Anzahl von Schritten, die die Markov-Kette beginnend im Zustand z_i vor der Absorption macht, und sei $s = (s_0,\ldots,s_{t-1})^{\mathsf{T}}$ und $c = (1,\ldots,1)^{\mathsf{T}} \in \mathbb{N}^t$. Dann gilt:

$$s = Nc.$$

Nehmen wir an, dass die Markov-Kette in Zustand z_i startet, dann ist die erwartete Anzahl von Schritten, bis die Markov-Kette erstmals einen absorbierenden Zustand erreicht,

$$\left(\sum_{j=0}^{t-1} n_{i,j}\right) + 1.$$

2.7 Berechnungsmodell in der Kryptographie

Kryptographische Verfahren sollen jedem realistischen Angriff standhalten. Für einen Angreifer bedeutet dies, dass er unter der Verwendung von allen praktisch einsetzbaren Methoden keine Chance hat, ein *kryptographisch sicheres* Verfahren zu brechen. In diesem Abschnitt beschreiben wir, wie die Begriffe realistisch und praktisch einsetzbar zu fassen sind. Wir möchten ein Modell, das nicht nur die aktuell bekannten, sondern alle realistischen Angriffe einbezieht. In der Komplexitätstheorie ist sehr genau untersucht, was praktisch einsetzbar bedeutet. In diesem Abschnitt werden wir daher auf einige Komplexitätsklassen zu sprechen kommen.

An dieser Stelle sei angemerkt, dass wir einen stärkeren Begriff der Sicherheit erhalten, wenn wir jeden berechenbaren Angriff zulassen. Verfahren, die gegen jene Angriffe sicher sind, nennen wir *informationstheoretisch* beziehungsweise *statistisch sicher*. In Abschnitt 2.13.2 wird das Modell der informationstheoretischen Sicherheit ausführlicher behandelt.

2.7.1 Turing-Maschinen und Algorithmen

Die *Turing-Maschine* ist das grundlegende Berechnungsmodell in der Komplexitätstheorie. Die *Übergangsrelation* einer Turing-Maschine beschreibt die Vorschrift zum Zustandswechsel sowie der Beschriftung und der Kopfbewegungen auf ihren Bändern. Ist die Übergangsrelation eine partielle Funktion, das heißt, dass es zu jedem Zeitpunkt höchstens einen möglichen Übergang gibt, so heißt die Turing-Maschine *deterministisch*, ansonsten *nicht-deterministisch*. Hat eine Turing-Maschine zusätzlich Zugriff auf ein Band mit gleichverteilten Zufallsbits, so nennen wir die Turing-Maschine *probabilistisch* oder *randomisiert*. Wir können uns dabei vorstellen, dass die Turing-Maschine in jedem Schritt ihrer Berechnung eine faire Münze wirft und das Ergebnis dieses Wurfs über eine erweiterte Übergangsrelation den Zustandsübergang beeinflusst. Für eine probabilistische Turing-Maschine betrachten wir ihre Ausgabe als Zufallsvariable und bezeichnen mit $\Pr[M(x) = y]$ die Wahrscheinlichkeit, dass die Maschine M auf Eingabe x den Wert y ausgibt. Eine *Polynomialzeit-Turing-Maschine* ist eine Turing-Maschine, die höchstens polynomiell viele Schritte in der Länge ihrer Eingabe ausführt. Für eine Polynomialzeit-Turing-Maschine M existiert daher ein Polynom p, so dass für alle Eingaben x die Laufzeit von M durch $p(|x|)$ begrenzt ist. Für eine probabilistische Polynomialzeit-Turing-Maschine ist die Anzahl der gelesenen Zufallsbits somit durch $p(|x|)$ beschränkt. Wir gehen ohne Einschränkung davon aus, dass sie so viele benutzt. In dieser Arbeit werden wir häufig den Begriff *Algorithmus* anstelle von Turing-Maschinen verwenden.

2.7.2 Effiziente Algorithmen

Klassisch werden effiziente Berechnungen mit der Klasse $\mathcal{P}$ in Verbindung gebracht. Die wohl größte offene Frage in der Komplexitätstheorie ist $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$, das heißt, ob die Klassen der deterministisch und nichtdeterministisch in Polynomialzeit erkennbaren Sprachen übereinstimmen. Diese Sichtweise auf effiziente Algorithmen spiegelt das exakte Lösen von Problemen wider. In der Kryptographie betrachten wir einen Angreifer als einen Algorithmus, der nicht immer die exakte Lösung finden muss und zudem Zufallsbits zur Verfügung hat. Daher fassen wir den Begriff der Effizienz etwas weiter und lassen zu, dass Algorithmen mit einer bestimmten Fehlerwahrscheinlichkeit nicht die richtige Lösung finden. Außerdem fassen wir zur Gruppe der effizienten Algorithmen auch die *probabilistische Polynomialzeitalgorithmen* und assoziieren effiziente Berechnungen mit der Komplexitätsklasse $\mathcal{BPP}$ (Bounded-Probability Polynomial Time).

Eine Sprache $L \subseteq \{0,1\}^*$ ist in $\mathcal{BPP}$, falls es einen probabilistischen Polynomialzeitalgorithmus $\mathcal{A}$ gibt, der L erkennt, das heißt,

1. für alle $x \in L$ gilt, dass
$$\Pr[\mathcal{A}(x) = 1] \geq \frac{2}{3},$$
2. für alle $x \notin L$ gilt, dass
$$\Pr[\mathcal{A}(x) = 0] \geq \frac{2}{3}.$$

Der Wert $\frac{2}{3}$ in der Definition lässt sich durch jede andere Konstante $c > \frac{1}{2}$ ersetzen, ohne dass sich die Klasse ändert. Wichtig für die Anwendung in der Kryptographie ist, dass sich c

auch durch den Wert $1-2^{-|x|}$ ersetzen lässt. Somit entspricht die Klasse $\mathcal{BPP}$ der Klasse von Sprachen, die von effizienten Algorithmen mit vernachlässigbar kleinem Fehler erkannt werden können.

Eine Funktion $f\colon \mathbb{N} \to \mathbb{R}$ nennen wir *vernachlässigbar*, falls es für jedes positive Polynom p einen Wert N gibt, so dass für alle $n > N$ gilt

$$f(n) < \frac{1}{p(n)}.$$

2.7.3 Nicht uniforme Schaltkreisfamilien als konservatives Angreifermodell

Kryptographische Verfahren sollen eine Berechnung vor Angriffen schützen. Machbare und damit realistische Angriffe können wir auf effiziente Algorithmen zurückführen. Sie dürfen randomisiert sein und müssen in Polynomialzeit berechenbar sein. Allerdings muss ein Verfahren, das wir sicher nennen, nicht nur bestimmten, sondern allen machbaren Angriffen widerstehen. Deshalb fassen wir das Berechnungsmodell für Angriffe etwas weiter:

Ein *Boolescher Schaltkreis* ist ein gerichteter azyklischer Graph, dessen innere Knoten durch die logischen Operatoren $\wedge$, $\vee$ und $\neg$ gekennzeichnet sind. Innere Knoten haben zwei ($\wedge$, $\vee$) oder eine ($\neg$) eingehende Kante und beliebig viele ausgehende Kanten. Knoten ohne eingehende Kanten nennen wir *Eingabeknoten*, Knoten ohne ausgehende Kanten nennen wir *Ausgabeknoten*. Die Knoten eines Schaltkreises bezeichnen wir als *Gatter*. Die Eingabe des Schaltkreises erfolgt durch Setzen der Eingabebits in den Eingabeknoten. Die Berechnung erfolgt durch sukzessives Auswerten der Operationen der inneren Knoten auf den Werten ihrer eingehenden Kanten. Das Ergebnis der Berechnung des Schaltkreises steht dann bitweise in den Ausgabeknoten. Die *Größe eines Schaltkreises* ist die Anzahl seiner Gatter.

Eine *nicht uniforme, polynomiell große Schaltkreisfamilie* $\{C_n\}_{n\in\mathbb{N}}$ ist eine unendliche Folge $C_1, C_2, \ldots$ von Booleschen Schaltkreisen, so dass der Schaltkreis C_n genau n Eingabebits hat und seine Größe polynomiell in n beschränkt ist. Nicht uniforme polynomiell große Schaltkreisfamilien entsprechen in ihrer Berechnungskapazität nicht uniformen Polynomialzeit-Turing-Maschinen. Diese können wir uns als Polynomialzeit-Turing-Maschinen vorstellen, die für jede Eingabelänge einen polynomiell langen String als zusätzliche Eingabe bekommen. Diesen String können wir als Hinweis eines externen Genies verstehen, der die Strategie der Maschine abhängig von der Länge der Eingabe steuert.

Nicht uniforme Schaltkreisfamilien können nicht alle Berechnungen ausführen, die probabilistische Polynomialzeitalgorithmen durchführen können. Wir können beispielsweise mit einer Schaltkreisfamilie keinen fairen Münzwurf umsetzen. Allerdings sind in den für uns interessanten Fällen (wie Entscheider, siehe Abschnitt 2.10) nicht uniforme, polynomiell große Schaltkreisfamilien mindestens genauso mächtig wie probabilistische Polynomialzeitalgorithmen. Für die Klasse $\mathcal{P}/\text{poly}$ der Sprachen, die durch nicht uniforme Polynomialzeit-Turing-Maschinen (nicht uniforme, polynomiell große Schaltkreisfamilien) erkannt werden können, gilt, dass $\mathcal{BPP} \subseteq \mathcal{P}/\text{poly}$. Speziell, wenn wir zeigen können, dass Angriffe in dem Modell von nicht uniformen, polynomiell großen Schaltkreisfamilien keinen Erfolg haben, dann ist dies auch gültig für alle Strategien von machbaren Angreifern. Man mag mit Recht sagen, dass dieses Berechnungsmodell unrealistisch ist, jedoch umfasst es jede machbare Angriffsstrategie und zeichnet sich deswegen als konservatives Berechnungsmodell für Angreifer aus.

2.8 Unlösbarkeitsannahmen bei kryptographischer Sicherheit

In der Kryptographie nutzen wir Aussagen wie: „Falls die `praktische Unlösbarkeitsannahme` stimmt, dann gilt unsere `nützliche Aussage`“. Als praktisch unlösbar betrachten wir alle Probleme, die nicht von probabilistischen Polynomialzeitalgorithmen gelöst werden können. Da wir hier effiziente Algorithmen mit der Gruppe der probabilistischen Polynomialzeitalgorithmen identifizieren, ist für uns die größte offene Frage, ob $\mathcal{NP} \not\subseteq \mathcal{BPP}$, welches $\mathcal{P} \neq \mathcal{NP}$ impliziert. Nur in diesem Fall ist unsere Herangehensweise an die kryptographische Sicherheit interessant, denn dann greift unsere `praktische Unlösbarkeitsannahme`. Wir müssen Aussagen in solcher Form verwenden, da unsere `nützlichen Aussagen` entweder die `praktische Unlösbarkeitsannahme` impliziert oder Aussagen der Form $\mathcal{NP} \setminus \mathcal{BPP} \neq \emptyset$. Somit können wir ohne Beantwortung der offenen komplexitätstheoretischen Fragen die `nützliche Aussage` nicht unabhängig von einer Unlösbarkeitsannahme machen.

2.9 Interaktive Algorithmen und Protokolle

In dieser Arbeit interessieren wir uns hauptsächlich für Berechnungen in interaktivem Kontext. Die Berechnungen werden nicht allein von einer Entität durchgeführt, sondern gemeinsam von mehreren unabhängigen. Diese voneinander getrennten Entitäten nennen wir *Parteien*. Jede Partei hat eine private Eingabe und kann beliebige effiziente Algorithmen ausführen. Die Parteien stehen paarweise untereinander über sogenannte *Kanäle* in Verbindung. Über Kanäle fließen *Nachrichten*. Zudem haben die Parteien Zugriff auf ein *schwarzes Brett* oder einen *Broadcast-Kanal* über das/den sie Nachrichten an alle Parteien gleichzeitig übertragen können. Soweit nicht anders gesagt, gehen wir davon aus, dass die Kanäle sicher sind. Nur Sender und Empfänger eines Kanals haben Zugriff auf diesen und auf seine Nachrichten.

Formal können wir die Parteien in diesem Szenario durch *interaktive Algorithmen* beziehungsweise *interaktive Turing-Maschinen* beschreiben [41, 42, 3]. Eine interaktive Turing-Maschine besitzt zusätzlich eine Identität und mehrere Kommunikationsbänder zum Senden und Empfangen von Nachrichten. Diese Bänder werden mit den entsprechenden Bändern der anderen Parteien verbunden und bilden somit die Kanäle. Neben ihrem individuellen Eingabeband haben die interaktiven Turing-Maschinen Zugriff auf ein gemeinsames Eingabeband. Die Eingabe einer interaktiven Turing-Maschine besteht somit aus der individuellen (privaten) und der gemeinsamen Eingabe. Ein *(interaktives) Protokoll* ist eine Menge von interaktiven Algorithmen, die die Strategien der teilnehmenden Parteien beschreiben. Wir können uns ein Protokoll als eine Vorschrift für die Parteien vorstellen, die angibt, welche Berechnungen die Parteien durchführen sollen und welche Nachrichten sie zu senden haben.

2.10 Wahrscheinlichkeits-Ensembles und ihre Ununterscheidbarkeit

Sei I eine abzählbare Indexmenge. Ein von I indiziertes *(Wahrscheinlichkeits-)Ensemble* $X = \{X\}_{i \in I}$ ist eine Sequenz von Zufallsvariablen X_i mit $i \in I$.

Für I benutzen wir natürliche Zahlen oder Strings, das heißt, eine Teilmenge von $\{0,1\}^*$. Ist $I = \mathbb{N}$, dann nehmen die Zufallsvariablen X_i typischerweise Werte aus den Strings der

Länge höchstens polynomiell in i an. Für $I \subseteq \{0,1\}^*$ gelte analog, dass die Zufallsvariable X_w des Ensembles $\{X_w\}_{w\in\{0,1\}^*}$ Werte aus den Strings der Länge polynomiell in $|w|$ annehmen. Indem wir einer natürlichen Zahl i ihre unäre Darstellung 1^i zuordnen und diese als String interpretieren, betrachten wir im Folgenden nur Indexmengen über Strings.

Der Begriff der *Ununterscheidbarkeit von Ensembles* spielt in der Definition der Privatheit und Sicherheit eine zentrale Rolle. Über sie können wir beliebige randomisierte Angreifer unabhängig von ihrer Strategie betrachten und Aussagen darüber machen, ob die Angreifer durch eine Attacke einen Informationsvorteil gegenüber einfachem Raten bekommen können.

Sei $I \subseteq \{0,1\}^*$. Zwei Ensembles $X = \{X_w\}_{w\in S}$ und $Y = \{Y_w\}_{w\in S}$ heißen *ununterscheidbar* ($X \equiv Y$), falls für alle $w \in S$ und beliebige α gilt:

$$|\Pr[X_w = \alpha] - \Pr[Y_w = \alpha]| = 0.$$

Zwei Ensembles $X = \{X_w\}_{w\in S}$ und $Y = \{Y_w\}_{w\in S}$ heißen *statistisch ununterscheidbar* ($X \stackrel{s}{\equiv} Y$), falls für alle $w \in S$ und alle positiven Polynome p gilt:

$$\sum_{\alpha} |\Pr[X_w = \alpha] - \Pr[Y_w = \alpha]| < \frac{1}{p(|w|)}.$$

Zwei Ensembles $X = \{X_w\}_{w\in S}$ und $Y = \{Y_w\}_{w\in S}$ heißen *in Polynomialzeit ununterscheidbar*, falls für alle probabilistischen Polynomialzeitalgorithmen $\mathcal{A}$, für alle positiven Polynome p und alle hinreichend langen Strings $w \in S$ gilt:

$$|\Pr[\mathcal{A}(X_w, w) = 1] - \Pr[\mathcal{A}(Y_w, w) = 1]| < \frac{1}{p(|w|)}.$$

Die Ensembles X und Y heißen *ununterscheidbar von polynomiell großen Schaltkreisen*, wir sagen auch *nicht effizient unterscheidbar* ($X \stackrel{c}{\equiv} Y$), falls für alle nicht uniformen, polynomiell großen Schaltkreisfamilien $\{C_n\}_{n\in\mathbb{N}}$, für alle positiven Polynome p und alle hinreichend langen Strings $w \in S$ gilt:

$$|\Pr[C_{|w|}(X_w) = 1] - \Pr[C_{|w|}(Y_w) = 1]| < \frac{1}{p(|w|)}.$$

Der Unterschied zwischen X und Y ist somit vernachlässigbar und kann von probabilistischen Polynomialzeitalgorithmen (polynomiell großen Schaltkreisen) nicht erkannt werden. Wir wollen an dieser Stelle bemerken, dass die Bedingung der zweiten Definition äquivalent ist zu der Forderung, dass

$$|\Pr[C_w(X_w) = 1] - \Pr[C_w(Y_w) = 1]| < \frac{1}{p(|w|)}.$$

Ununterscheidbare Ensembles sind statistisch ununterscheidbar. Statistisch ununterscheidbare Ensembles sind nicht effizient unterscheidbar. Wir betrachten an dieser Stelle nur Algorithmen / Schaltkreise, die die Werte 0 oder 1 ausgeben (Entscheider). Dies bedeutet keine Einschränkung für Algorithmen, die beliebige Strings liefern, da wir für solche Algorithmen einen entsprechenden Entscheider konstruieren können. Wir können uns einen praktischen Angreifer auch als eine Reihe von randomisierten Polynomialzeitalgorithmen (höchstens aber polynomiell in $|w|$ viele) vorstellen, die jeweils nach einem Merkmal suchen und somit Entscheider sind.

2.11 Interaktive Beweissysteme und Zero-Knowledge

Im Gegensatz zu einem klassischen, niedergeschriebenen Beweis ist ein interaktiver Beweis ein Prozess zwischen Parteien, durch den die Gültigkeit einer Aussage nachgewiesen wird. Dabei erzeugt eine Partei, der *Beweiser P*, einen Beweis zu einer Aussage, von der er den *Verifizierer V* in einem Protokoll überzeugt.

Ein *interaktives Beweissystem für eine Sprache* $L \subseteq \{0,1\}^*$ ist ein Paar von interaktiven Algorithmen (P,V) und eines Protokolls $\mathcal{P}$, so dass V ein probabilistischer Polynomialzeitalgorithmus ist und es gilt:

Vollständigkeit:

Für alle $x \in L$ gilt:

$$\Pr[\text{OUTPUT}_V^{\mathcal{P}}\langle P\rangle = 1] \geq \frac{2}{3} \text{ und}$$

Zuverlässigkeit:

Für alle $x \notin L$ und jeden Algorithmus B gilt:

$$\Pr[\text{OUTPUT}_V^{\mathcal{P}}\langle B\rangle = 1] \leq \frac{1}{3}.$$

Dabei bezeichnet $\text{OUTPUT}_V^{\mathcal{P}}\langle A\rangle$ die Zufallsvariable der Ausgabe von V bei Interaktion mit dem Algorithmus A über das Protokoll $\mathcal{P}$. Die Wahrscheinlichkeiten berechnen sich über alle Eingaben und Zufallsbits.

Wir schränken die Berechnungskapazität des Beweisers nicht ein, da wir in einem interaktiven Beweissystem nur betrachten, dass der Verifizierer von einem Beweiser auch überzeugt wird (Vollständigkeit) und von einem „falschen“ Beweiser nicht betrogen wird (Zuverlässigkeit). Führen wir polynomiell viele Wiederholungen durch, so wird die Fehlerwahrscheinlichkeit vernachlässigbar klein.

Besondere interaktive Beweissysteme sind *Zero-Knowledge-Beweise*. *Zero-Knowledge* bezeichnet eine Eigenschaft des Systems bezüglich des Beweisers: Durch die Interaktion mit dem Beweiser eines Zero-Knowledge-Beweises können wir kein weiteres Wissen über den Beweiser gewinnen als durch die (gemeinsame) Eingabe.

Sei (P,V) mit $\mathcal{P}$ ein interaktives Beweissystem für eine Sprache L. Die Zufallsvariable $\text{VIEW}_{V^*}^{\mathcal{P}}\langle P\rangle(x)$ beschreibt die Zufallsbits und alle erhaltenen Nachrichten des Verifizierers V^* bei Interaktion mit P durch $\mathcal{P}$ auf der gemeinsamen Eingabe x. Wir bezeichnen dieses Beweissystem als *zero-knowledge*, falls es für jeden probabilistischen Polynomialzeitalgorithmus V^* einen probabilistische Polynomialzeitalgorithmus M^* gibt, so dass

$$\{\text{VIEW}_{V^*}^{\mathcal{P}}\langle P\rangle(x)\}_{x\in\{0,1\}^*} \stackrel{c}{\equiv} \{M^*(x)\}_{x\in\{0,1\}^*},$$

wobei $M^*(x)$ die Zufallsvariable ist, die die Ausgabe von M^* auf Eingabe x beschreibt.

Wir können also die Kommunikation eines Zero-Knowledge-Beweises mit Kenntnis der Eingabe effizient simulieren ohne auf das Wissen von P zurückzugreifen. Somit gibt die Kommunikation keine oder nur vernachlässigbare Information über die Geheimnisse von P.

2.12 Angreifertypen und Rollen der Parteien

In diesem Abschnitt wollen wir darauf eingehen, wie sich die Parteien bei der Durchführung eines Protokolls verhalten und ihre Angriffsarten klassifizieren. Anschließend werden wir typische Rollen für Angreifertypen und ausgezeichnete Aufgaben benennen.

2.12.1 Angreifertypen

Wir können nicht immer davon ausgehen, dass alle Parteien dem Protokoll folgen und sich nur für die gemeinsame und die eigene Eingabe sowie die Ausgabe des Protokolls interessieren. Eine Partei, die auf diese ideale Art einem Protokoll folgt, nennen wir eine *ehrliche Partei*. Insbesondere darf eine ehrliche Partei ihre Eingabe und ihren Kommunikationsstring nur entsprechend dem Protokoll analysieren. Im Normalfall entsprechen die Parteien meistens leider nicht diesem Ideal.

Wir sprechen bei jeder *nicht ehrlichen Partei* von einem *Angreifer*.

Im besten Fall ist ein Angreifer *halb ehrlich* oder *passiv*. Dieser Angreifer befolgt ein Protokoll wie eine ehrliche Partei, versucht jedoch aus seinen Ein- und Ausgaben sowie seinem Kommunikationsstring Information über die Eingaben der anderen Parteien zu gewinnen. Umgangssprachlich können wir den Begriff der *Privatheit* eines Protokolls wie folgt fassen:

Ein Protokoll nennen wir *privat*, falls ein halb ehrlicher Angreifer aus seinem Kommunikationsstring kein weiteres Wissen über die Eingaben anderer Parteien gewinnen kann als er aus seiner Ein- und Ausgabe berechnen kann. Eine formale Definition von Privatheit und Sicherheit folgt in 2.13.

Weicht ein Angreifer hingegen von der Protokollvorschrift ab, so nennen wir ihn *bösartig* oder *aktiv*. Ein bösartiger Angreifer kann beliebig Nachrichten verändern, löschen oder hinzufügen und senden. Wir nennen ein Protokoll *sicher*, wenn ein aktiver Angreifer aus seinem Kommunikationsstring kein weiteres Wissen über die Eingaben anderer Parteien bekommen als er aus seiner Ein- und Ausgabe berechnen kann. Ein Angriff ist *erkennbar (detektierbar)*, wenn zumindest eine (halb-)ehrliche Partei den Angriff feststellt. Ein Protokoll heißt *robust* gegen einen aktiven Angriff, falls dieser die Ausgabe der (halb-)ehrlichen Parteien nicht verändert.

Insbesondere wenn mehr als zwei Parteien an einem Protokoll teilnehmen, müssen wir in Betracht ziehen, dass mehrere Angreifer zusammenarbeiten. Arbeiten t Angreifer zusammen, so nennen wir sie eine *t-Koalition*. Wir gehen davon aus, dass eine Koalition wie ein Angreifer agiert, der alle Parteien der Koalition steuert und betrachten die Gesamtheit aller Eingaben, Ausgaben und Kommunikationsstrings der Koalition als Eingabe, Ausgabe und Kommunikationsstrings des Angreifers. Analog zu den Definitionen oben nennen wir eine Koalition halb ehrlich/passiv beziehungsweise bösartig/aktiv. Ein Angreifer, der eine passive oder aktive t-Koalition kontrolliert, heißt *adaptiv*, wenn er im Laufe des Protokolls die Mitglieder der Koalition wählt. Dieser Angreifer kann sich während der Protokolldurchführung nach und nach für t Parteien entscheiden, die er kontrollieren möchte. Die Erweiterungen für die Privatheit, Sicherheit und Robustheit von Protokollen gelten entsprechend den obigen Definitionen.

Wir können passive und aktive Angreifer auch im Zusammenhang mit Kommunikationskanälen betrachten. Passive *Angreifer eines Kommunikationskanals* können die Nachrichten auf dem Kanal abhören, aktive können Nachrichten verändern, löschen oder hinzufügen. Wir nehmen an, dass die Kanäle gegen diese Angreifer gesichert sind.

2.12.2 Typische Rollen der Parteien

Bei der Definition von kryptographischen Primitiven ist es üblich, den einzelnen teilnehmenden Parteien Namen zu geben und mit diesen Namen ein bestimmtes Verhalten zu assoziieren.

Alice, Bob, Cecille und Dave

Diese Rollen bezeichnen die Teilnehmer eines Protokolls, um die Beschreibung des Szenarios anschaulich zu halten. Insbesondere wenn die Aufgaben in dem Protokoll unterschiedlich sind, bietet es sich an, diese Personen anstatt einer abstrakten Identität zu verwenden. Normalerweise beginnt Alice die Ausführung des Protokolls. Bob, Cecille und Dave sind die folgenden Akteure. Bei der Analyse der Sicherheit gehen wir in den meisten Fällen davon aus, dass diese Personen halb ehrlich sind.

Eve

Eine Partei, die wir mit Eve identifizieren, ist keine Partei, die am Protokoll teilnimmt, sondern ein passiver Angreifer auf die Kommunikationskanäle. Eve entspricht einem lauschenden oder abhörenden Dritten. Sprechen wir in dieser Arbeit von Eve, so gehen wir im Gegensatz vom Normalfall implizit von unsicheren Kommunikationskanälen aus.

Mallory

Mallory kennzeichnet explizit einen aktiven Angreifer und ersetzt eine der zuvor genannten Rollen. Er kann verschiedene Ziele verfolgen. Typischerweise versucht Mallory, Wissen über die privaten Daten der anderen Parteien zu gewinnen. Dabei kann er rücksichtslos oder vorsichtig vorgehen, so dass seine Angriffe entdeckt werden dürfen oder nicht entdeckt werden sollen. Darüber hinaus könnte Mallory auch versuchen, destruktiv zu wirken und den Fortschritt der Berechnung zu stören oder zu verfälschen.

Trusted Third Party, Trent

Die *Trusted Third Party* Trent (auch Trusted Party) ist eine herausgestellte ehrliche Partei ohne Ein- und Ausgabe. Da Trent keine Ausgabe hat und ehrlich ist, erlangt sie keine Information über die Eingaben der anderen Parteien. Da alle teilnehmenden Parteien eines Protokolls dieser Partei vertrauen können, kommen ihr in manchen Protokollen wichtige Aufgaben in Bezug auf die Wahrung der Sicherheit zu.

2.13 Privatheit und Sicherheit von Protokollen

Ein Protokoll ermöglicht es, dass mehrere Parteien zusammen eine Funktionalität über ihren Eingaben berechnen können. Durch die Durchführung eines Protokolls erhalten die Parteien ihren Kommunikationsstring. Aus diesem Kommunikationsstring und ihrer Eingabe kann jede Partei ihre persönliche Ausgabe berechnen. Idealerweise soll eine Partei nur ihre Eingabe und ihre Ausgabe kennen und nichts Weiteres über die Daten der anderen Parteien lernen. Der Kommunikationsstring ist die einzige Informationsquelle, aus der eine Partei weiteres Wissen über die Daten anderer gewinnen kann. Wenn dieser String nicht mehr Wissen als die Ein- und Ausgabe liefert, gibt die Durchführung des Protokolls auch kein zusätzliches Wissen. Genau dieses fordern wir von privaten beziehungsweise sicheren Protokollen.

In den meisten Fällen, auch bei sicheren Protokollen, ist es so, dass die Parteien Wissen über die Ein- und Ausgaben der anderen Parteien lernen können. Dieses Wissen erhalten sie bereits daher, dass ihnen ihre Ausgabe und die Funktionalität bekannt sind. Beispielsweise weiß eine Partei bei einem berechneten logischen `OR` über die Eingabebits aller Parteien

und der Ausgabe 0, dass alle Eingabebits 0 sein müssen. Der Ausdruck „weiteres Wissen“ bedeutet hier also alles an Wissen, was eine Partei über ihre Ein- und Ausgabe hinaus gewinnen kann.

Wir werden in diesem Abschnitt den Begriff der Privatheit und Sicherheit von Protokollen formalisieren und wichtige Techniken aufzeigen, wie wir die Sicherheit von Protokollen zeigen können.

2.13.1 Ideale Funktionalitäten von Protokollen

Sei $\Sigma = \{0,1\}^*$ die Menge aller Binärstrings. Unter *Funktionalitäten* verstehen wir randomisierte Erweiterungen von Abbildungen. Wir können uns eine n-stellige Funktionalität $\mathcal{F}(x_1,\ldots,x_n)$ vorstellen, indem wir zunächst einen gleichverteilten Zufallsstring r wählen und den Wert einer $(n+1)$-stelligen Funktion $\mathcal{F}'(r,x_1,\ldots,x_n)$ zurückgeben.

Wir spezifizieren ein *n-Parteien-Protokoll-Problem*, indem wir einen Zufallsprozess beschreiben, der den Eingaben der n Parteien ihre entsprechenden Ausgaben zuordnet. Dazu geben wir die *ideale Funktionalität* $\mathcal{F}\colon \Sigma^n \to \Sigma^n$ als eine randomisierte Abbildung der Sequenz $\bar{x} = (x_1,\ldots,x_n)$ auf die Sequenz von Zufallsvariablen $\mathcal{F}(\bar{x}) = (\mathcal{F}(\bar{x})_1,\ldots,\mathcal{F}(\bar{x})_n)$ an. Wir interpretieren x_i als Eingabe der Partei i sowie $\mathcal{F}(\bar{x})_i$ als ihre Ausgabe. Für $I = \{i_1,\ldots,i_t\} \subseteq \{1,\ldots,n\}$ bezeichnet $\mathcal{F}(\bar{x})_I$ die Subsequenz $\mathcal{F}(\bar{x})_{i_1},\ldots,\mathcal{F}(\bar{x})_{i_t}$.

Wir sagen ein Protokoll $\mathcal{P}$ *berechnet* $\mathcal{F}$ oder auch *ist ein Protokoll für* $\mathcal{F}$, falls die Verteilung der Ausgabe von $\mathcal{P}$ über Eingaben $\bar{x}$ bei passiven Angreifern und $\mathcal{F}(\bar{x})$ identisch verteilt sind.

Sei $\mathcal{P}$ ein Protokoll für die Funktionalität $\mathcal{F}$. Die *Sicht der Partei i bei der Ausführung von* $\mathcal{P}$ *auf* $\bar{x}$ bezeichnen wir mit $\text{VIEW}_i^{\mathcal{P}}(\bar{x}) = (x_i, r, m_1, \ldots, m_t)$. Sie besteht aus der Eingabe x_i, den Zufallsbits r der Partei i sowie allen Nachrichten m_j, die die Partei i während der Ausführung von $\mathcal{P}$ auf Eingabe $\bar{x}$ gesendet oder erhalten hat. Die *Ausgabe der Partei i bei der Ausführung von* $\mathcal{P}$ *auf* $\bar{x}$ bezeichnen wir mit $\text{OUTPUT}_i^{\mathcal{P}}(\bar{x})$. Jede Partei ist in der Lage, ihre Ausgabe aus ihrer Sicht zu berechnen. Sei $\text{OUTPUT}^{\mathcal{P}}(\bar{x}) = (\text{OUTPUT}_1^{\mathcal{P}}(\bar{x}),\ldots,\text{OUTPUT}_n^{\mathcal{P}}(\bar{x}))$ und für $I = \{i_1,\ldots,i_t\}$ sei $\text{VIEW}_I^{\mathcal{P}}(\bar{x}) = (I, \text{VIEW}_{i_1}^{\mathcal{P}}(\bar{x}),\ldots,\text{VIEW}_{i_t}^{\mathcal{P}}(\bar{x}))$. Im Folgenden benutzen wir $\mathcal{F}(\bar{x})_i$, wenn wir uns für die zu berechnende Funktionalität interessieren und $\text{OUTPUT}_i^{\mathcal{P}}(\bar{x})$, wenn das Protokoll analysiert werden soll.

Durch die ideale Funktionalität beschreiben wir das beabsichtigte Ein- und Ausgabeverhalten, welches ein Protokoll zeigen soll. Damit geben wir auch das erlaubte Wissen an, das jede Partei gewinnen soll: jeweils nur ihre Ein- und Ausgabe. Wir können die ideale Funktionalität in die Welt der Protokolle übertragen. Die *ideale Welt* besteht aus den n teilnehmenden Parteien sowie einer Trusted Third Party. Die Parteien senden ihre Eingaben an die Trusted Third Party. Diese berechnet die ideale Funktionalität. Zum Schluss sendet die Trusted Third Party die entsprechenden Ausgaben an jede Partei. Wir sehen, dass der Kommunikationsstring einer teilnehmenden Partei genau ihrer Ein- und Ausgabe entspricht. Also gibt dieser String der Partei (oder einer Koalition von Parteien) offensichtlich dieselbe Information wie ihre Ein- und Ausgabe. Dem entgegen setzen wir die *reale Welt*, in der die n teilnehmenden Parteien ohne eine Trusted Third Party nach einem Protokoll dasselbe Problem lösen. Wenn die Parteien in der realen Welt dasselbe Wissen wie in der idealen Welt bekommen, „emulieren“ die n Parteien die Trusted Third Party. Wir können dann von einem sicheren Protokoll sprechen. Abhängig von Parametern wie dem Angreifertyp und der Stufe der Emulation der Trusted Third Party beschreiben wir nachfolgend wichtige Modelle für die Sicherheit von Protokollen. Die ersten zufriedenstellenden Definitionen der Sicherheit

von Mehrparteien-Protokollen gehen zurück auf Micali und Rogaway [77] sowie auf Beaver [4, 5]. Durch die Arbeit von Canetti [18] wurden sie erweitert.

2.13.2 Perfekte Sicherheit

Wir behandeln zunächst die perfekte (informationstheoretische) Sicherheit, da dies der strengste Begriff der Sicherheit ist. Häufig ist diese leichter zu zeigen als kryptographische Sicherheit, da wir hier Wissen mit Information gleichsetzen können und die Werkzeuge aus der Informationstheorie einsetzen können (siehe zum Beispiel [72, 12]).

Eines der Werkzeuge ist die *Entropie*. Sie misst die Unsicherheit im Zusammenhang mit einer Zufallsvariable. Für eine diskrete Zufallsvariable X, die Werte in $\mathcal{X}$ annimmt, ist die Entropie $H(X)$ definiert durch

$$H(X) = -\sum_{x \in \mathcal{X}} \Pr[X = x] \cdot \log\left(\Pr[X = x]\right).$$

Claude E. Shannon zeigte in [102], dass die Funktionen aus der Klasse $K \cdot H(X)$ mit $K > 0$ die einzigen sind, die den Anforderungen eines Informationsmaßes für eine diskrete Informationsquelle genügen. Es gilt, dass $0 \leq H(X) \leq \log\left(|\mathcal{X}|\right)$. Im Fall $H(X) = 0$ haben wir vollständige Information über den Ausgang des von X beschriebenen Zufallsexperiments. Falls $H(X) = \log|(\mathcal{X})|$, dann sind wir maximal unsicher. Gegeben zwei Zufallsvariablen X und Y mit Wertebereichen $\mathcal{X}$ und $\mathcal{Y}$, so ist die *bedingte Entropie* $H(X \mid Y)$ definiert durch

$$H(X \mid Y) = -\sum_{y \in Y} \sum_{x \in X} \Pr[Y = y] \cdot \Pr[X = x \mid Y = y] \cdot \log\left(\Pr[X = x \mid Y = y]\right).$$

Es gilt $H(X \mid Y) = 0$, wenn der aus $\mathcal{Y}$ gezogene Wert den Ausgang des Experiments von X vollständig bestimmt. Falls $H(X \mid Y) = H(X)$, dann sind X und Y unabhängig. Die bedingte Entropie $H(X \mid Y)$ beschreibt die Unsicherheit über X, wenn wir den Ausgang für Y kennen. Die *Mutual Information* $I(X;Y)$ gegeben durch

$$\begin{aligned} I(X;Y) &= H(X) - H(X \mid Y) \\ &= H(Y) - H(Y \mid X) = I(Y;X) \end{aligned}$$

misst die Information, die Y über X beziehungsweise X über Y gibt. Wir können auch sagen, dass die Mutual Information ein Maß für die Reduktion der Unsicherheit ist. Nur wenn $I(X;Y) = 0$, dann gibt Y keine Information über X.

Sei Z eine diskrete Zufallsvariable mit Werten in $\mathcal{Z}$, dann ist die *bedingte Mutual Information* $I(X;Y \mid Z)$ definiert durch

$$I(X;Y \mid Z) = H(X \mid Z) - H(X \mid YZ).$$

und misst die Information, die Y über X bei Kenntnis von Z gibt.

Nun können wir die informationstheoretische Privatheit eines n-Parteien-Protokolls definieren. Für jede teilnehmende Partei i bestimme die Zufallsvariable X_i die entsprechende Eingabe. Sei $\mathcal{F}$ eine Funktionalität für ein n-Parteien-Protokoll-Problem. Sei F_i die Zufallsvariable, die die Ausgabe für Partei i von $\mathcal{F}$ abhängig von den Eingaben $X_1, \ldots, X_n$ angibt. Die Zufallsvariable R_i beschreibt die Zufallsbits, die von Partei i verwendet werden. Für $1 \leq t \leq n-1$ sagen wir, dass ein (randomisiertes) Protokoll $\mathcal{P}$ ein *perfekt (informationstheoretisch) t-privates Protokoll* zur Berechnung von $\mathcal{F}$ ist, falls folgende Eigenschaften erfüllt sind:

Berechnung:

Für jede Partei i gilt $H(F_i \mid C_i X_i) = 0$, wobei C_i der erhaltene Kommunikationsstring von i bei der Durchführung von $\mathcal{P}$ ist. Aus der Kommunikation und ihrer Eingabe kann jede Partei ihre Ausgabe bestimmen, das heißt, $\mathcal{P}$ *berechnet* $\mathcal{F}$.

Privatheit:

Für alle $K, E \subseteq \{1,\ldots,n\}$ mit $|K| = t$ und $E \cap K = \emptyset$ gilt:

$$I(X_E\,; C_K \mid X_K F_K) = 0.$$

Dabei bezeichnet A_B die Zufallsvariable bestehend aus der Folge der Zufallsvariablen A_j mit $j \in B$. Somit kann jede Koalition K von t passiven Angreifern aus der Kommunikation keine weitere Information über die Eingaben der anderen (ehrlichen) Parteien in E bekommen.

Falls die zweite Bedingung für Koalitionen von aktiven Angreifern gilt, so nennen wir das Protokoll $\mathcal{P}$ *perfekt (informationstheoretisch) t-sicher*. Wenn die erste Bedingung bei einer aktiven t-Koalition gilt, dann heißt $\mathcal{P}$ *t-robust*.

Die zweite Bedingung ist äquivalent mit der Forderung $I(X_E\,; C_K \mid R_K X_K F_K) = 0$. Die Zufallsbits R_K der Koalition müssen wir nicht betrachten, da sie unabhängig von X_E gewählt werden. Bei aktiven Angreifern könnte man entgegenhalten, dass sie ihre Zufallsbits nach einer anderen Verteilung oder adaptiv wählen. Dieses können wir jedoch auch so modellieren, dass aktive Angreifer ihren Zufallsstring komplett ignorieren und adaptiv neue Zufallsbits generieren. Somit kann nur der Kommunikationsstring C_K Information über die Daten der anderen Parteien enthalten.

Wir können die zweite Bedingung alternativ für deterministische Funktionalitäten, das heißt Funktionen, definieren [59]. Sei $\mathcal{F}$ eine Funktion, die für ein n-Parteien-Protokoll-Problem die Eingaben auf die entsprechenden Ausgaben abbildet, und sei $\mathcal{P}$ ein Protokoll, das $\mathcal{F}$ berechnet. Das Protokoll $\mathcal{P}$ ist *t-privat*, falls für alle Strings c und r, für alle unterschiedlichen Eingaben $\overline{x} = (x_1,\ldots,x_n)$ und $\overline{y} = (y_1,\ldots,y_n)$ mit $\overline{x}_K = \overline{y}_K$ und $\mathcal{F}(\overline{x})_K = \mathcal{F}(\overline{y})_K$ gilt, dass

$$\Pr[C_K = c \mid R_K = r, X = \overline{x}] = \Pr[C_K = c \mid\ R_K = r, X = \overline{y}].$$

Somit folgt die Privatheit eines Protokolls bereits, wenn wir zeigen können, dass der Kommunikationsstring C_k für alle Eingaben gleichverteilt unter allen Strings ist. Dieses ist in vielen Fällen gut nachzuweisen.

2.13.3 Simulation der Kommunikation

Nun widmen wir uns der kryptographischen Sicherheit. Zuerst stellen wir die Definition der Privatheit gegenüber passiven Angreifern durch Simulation der Kommunikation vor. Im sich anschließenden Abschnitt geben wir eine Definition der Sicherheit an, die die Emulation der idealen Welt durch ein Protokoll in der realen Welt beschreibt. Für passive Angreifer sind beide Definitionen äquivalent [42].

Wie bereits erwähnt beruht die Privatheit eines Protokolls darauf, dass ein Angreifer kein weiteres Wissen über die Eingaben der anderen Parteien bekommen kann. Wenn wir aus der Ein- und Ausgabe eines passiven Angreifers einen String effizient generieren können, der ununterscheidbar vom tatsächlichen Kommunikationsstring ist, dann können wir jedes

Wissen eines Angreifers nur aus seiner Ein- und Ausgabe „simulieren“. Folglich kann der Angreifer kein weiteres Wissen aus der Kommunikation gewinnen.

Wir sagen $\mathcal{P}$ *berechnet* $\mathcal{F}$ *(kryptographisch) privat*, falls es einen probabilistischen Polynomialzeitalgorithmus S (*Simulator*) gibt, so dass für alle $I \subseteq \{1,\ldots,n\}$ gilt:

$$\{(S(I,(x_{i_1},\ldots,x_{i_t}),\mathcal{F}(\overline{x})_I))\}_{\overline{x}\in\Sigma^m} \stackrel{c}{\equiv} \{(\text{VIEW}_I^{\mathcal{P}}(\overline{x}),\text{OUTPUT}^{\mathcal{P}}(\overline{x}))\}_{\overline{x}\in\Sigma^m}.$$

Falls wir I in der Größe durch t beschränken, so nennen wir $\mathcal{P}$ *(kryptographisch) t-privat*.

2.13.4 Emulation der idealen Welt in der realen Welt

Im Folgenden betrachten wir die kryptographische Sicherheit gegenüber (nicht adaptiven) aktiven Angreifern. Wir gehen davon aus, dass mehrere Parteien unter der Kontrolle eines Angreifers stehen und auf seine Anweisung vom Protokoll abweichen können. Für unsere Zwecke genügt der nicht adaptive Fall, da wir unsere Mehr-Parteien-Protokolle auf den Zwei-Parteien-Fall zurückführen können. Wir verweisen für die Definitionen der Sicherheit für adaptive Angreifer auf [18].

Sei $\mathcal{F}$ eine Funktionalität und $\mathcal{P}$ ein Protokoll für $\mathcal{F}$. Wir beschreiben zunächst das *Modell der realen Welt*. Die Parteien rechnen und kommunizieren, wie in den vorherigen Abschnitten beschrieben. Der aktive Angreifer $\mathcal{A}$ der realen Welt kontrolliert eine Koalition $K \subseteq \{1,\ldots,n\}$ von teilnehmenden Parteien. $\mathcal{A}$ ist ein probabilistischer Polynomialzeitalgorithmus und hat Zugriff auf alle Daten der kontrollierten Parteien. Außerdem bekommt er eine *Hilfseingabe*, die beispielsweise externes Wissen repräsentiert. Zum Ende der Durchführung des Protokolls berechnet der Angreifer seine Ausgabe aus seiner Sicht $\text{VIEW}_{\mathcal{A}}^{\mathcal{P}}$. Diese besteht aus allen Sichten der kontrollierten Parteien sowie der Hilfseingabe und den Zufallsbits von $\mathcal{A}$. Wir bezeichnen mit $\text{ADVR}_{\mathcal{A}}^{\mathcal{P}}(\overline{x},z,\overline{r})$ die Ausgabe des Angreifers $\mathcal{A}$ in der realen Welt für den Eingabevektor $\overline{x}$, die uniformen Zufallsbits $\overline{r}$ aller Parteien und die Hilfseingabe z. Sei $\text{REAL}_{\mathcal{A}}^{\mathcal{P}}(\overline{x},z,\overline{r})_i$ die Ausgabe der Partei i für die Ausführung von $\mathcal{P}$ auf den entsprechenden Werten,

$$\text{REAL}_{\mathcal{A}}^{\mathcal{P}}(\overline{x},z,\overline{r}) = \text{ADVR}_{\mathcal{A}}^{\mathcal{P}}(\overline{x},z,\overline{r}),\text{REAL}_{\mathcal{A}}^{\mathcal{P}}(\overline{x},z,\overline{r})_1,\ldots,\text{REAL}_{\mathcal{A}}^{\mathcal{P}}(\overline{x},z,\overline{r})_n$$

und $\text{REAL}_{\mathcal{A}}^{\mathcal{P}}(\overline{x},z)$ die Zufallsvariable für die Ausgaben, wenn $\overline{r}$ gleichverteilt gewählt wird.

In dem *Modell der idealen Welt* kommunizieren die Parteien mit einer Trusted Third Party, die die Funktionalität $\mathcal{F}$ berechnet. Wir werden jetzt beschreiben, wie die Berechnung mit einem aktiven Angreifer $\mathcal{S}$ in der idealen Welt aussieht. Auch hier kontrolliert der probabilistische Polynomialzeitalgorithmus $\mathcal{S}$ die Koalition K, kennt alle Daten von K sowie seine Hilfseingabe z und seine Zufallsbits. Zunächst kann $\mathcal{S}$ die Eingaben von K manipulieren. Seien $\overline{x}$ die originäre und $\overline{y}$ die manipulierte Eingabe, wobei die Einträge der ehrlichen Parteien in $\overline{x}$ und $\overline{y}$ identisch sind. Im nächsten Schritt senden alle Parteien ihre Eingaben an die Trusted Party. Diese berechnet $\mathcal{F}(\overline{y})$ und sendet $\mathcal{F}(\overline{y})_i$ an alle $i \in K$. Sendet nun der Angreifer $\mathcal{S}$ die Nachricht `ok` an die Trusted Party, dann schickt diese $\mathcal{F}(\overline{y})_i$ an alle ehrlichen Parteien i. Ansonsten bekommen die ehrlichen Parteien die Nachricht $\perp$ als Zeichen, dass der Angreifer das Protokoll abgebrochen hat. Alle Parteien sowie $\mathcal{S}$ berechnen ihre Ausgabe. Falls eine Partei $\perp$ erhalten hat, so gibt sie $\perp$ aus. Sei $\text{ADVR}_{\mathcal{S}}^{\mathcal{F}}(\overline{x},z,\overline{r})$ die Ausgabe von $\mathcal{S}$, wobei $\overline{r}$ der Vektor der Zufallsstrings von $\mathcal{S}$, der Parteien und der Trusted Third Party ist. Weiterhin sei $\text{IDEAL}_{\mathcal{S}}^{\mathcal{F}}(\overline{x},z,\overline{r})_i$ die Ausgabe der i-ten Partei und

$$\text{IDEAL}_{\mathcal{S}}^{\mathcal{F}}(\overline{x},z,\overline{r}) = \text{ADVR}_{\mathcal{S}}^{\mathcal{F}}(\overline{x},z,\overline{r}),\text{IDEAL}_{\mathcal{S}}^{\mathcal{F}}(\overline{x},z,\overline{r})_1,\ldots,\text{IDEAL}_{\mathcal{S}}^{\mathcal{F}}(\overline{x},z,\overline{r})_n.$$

Analog zu dem Modell der realen Welt bezeichnet $\text{IDEAL}_{\mathcal{S}}^{\mathcal{F}}(\bar{x},z)$ die Zufallsvariable der Ausgaben bei uniformer Wahl von $\bar{r}$.

Ein Protokoll $\mathcal{P}$, das die Funktionalität $\mathcal{F}$ berechnet, heißt *(kryptographisch) sicher*, falls es für jeden (nicht adaptiven) Angreifer $\mathcal{A}$ in der realen Welt einen (nicht adaptiven) Angreifer $\mathcal{S}$ in der idealen Welt gibt, so dass

$$\{\text{IDEAL}_{\mathcal{S}}^{\mathcal{F}}(\bar{x},z)\}_{\bar{x}\in\Sigma^n,z\in\{0,1\}^*} \stackrel{c}{\equiv} \{\text{REAL}_{\mathcal{A}}^{\mathcal{P}}(\bar{x},z)\}_{\bar{x}\in\Sigma^n,z\in\{0,1\}^*}.$$

Wenn $\mathcal{A}$ passiv ist, dann heißt $\mathcal{P}$ auch *(kryptographisch) privat*. Falls $\mathcal{A}$ höchstens t Parteien kontrolliert, nennen wir $\mathcal{P}$ *(kryptographisch) t-sicher / t-privat*.

2.13.5 Black-Box-Reduktionen

In den meisten Fällen des Entwurfs von kryptographischen Protokollen werden kryptographisch sichere Primitive als Grundlage benutzt, so auch in dieser Arbeit. Die Sicherheit der Protokolle basiert dann auf der Sicherheit der Primitive. Wir führen also ein kryptographisches Problem auf ein anderes zurück, das heißt, wir *reduzieren* das eine Problem auf das andere. Diese Reduktion nennen wir eine *Black-Box-Reduktion*, falls wir bei der Spezifizierung des Protokolls nur das Ein-/ Ausgabeverhalten des unterliegenden Primitivs nutzen, jedoch nicht seine Implementierung.

Die meisten Reduktionen zwischen kryptographischen Primitiven sind Black-Box-Reduktionen. Eingehend werden die Varianten von Black-Box-Reduktionen in [93] diskutiert. Beginnend mit der Arbeit von Impaglazzio und Rudich [60] wurde in einer Reihe von Arbeiten untersucht, ob Black-Box-Reduktionen zwischen bestimmten Primitiven existieren. Kürzlich zeigten Ishai et al. [61], dass mit einem homomorphen Verschlüsselungsschema oder erweiterten Falltür-Permutationen als Black-Box ein Oblivious-Transfer (siehe Abschnitt 3.7) implementiert werden kann, das sicher gegen beliebig viele aktive Angreifer ist. Somit kommen wir bei Umsetzung eines sicheren Oblivious-Transfers ohne aufwendige Zero-Knowledge-Beweise aus. Da wir nach Kilian [66] alle Protokolle auf Oblivious-Transfer Black-Box reduzieren können, gibt es für jede Protokollfunktionalität ein Protokoll, das sicher bei beliebig vielen aktiven Angreifern ist.

3

Grundlagen: Kryptographische Primitive

Wir werden in dieser Arbeit eine Reihe von kryptographischen Primitiven einsetzen. Diese werden wir in diesem Kapitel vorstellen. Einen Einstieg in die wichtigsten Primitive und Hinweise zur Literatur finden wir in den Büchern von Goldreich [41, 42], Schneier [99] und Menezes et al. [73]. Bei den meisten Definitionen beziehen wir uns auf Goldreich.

3.1 Einwegfunktionen

Einwegfunktionen sind ein zentrales Primitiv der Kryptographie. Viele Verfahren können mit Einwegfunktionen beziehungsweise mit Sammlungen von Einweg(-Falltür)-Permutationen realisiert werden, zum Beispiel Public-Key-Kryptographie (siehe Abschnitt 3.4) oder Pseudozufallszahlengeneratoren (siehe Abschnitt 3.3). Eine Einwegfunktion ist einfach zu berechnen, jedoch schwer zu invertieren. Wir werden Einwegfunktionen oder -permutationen in dieser Arbeit nicht direkt benutzen. Da sie aber als Bauelement für die weiteren Primitive wichtig sind, stellen wir die Definition einer Einwegfunktion vor.

Einweg-Falltür-Permutationen werden von Diffie und Hellman in ihrer grundlegenden Arbeit [31] eingeführt. Yao [111] führt schwache Einwegfunktionen ein und zeigt, dass die Existenz von schwachen und den hier präsentierten starken Einwegfunktionen äquivalent ist. Kandidaten für Einwegfunktionen sind die RSA-Funktion [95] und Rabins Quadratwurzelverfahren [91]. Für eine einfache Einführung in Einwegfunktionen verweisen wir auf [94] und für eine eingehende Betrachtung auf [41].

Eine Funktion $f\colon \{0,1\}^* \to \{0,1\}^*$ ist eine *Einwegfunktion*, wenn folgende Bedingungen gelten:

Berechenbarkeit:
Es gibt einen deterministischen Polynomialzeitalgorithmus A, der f berechnet, das heißt, für alle Eingaben x ist $A(x) = f(x)$.

schwere Invertierbarkeit:
Für jeden probabilistischen Polynomialzeitalgorithmus $\mathcal{A}$, jedes positive Polynom p

und alle genügend großen Werte n gilt:

$$\Pr[\mathcal{A}(f(U_n), 1^n) \in f^{-1}(f(U_n))] < \frac{1}{p(n)},$$

wobei die Zufallsvariable U_n uniform über alle Strings der Länge n verteilt ist und $f^{-1}(y) = \{x \mid f(x) = y\}$. Kein effizienter Algorithmus kann also ein Urbild für eine Ausgabe von f mit nicht vernachlässigbarer Erfolgswahrscheinlichkeit berechnen.

3.2 Einweg-Hash-Funktionen

Wie Einwegfunktionen sind auch Einweg-Hash-Funktionen ein wichtiger Grundbaustein in der Kryptographie, unter anderem für digitale Signaturen (siehe Abschnitt 3.5). *Hash-Funktionen* sind Funktionen, die einen langen String zu einem kürzeren, dem *Hashwert*, komprimieren. Für eine Einweg-Hash-Funktion ist es schwer, zwei Strings zu finden, die denselben Hashwert ergeben.

In Anlehnung an [90, 76, 81] definieren wir Einweg-Hash-Funktionen wie folgt:
Eine Funktion $h\colon \{0,1\}^* \to \{0,1\}^n$ nennen wir *Einweg-Hash-Funktion*, falls gilt:

Effizienz:
Es gibt einen Polynomialzeitalgorithmus, der h berechnet.

Kollisionsfreiheit I:
Für jeden probabilistischen Polynomialzeitalgorithmus ist die Wahrscheinlichkeit vernachlässigbar, auf Eingabe x einen Wert $y \neq x$ mit $h(x) = h(y)$ zu finden.

Kollisionsfreiheit II:
Für jeden probabilistischen Polynomialzeitalgorithmus ist die Wahrscheinlichkeit vernachlässigbar, zwei Eingaben $x \neq y$ mit $h(x) = h(y)$ zu finden. Hierbei müssen x und y polynomiell in der Länge der Eingabe beschränkt sein.

Die letzte Eigenschaft macht die Einweg-Hash-Funktion sicher gegen sogenannte Geburtstagsattacken [63].

3.3 Pseudozufallszahlengeneratoren

Pseudozufallszahlengeneratoren haben drei wichtige Eigenschaften: Erstens sind sie deterministisch. Somit ist die Ausgabe reproduzierbar. Zweitens und im Gegensatz zu einer Hash-Funktion produzieren sie Strings, die länger sind als ihre Eingabe. Drittens ist ihre Ausgabe *pseudozufällig*, das heißt, kein außenstehender (effizienter) Betrachter kann die Ausgabe eines Pseudozufallszahlengenerators von einem zufällig gewählten String unterscheiden.

Um Pseudozufallszahlengeneratoren zu konstruieren, besteht ein Ansatz darin, ausgiebig spezielle statistische Tests auf einen Kandidaten für einen Pseudozufallszahlengenerator anzuwenden. Erst nach Bestehen aller Test hält man den Generator für pseudozufällig. Allerdings ist dieser Einsatz in der Kryptographie riskant, da nicht bekannt ist, ob so ein Generator robust gegenüber allen Angriffen ist. Wir wählen daher einen anderen Ansatz.

Ein *Pseudozufallszahlengenerator* ist ein deterministischer Polynomialzeitalgorithmus G der folgende Eigenschaften erfüllt:

Verlängerung:
Es gibt eine Funktion $\ell\colon \mathbb{N} \to \mathbb{N}$, so dass $\ell(n) > n$ für alle $n \in \mathbb{N}$ und $|G(s)| = \ell(|s|)$ für alle $s \in \{0,1\}^*$. Wir nennen die Funktion ℓ die *Stretch-Funktion* und die Eingabe s den *Seed von G*.

Pseudozufälligkeit:
Für $i \in \mathbb{N}$ sei U_i die uniform verteilte Zufallsvariable über der Menge $\{0,1\}^i$ der Strings der Länge i. Dann gilt für alle probabilistischen Polynomialzeitalgorithmen $\mathcal{A}$, alle Polynome p und alle genügend großen Werte n, dass

$$|\Pr[\mathcal{A}(1^n, G(U_n)) = 1] - \Pr[\mathcal{A}(1^n, U_{\ell(n)}) = 1]| < \frac{1}{p(n)}.$$

Bei zufälliger Eingabe für G kann also kein effizienter Angreifer die Ausgabe von G von einem gleichlangen Zufallsstring unterscheiden.

Eine typische Anwendung von Pseudozufallszahlengeneratoren ist die Reduktion der Zufallskomplexität von Algorithmen. Sei G ein Pseudozufallszahlengenerator mit Stretch-Funktion $\ell\colon \mathbb{N} \to \mathbb{N}$. Sei A ein probabilistischer Polynomialzeitalgorithmus und bezeichne $\rho\colon \mathbb{N} \to \mathbb{N}$ seine Zufallskomplexität. Somit liefert A auf eine Eingabe der Länge n höchstens $\rho(n)$ viele Zufallsbits. Wir bezeichnen mit $A(x,r)$ die Ausgabe von A auf Eingabe x bei Benutzung der Zufallsbitsequenz $r \in \{0,1\}^{\rho(|x|)}$. Wir betrachten den folgenden probabilistischen Algorithmus A_G:

1. Auf Eingabe x berechne die kleinste natürliche Zahl k, so dass $\ell(k) \geq \rho(|x|)$.
2. Wähle den Seed $s \in \{0,1\}^k$ uniform.
3. Gebe $A(x, G'(s))$ aus, wobei $G'(s) = G(s)_{1,\ldots,\rho(|x|)}$ die ersten $\rho(|x|)$ Bits von $G(s)$ sind.

Benutzen wir A_G anstelle von A, so reduzieren wir die Zufallskomplexität von $\rho(|x|)$ auf $\ell^{-1}(\rho(|x|))$. Außerdem ist es für keinen probabilistischen Polynomialzeitalgorithmus effizient möglich, einen Unterschied zwischen der Ausgabe von A und der entsprechenden Ausgabe von A_G bei gleicher Eingabe x festzustellen [43, 44].

Die folgende Konstruktion zeigt, wie man die Stretch-Funktion eines Pseudozufallszahlengenerators verlängern kann [53]. Sei G ein Pseudozufallszahlengenerator mit Stretch-Funktion $\ell(n) = n+1$. Sei $G^{(1)}(s) = G(s)$ und für alle $i \geq 1$ definieren wir induktiv:

$$G^{(i+1)}(s) = G(G^{(i)}(s)_{\{1,\ldots,|s|\}}) \circ G^{(i)}(s)_{\{|s|+1,\ldots,|s|+i\}}.$$

Dann gilt für jedes Polynom q, dass

$$G^{q(n)}\colon \{0,1\}^n \to \{0,1\}^{n+q(n)}$$

ebenfalls ein Pseudozufallszahlengenerator mit Stretch-Funktion $n+q(n)$ ist.

Blum und Micali [11] sowie Yao [111] führen die Notation für kryptographisch sichere Pseudozufallszahlengeneratoren ein. Einen kryptographisch sicheren Pseudozufallszahlengenerator basierend auf der Schwierigkeit der Primfaktorzerlegung und der Erkennung quadratischer Reste stellen Blum, Blum und Shub vor [10]. Goldreich et al. [45] zeigen, dass Pseudozufallszahlengeneratoren benutzt werden können, um Sammlungen von pseudozufälligen Funktionen zu generieren. Mit jeder Einwegfunktion können wir einen Pseudozufallszahlengenerator konstruieren (Håstad et al. [53]). Håstad et al. beweisen zudem, dass Pseudozufallszahlengeneratoren genau dann existieren, wenn es Einwegfunktionen gibt.

3.4 Symmetrische und asymmetrische Verschlüsselung

Verschlüsselungsverfahren bilden den Ursprung der Kryptographie. Im Folgenden beschreiben wir das grundsätzliche Szenario der Verschlüsselung, definieren den Begriff Verschlüsselungsschema und seine Sicherheit. Außerdem geben wir einen kurzen Überblick über wichtige Verfahren und dazugehörige Beispiele.

3.4.1 Grundszenario und Grundbegriffe

Das Grundszenario der Verschlüsselung besteht aus zwei Parteien, dem *Sender* und dem *Empfänger*. Diese Parteien sind über einen unsicheren Kanal miteinander verbunden. Der Sender möchte eine Nachricht über den Kanal übertragen, so dass eine abhörende Partei (Eve) die Nachricht nicht lesen kann. Dazu verschlüsselt der Sender den *Plaintext* (die Nachricht) mit einem *Verschlüsselungsalgorithmus* zu dem *Ciphertext*. Letzterer wird dann über den Kanal übertragen und vom Empfänger mit einem *Entschlüsselungsalgorithmus* zu dem *Plaintext* entschlüsselt. Die Schlüssel zur Ver- und Entschlüsselung werden dabei von einem probabilistischen Algorithmus, dem *Schlüsselgenerator*, erzeugt.

Ein *Verschlüsselungsschema* ist ein Tripel (G,E,D) von probabilistischen Polynomialzeitalgorithmen, das folgende Bedingungen erfüllt:

1. Auf Eingabe 1^n berechnet der Schlüsselgenerator G ein Paar von Bitstrings, das heißt, $G(1^n) = (G(1^n)_1, G(1^n)_2) = (e,d)$, wobei die Zufallsvariablen e und d dem Ver- und Entschlüsselungsschlüssel entsprechen. Falls für alle Ausgaben (e,d) von G gilt, dass $e = d$, dann nennen wir das Schema *symmetrisch*, andernfalls nennen wir es *asymmetrisch*. Bei asymmetrischen Verfahren nennen wir e den *öffentlichen* Schlüssel, d ist der *private* Schlüssel.

2. Für alle Plaintexte $\kappa \in \{0,1\}^*$ gilt für den Verschlüsselungsalgorithmus E und den Entschlüsselungsalgorithmus D, dass

$$\Pr[D(d,E(e,\kappa)) = \kappa] = 1,$$

   wobei die Wahrscheinlichkeit über alle Zufallsbits der Algorithmen E und D genommen werden. Für $D(d,\cdot)$ und $E(e,\cdot)$ schreiben wir auch $D_d(\cdot)$ beziehungsweise $E_e(\cdot)$.

Den Wert n bezeichnen wir als den *Sicherheitsparameter* des Schemas. Diese Definition besagt lediglich, dass die Anwendung der Entschlüsselung mit dem Schlüssel d auf die Verschlüsselung des Plaintextes κ mit dem Schlüssel e wieder den Plaintext ergibt. Die Ver- und Entschlüsselung ist also konsistent. Allerdings sagt dieses nichts über die Sicherheit des Schemas aus, denn auch die Identität als Ver- und Entschlüsselungsfunktion erfüllt diese Eigenschaften.

3.4.2 Sicherheit von Verschlüsselungsalgorithmen

Es gibt mehrere Ansätze, um die Sicherheit von Verschlüsselungsverfahren zu definieren. Ein Ansatz ist die *semantische Sicherheit*, die besagt, dass ein passiver Angreifer aus dem Ciphertext kein Wissen über den Plaintext bekommt. Wir beschreiben hier die Sicherheit eines Verschlüsselungsverfahrens durch die *Unterscheidbarkeit von Verschlüsselungen*. Diese

Definition ist äquivalent zur semantischen Sicherheit, und ist verwandt mit den Definitionen der Sicherheit aus Abschnitt 2.13.

Ein Verschlüsselungsschema (G,E,D) hat *ununterscheidbare Verschlüsselungen*, falls für jede polynomiell große Schaltkreisfamilie $\{C_n\}$, jedes Polynom p, für alle genügend großen Werte n und alle Strings x und y, wobei $|x| = |y|$ höchstens polynomiell in n ist, die folgenden Forderungen erfüllt sind:

- Für ein symmetrisches Schema fordern wir:

$$|\Pr[C_n(E_{G(1^n)_1}(x)) = 1] - \Pr[C_n(E_{G(1^n)_1}(y)) = 1]| < \frac{1}{p(n)},$$

- Für ein asymmetrisches Schema fordern wir:

$$|\Pr[C_n(G(1^n)_1, E_{G(1^n)_1}(x)) = 1] - \Pr[C_n(G(1^n)_1, E_{G(1^n)_1}(y)) = 1]| < \frac{1}{p(n)},$$

  das heißt, ein Angreifer kennt bei asymmetrischen Verfahren den öffentlichen Schlüssel.

Falls ein Verschlüsselungsschema (G,E,D) ununterscheidbare Verschlüsselungen hat, sagen wir, (G,E,D) ist *sicher gegen passive Angreifer*, Verfahren, die die obigen Sicherheitseigenschaften erfüllen, sind ebenfalls sicher, wenn wir mehrere (polynomiell viele) Nachrichten mit demselben Schlüssel verschlüsseln.

Für stärkere Definitionen der Sicherheit bei gewählten Plaintext- und Ciphertextangriffen verweisen wir auf [41]. Im Allgemeinen genügen uns in dieser Arbeit sichere Verschlüsselungsverfahren gegenüber passiven Angreifern, da wir aufgrund des eingeschränkten Informationsflusses annehmen können, dass die Parteien die erweiterten Attacken nicht durchführen können.

Eine wichtige Anmerkung ist, dass die Verschlüsselungsfunktion eines asymmetrischen Schemas nicht deterministisch sein darf. Ansonsten können wir leicht die Verschlüsselungen unterscheiden, indem wir die Verschlüsselung mit dem bekannten öffentlichen Schlüssel $e = G(1^n)_1$ auf x und y anwenden und den Unterschied feststellen. Somit hat das Schema keine ununterscheidbaren Verschlüsselungen und ist auch nicht semantisch sicher.

3.4.3 Wichtige Verschlüsselungsverfahren

Symmetrische Verschlüsselungsverfahren gibt es bereits seit der Antike. Komplexitätstheoretische Untersuchungen wurden aber erst von Shannon [103] eingeführt. Er zeigt, dass ein Verschlüsselungssystem nur dann perfekt sicher ist, wenn die Entropie des Schlüssels größer ist als die Entropie des Plaintextes. Dies erfüllt das *One-time Pad*. Der Schlüssel ist hierbei ein echter Zufallsstring, der genauso lang ist wie der Plaintext. Die Ver- und Entschlüsselung ergibt sich durch die `XOR`-Verknüpfung der beiden Strings. Kryptographische Sicherheit erhalten wir, wenn wir als Schlüssel einen pseudozufälligen String benutzen. Heute werden meist standardisierte Verfahren benutzt wie (Triple-)DES [83] oder AES [84, 28]. DES gilt als nicht mehr sicher, da eine vollständige Suche des Schlüsselraums in der Praxis möglich ist.

Asymmetrische Verfahren sind ebenfalls als *Public-Key-Kryptosysteme* bekannt. Diffie und Hellman führen dieses Prinzip ein [31]. Erste Verfahren stellen Merkle und Hellman [74] sowie Rivest, Shamir und Adleman [95] (RSA) vor. RSA ist eines der bekanntesten

asymmetrischen Verfahren. Der Schlüsselgenerator von RSA wählt zufällig zwei große Primzahlen p und q und berechnet $n = pq$. Weiterhin wählt er zufällig einen Wert ε, so dass ε und $(p-1)(q-1)$ teilerfremd sind. Der Wert δ bezeichne das modulare Inverse zu ε modulo $(p-1)(q-1)$. Dann ist der *öffentliche Schlüssel* $e = (\varepsilon, n)$. Der *private Schlüssel* ist $d = (\delta, n)$. Zur Verschlüsselung wird der Plaintext m in Blöcke m_i zerlegt, so dass $|m_i| < \log(n)$. Der Ciphertextblock ergibt sich durch $c_i = {m_i}^e \bmod n$. Bei der Entschlüsselung ergibt sich der Plaintext durch ${m_i}^d \bmod n$.

Weitere Verfahren zur asymmetrischen Kryptographie basieren auf elliptischen Kurven über endlichen Körpern. Dieser Ansatz wurde von Miller [78] und Koblitz [67] eingeführt. Eine detaillierte Darstellung der Verwendung von elliptischen Kurven in der Kryptographie kann man in [109] finden.

Goldwasser und Micali führen mit ihrer Arbeit „Probabilistic Encryption“ [48] die semantische Sicherheit und die Ununterscheidbarkeit von Verschlüsselungen sowie grundlegende Definitionen und Techniken der modernen Kryptographie wie effiziente Unterscheidbarkeit, das Simulationsparadigma oder das Hybridargument ein. Da bei einem bekannten öffentlichen Schlüssel die Verschlüsselung von RSA deterministisch ist, ist das Verfahren nach den obigen Definitionen nicht sicher. Ein sicheres Verfahren muss probabilistisch sein. Oft werden daher die Plaintextblöcke durch (pseudo)zufällige Strings verlängert. Wir erhalten so eine randomisierte Verschlüsselung. Ob dieser Ansatz sicher ist, ist unbekannt. Ein anderer Ansatz [42], der unter einer kryptographischen Annahme sicher ist, kombiniert RSA mit einem kryptographisch sicheren One-time Pad. Zur besseren Darstellung vernachlässigen wir einige Details. Ein (pseudo)zufälliger String r_i wird anstatt des Plaintextblocks m_i mit RSA verschlüsselt. Der Ciphertextblock besteht dann aus dem verschlüsselten Zufallsstring $\hat{r}_i$ sowie aus $\hat{c}_i = r_i \oplus m_i$. Zur Entschlüsselung wird $\hat{r}_i$ entschlüsselt. Der Plaintext ergibt sich durch $m_i = \hat{c}_i \oplus r_i$.

Dieses führt uns zu den hybriden Verfahren. Da symmetrischen Verfahren im Allgemeinen effizienter als asymmetrische Verfahren sind, eignen diese sich besser zur eigentlichen Verschlüsselung. Durch ein asymmetrisches Verfahren können wir jedoch vor der eigentlichen Verschlüsselung den gemeinsamen Schlüssel austauschen.

3.5 Digitale Signaturen

Den Nachweis über die Echtheit eines Dokuments erbringen wir in der analogen Welt durch unsere Unterschrift, da es für andere Personen schwer sein sollte, diese zu fälschen. Eine eingescannte und damit digitalisierte Unterschrift können wir nicht zum gleichen Zweck verwenden. Eine eingescannte Unterschrift kann beliebig kopiert werden. Zudem kann sie auch von anderen leicht benutzt werden. Das Problem, die Echtheit oder Authentizität eines Dokumentes durch eine Signatur zu bestätigen und zu validieren, war neben der Verschlüsselung eines der ersten Probleme, dass in der modernen Kryptographie betrachtet wurde.

Die Notation eines Signaturschemas wird von Diffie und Hellman [31] eingeführt. Die ersten Vorschläge zu *digitale Signaturen* gehen mit den ersten Vorschläge zur Public-Key-Kryptographie einher [95, 91]. Goldwasser, Micali und Yao [49] befassen sich als erste ausführlich mit der Sicherheit von digitalen Signaturen. In [50] gehen Goldwasser, Micali und Rivest umfassend auf die Sicherheit von Signaturen ein und präsentieren ein Verfahren, das sicher gegen gewählte Nachrichten-Attacken ist. Diese starke Definition der Sicherheit stellen wir im Folgenden vor.

3.5.1 Sicherheit von Digitalen Signaturen

Ein *Signaturschema* ist ein Tripel (G,S,V) von probabilistischen Polynomialzeitalgorithmen, das folgende Bedingungen erfüllt:

1. Auf Eingabe 1^n berechnet der Schlüsselgenerator G ein Paar von Bitstrings, das heißt, $G(1^n) = (G(1^n)_1, G(1^n)_2) = (s,v)$, wobei die Zufallsvariablen s und v dem *Signier-* und dem *Verifizierungsschlüssel* entsprechen.

2. Für alle Strings $\kappa \in \{0,1\}^*$ gilt für den *Signaturalgorithmus S* und den *Verifikationsalgorithmus V*, dass

$$\Pr[V(v,\kappa,S(s,\kappa)) = 1] = 1,$$

   wobei die Wahrscheinlichkeit über alle Zufallsbits der Algorithmen S und V genommen werden. Für $V(v,\cdot,\cdot)$ und $S(s,\cdot)$ schreiben wir auch $V_v(\cdot,\cdot)$ beziehungsweise $S_s(\cdot)$.

Den Wert n bezeichnen wir als den *Sicherheitsparameter* des Schemas.

Wir beschreiben hier die strenge Sicherheit für Signaturschemata, das heißt Sicherheit gegen gewählte Nachrichten-Attacken. Bei einer *gewählten Nachrichten-Attacke* wählt ein Angreifer adaptiv (polynomiell viele) Nachrichten und lässt sich von dem Signierer die entsprechenden Signaturen bezüglich desselben von G generierten Schlüsselpaares (s,v) erstellen. Wir sagen, dass die Attacke *erfolgreich* ist, falls der Angreifer eine gültige Signatur eines Strings ausgeben kann, zu dem er keine Signatur von dem Signierer erhalten hat. Das heißt, der Angriff ist erfolgreich, falls der Angreifer nach Erhalt des Verifizierungsschlüssels v und Abfrage der Signaturen $S_s(m_1),\ldots,S_s(m_t)$ ein Paar (α,β) ausgibt, so dass $V_v(\alpha,\beta) = 1$ und $\alpha \notin \{m_1,\ldots,m_t\}$. Das Signaturschema heißt *(fälschungs)sicher*, falls jeder probabilistische Polynomialzeitalgorithmus (Angreifer), der eine gewählte Nachrichten-Attacke durchführt, nur mit vernachlässigbarer Wahrscheinlichkeit erfolgreich ist.

3.5.2 Digitale Signaturen durch Public-Key-Kryptosysteme

Wir können Signaturschemata auf der Grundlage von symmetrischen und asymmetrischen Verschlüsselungsverfahren realisieren. Besonders interessant sind dabei die asymmetrischen Verfahren, da wir bei ihnen keine Trusted Third Party benötigen.

Wir nehmen an, dass Alice ein Dokument M signieren und Bob die Signatur verifizieren möchte. Sie haben sich auf ein asymmetrisches Verschlüsselungsschema (G,E,D) verständigt. Dabei bezeichne d den privaten und e den öffentlichen Schlüssel von Alice. Als Signaturschema verwenden die beiden das Schema $(G' = (G(\cdot)_2, G(\cdot)_1), S = D, V = E)$:

1. Der Schlüsselgenerator G' verwendet G und berechnet

$$G'(1^n) = (s,v) = (d,e) = (G(1^n)_2, G(1^n)_1).$$

2. (Signatur:) Alice generiert die Signatur

$$\sigma = S_s(M) = D_d(M).$$

3. (Verifizierung:) Bob überprüft die Signatur. Dazu prüft $V_v(M,\sigma)$, ob

$$E_e(\sigma) = M.$$

3.5.3 Hash-and-Sign

Bisher haben wir nur Wert darauf gelegt, dass die Signatur fälschungssicher ist und verifiziert werden kann. Jedoch lassen diese Eigenschaften keinen Schluss darüber zu, ob die Signatur privat ist, das heißt, ob sie kein Wissen über die signierte Nachricht verrät. Dies erreichen wir, wenn wir eine Einweg-Hash-Funktion mit einem sicheren Signaturschema kombinieren [30, 29]. Alice berechnet dazu den Hash-Wert $h(M)$ der Nachricht M. Dann generiert sie die Signatur $\sigma = S_s(h(M))$. Bob kann die Signatur verifizieren, wenn er Zugang zu der Nachricht M hat. Dazu berechnet er ebenfalls $h(M)$ und verifiziert σ. Die Signatur σ ist fälschungssicher. Sie ist zudem privat, da σ kein Wissen über M gibt, wenn Bob keinen Zugang zu M hat. Ein weiterer Vorteil der Kombination ist, dass die Signatur beschleunigt wird, weil wir mit einer Einweg-Hash-Funktion die Nachricht zu einem kürzeren String komprimieren und anschließend diesen kurzen String signieren.

3.5.4 Weitere Anwendungen

Message-Authentication-Schemes sind eng verwandt mit digitalen Signaturen. Sie sollen die Integrität einer Nachricht sicherstellen, beispielsweise wenn Alice an Bob eine Nachricht über einen von Mallory gestörten Kanal sendet. Alice sendet zusätzlich eine Signatur (Message-Authentication-Code, MAC) an Bob. Im Unterschied zu digitalen Signaturen nehmen wir nun an, dass Mallory den Verifizierungsschlüssel nicht kennt, Bob jedoch schon. Somit kann Bob die Nachricht verifizieren. Mallory kann hingegen die Nachricht nicht fälschen.

Mit digitalen Signaturen und einem Public-Key-Verschlüsselungsverfahren können wir eine *Public-Key-Infrastruktur* aufbauen. Dabei signiert eine Zertifizierungsstelle die öffentlichen Schlüssel von teilnehmenden Parteien, welche wiederum Nachrichten verschlüsseln und signieren können. Durch die Verwendung signierter öffentlicher Schlüssel und unter der realistischen Annahme, dass die Parteien ihren privaten Schlüssel geheim halten, können die Parteien Nachrichten austauschen und authentifizieren. Eine wichtige Erweiterung ist der Aufbau einer hierarchischen Struktur der Zertifizierungsstellen. Die Grundlage dieser Struktur bildet der sogenannte Signaturbaum, der auf Merkle zurückgeht [75]. Somit reicht es aus, dass wir der Wurzelzertifizierungsstelle vertrauen, auch wenn die Zertifizierungsstellen dezentral Schlüssel unterschreiben. In dieser Arbeit gehen wir davon aus, dass alle Parteien Zugang zu einer solchen vertrauenswürdigen Infrastruktur haben.

3.6 Blinde Signaturen

Nutzen wir digitale Signaturen, so wissen wir, welche Nachricht wir unterschreiben. Blinde Signaturen ermöglichen es, dass wir eine uns unbekannte Nachricht signieren können. Sie sind zentral für viele E-Cash- und E-Voting-Schemata. Durch sie kann ein Geldschein oder Stimmzettel als gültig erklärt werden, ohne dass die Bank oder der Wahlausschuss auf den Kunden oder den Wähler zurückschließen können. Chaum [20] stellt als erster ein Blinde-Signatur-Schema vor. Es basiert auf RSA. Juels et al. [64] und Pointcheval et al. [89] definieren die Sicherheit für blinde Signaturen. Bellare et al. zeigen [6], dass Chaums Schema sicher ist. Die exakte Definition der Sicherheit ist sehr technisch. Daher beschreiben wir hier sichere Blinde-Signatur-Schemata lediglich. Für die genaue Definition verweisen wir beispielsweise auf [37].

3.6.1 Definition

Ein *Blinde-Signatur-Schema* (G,V,BS) besteht aus einem probabilistischen und effizienten *Schlüsselgenerator* $G\colon \{0,1\}^* \to \{0,1\}^*$, einem effizienten *Verifikationsalgorithmus* $V\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ und einem Protokoll *BS* für zwei Parteien, die wir den *Signierer* $\mathcal{S}$ und den *Benutzer* $\mathcal{B}$ nennen. Zudem gilt Folgendes:

1. Der Schlüsselgenerator generiert den Signierungsschlüssel s und den Verifizierungsschlüssel v, so dass $G(1^n) = (s,v)$. Dabei nennen wir n den *Sicherheitsparameter* des Schemas.
2. Für jede Nachricht $M \in \{0,1\}^*$ gilt:
   (a) Nach Durchführung von *BS* mit $\mathcal{S}$ auf Eingabe s und $\mathcal{B}$ auf Eingabe (v,M) erhält $\mathcal{S}$ als Ausgabe den leeren String λ und $\mathcal{B}$ die Signatur σ von M.
   (b) Die Signatur σ von M ist *valide*, das heißt,
   $$\Pr[V_v(M,\sigma) = 1] = 1.$$

3.6.2 Sicherheit

Ein Blinde-Signatur-Schema (G,V,BS) heißt *sicher*, wenn es fälschungssicher und blind ist:

Fälschungssicherheit:
Nach k-maliger Durchführung des Protokolls *BS* zwischen $\mathcal{S}$ und einem bösartigen Benutzer $\mathcal{B}^*$ ist die Wahrscheinlichkeit vernachlässigbar, dass $\mathcal{B}^*$ $k+1$ valide Signaturen $\sigma_1,\ldots,\sigma_{k+1}$ für $k+1$ unterschiedliche Nachrichten $M_1,\ldots,M_{k+1}$ ausgeben kann.

Blindheit:
Angenommen, dass ein bösartiger Signierer $\mathcal{S}^*$ den Verifizierungsschlüssel v, sowie zwei Nachrichten M_0 und M_1 wählen darf. Nach zweimaliger Ausführung von *BS* zwischen $\mathcal{B}$ (mit Eingabe (v,M_0) beziehungsweise (v,M_1)) und $\mathcal{S}^*$ erhält $\mathcal{B}$ die Signaturen σ_1 beziehungsweise σ_2. Sei b ein für $\mathcal{S}^*$ unbekanntes und uniform gewähltes Bit. Dann ist die Wahrscheinlichkeit q, dass $\mathcal{S}^*$ bei Kenntnis des Tupels $(\sigma_b,\sigma_{1\oplus b})$ den Wert b berechnen kann, vernachlässigbar nah an $\frac{1}{2}$. Für alle Polynome p und alle genügend großen Werte des Sicherheitsparameters n gilt somit:
$$q < \frac{1}{2} + \frac{1}{p(n)}.$$

3.6.3 Nicht-Existenz von verdeckten Kanälen

Aus der Blindheit können wir direkt folgern, dass ein sicheres Blinde-Signatur-Schema keinen *verdeckten Kanal* enthält. Das bedeutet, der Signierer kann in einer blinden Signatur nur mit vernachlässigbarer Wahrscheinlichkeit eine Nachricht verstecken, die er bei einem Zugriff auf die Signatur auch zurückberechnen kann. Ansonsten könnte er je ein Bit in die Signatur von σ_0 und σ_1 mit nicht vernachlässigbarer Wahrscheinlichkeit einschleusen und somit b bestimmen. Stellt der Signierer mehreren Benutzern blinde Signaturen aus, dann kann er aufgrund der Blindheit des Schemas nicht zwischen den Signaturen der verschiedenen Benutzern unterscheiden. In der vorliegenden Arbeit benutzen wir blind unterschriebene Tokens als Beweis dafür, dass eine bestimmte Aktion einmal ausgeführt werden darf, wobei die Erlaubnis zur Durchführung anonym erteilt wurde. Versucht ein Angreifer die Aktion mehrmals durchzuführen, so können wir ihn leicht an demselben Token erkennen.

3.7 Oblivious-Transfer

Wie schon im Abschnitt 2.13.5 beschrieben, können wir nach Kilian [66] und Ishai et al. [61] jedes Mehr-Parteien-Protokoll durch Oblivious-Transfer sicher implementieren. Damit nimmt Oblivious-Transfer (OT) eine zentrale Rolle in der Kryptographie ein. Das erste Oblivious-Transfer-Protokoll stammt von Rabin [92]. Even et al. [34] stellen das 1-aus-2-Bit-Oblivious-Transfer-Protokoll vor, das nach Crépeau [27] äquivalent zu Rabins Verfahren ist. Erweiterungen sind 1-aus-n-Bit-Oblivious-Transfer [15] und verschiedene String-Oblivious-Transfers, zum Beispiel [27]. Kolesnikov [68] stellt dar, dass sich Varianten von sicherem 1-aus-n-String-Oblivious-Transfer sehr effizient bei Benutzung gesicherter Hardware-Token realisieren lassen, die zustandslos und damit günstig sind.

Unter *sicherem 1-aus-n-String-Oblivious-Transfer* verstehen wir ein Protokoll, das folgende Funktionalität sicher berechnet:

Eingabe:
Der *Sender* hat n Strings $s_0, \ldots, s_{n-1} \in \{0,1\}^*$. Der *Empfänger* hat einen Wert $b \in \{0, \ldots, n-1\}$.

Ausgabe:
Der Sender erhält als Ausgabe den leeren String λ. Der Empfänger erhält s_b.

Nach Ausführung des Protokolls hat der Sender kein Wissen über den Wert b und der Empfänger erlangt kein Wissen über die Eingabe des Senders außer dem Ergebnis s_b.

In dieser Arbeit nutzen wir bei der Verwendung eines sicheren 1-aus-n-(String)-Oblivious-Transfer-Protokolls die funktionale Schreibweise

$$\mathrm{OT}_b(s_0, \ldots, s_{n-1}) = s_b.$$

3.8 Secret-Sharing

In diesem Abschnitt gehen wir darauf ein, wie man ein *Geheimnis* aufteilen kann. Dabei hält der *Sender* das Geheimnis und verteilt es auf mehrere *Empfänger*, indem er an jeden Empfänger ein sogenanntes *Share* sendet. Der Sender bestimmt zudem Teilgruppen, die aus ihren Shares das Geheimnis berechnen dürfen. Vor allen anderen Teilgruppen bleibt das Geheimnis sicher. Sie können kein Wissen über das Geheimnis aus ihren Shares ableiten.

Ein Protokoll, das diese Funktionalität berechnet nennen wir ein *Secret-Sharing-Schema*. Die Teilgruppen werden dabei durch sogenannte *Zugriffsstrukturen* bestimmt. Unabhängig voneinander entwickeln Shamir [101] und Blakley [7] die ersten Secret-Sharing-Schemata. Sie definieren dabei auch die gängigste Form von Zugriffstrukturen: *Schwellwerte*. Der Schwellwert legt fest, wie viele der Empfänger mindestens zusammenkommen müssen, um das Geheimnis zu rekonstruieren.

3.8.1 Secret-Sharing-Schemata mit Schwellwert

Ein *perfektes (k,n)-Secret-Sharing-Schema für das Universum U* ist ein Protokoll für einen Sender und n Empfänger, das folgende Funktionalität effizient berechnet:

1. Der Sender hat als Eingabe ein Geheimnis $s \in U$. Die Empfänger haben keine Eingabe.

2. Als Ausgabe erhält der Sender den leeren String und der Empfänger E_i das Share s_i.

Zusätzlich muss das Schema folgende Eigenschaften erfüllen:

Rekonstruierbarkeit:
Für eine Menge K von Parteien sei $S_K = \{s_i \mid i \in K\}$ die Menge aller Shares der Parteien in K. Es gibt einen probabilistischen Polynomialzeitalgorithmus R, so dass für alle $s \in U$ und alle Mengen $K \subseteq \{1,\ldots,n\}$ der Größe $|K| = k$ gilt:

$$R(S_K) = s.$$

Der Wert k wird *Schwellwert* genannt. Wenn wir k Shares von s kennen, dann können wir s rekonstruieren.

Privatheit:
Seien S eine Zufallsvariable, die das Geheimnis beschreibt und Werte aus U annimmt, sowie $S_1,\ldots,S_n$ die Zufallsvariablen der Shares von S. Dann gilt für alle Mengen $K \subseteq \{1,\ldots,n\}$ der Größe $|K| < k$, dass

$$I(S;\mathcal{S}_K) = 0,$$

wobei $\mathcal{S}_K = \{S_i \mid i \in K\}$. Die Verteilung der Shares in $\mathcal{S}_K$ ist somit unabhängig von der Wahl des Geheimnisses. Solange wir weniger als k Shares kennen, haben wir keine Information über den Wert von S. Das Geheimnis bleibt also informationstheoretisch privat.

Die Definition für ein kryptographisches Secret-Sharing-Schema erhalten wir durch entsprechende Anpassung der Privatheitsbedingung an die kryptographische Privatheit.

3.8.2 Shamirs Secret-Sharing-Schema

Die Grundlage von Shamirs (k,n)-Secret-Sharing-Schema [101] bildet die Interpolation eines Polynoms über einen endlichen Körper. Wir verwenden zur Darstellung den Körper $\mathbb{Z}_p$ für eine Primzahl p. Um das Geheimnis $s \in \mathbb{Z}_p$ aufzuteilen, wählt der Sender randomisiert Koeffizienten $a_i \in \mathbb{Z}_p$ für $i \in \{1,\ldots,t-1\}$, setzt $a_0 = s$ und erhält so das Polynom

$$p(x) = \sum_{i=0}^{k-1} a_i x^i.$$

Der Sender sendet das *Shamir Share* $s_i = p(i)$ an den i-ten der n Empfänger.

Sei $K \subseteq \{1,\ldots,n\}$ mit $|K| = k$, dann können die Empfänger aus K das Polynom p und damit auch das Geheimnis $s = p(0)$ durch die *Lagrange-Interpolation* zurückberechnen:

$$p(x) = \sum_{i \in K} \ell_i(x) \cdot s_i,$$

wobei

$$\ell_i(x) = \prod_{j \in K, j \neq i} (x-j) \cdot (i-j)^{-1}$$

und a^{-1} das multiplikative Inverse von a in dem endlichen Körper ist. Für Mengen K mit $|K| > k$ wählen wir zur Rekonstruktion eine k-elementige Teilmenge aus K aus. Für $|K| < k$ ist es jedoch unmöglich, s zu bestimmen.

3.8.3 XOR-Secret-Sharing-Schema

Ein sehr einfaches, perfektes (n,n)-Secret-Sharing-Schema erhalten wir durch Verwendung der bitweisen XOR-Funktion $\oplus$. Sei $s \in \{0,1\}^m$ das Geheimnis des Senders. Als Shares $s_1, \ldots, s_{n-1}$ wählt der Sender unabhängig und uniform Strings der Länge m. Das Share s_n berechnen wir durch

$$s_n = s \oplus s_1 \oplus \ldots \oplus s_{n-1}.$$

Damit gilt für die Rekonstruktion:

$$s = s_1 \oplus \ldots \oplus s_n.$$

3.8.4 Verifiable-Secret-Sharing-Schemata

Beteiligt sich Mallory am Secret-Sharing, so kann er als Sender oder als Empfänger das Geheimnis oder dessen Rekonstruktion beeinflussen. Um aktiven Angreifern beim Secret-Sharing zu begegnen, beschreiben Chor et al. [25] das *Verifiable Secret-Sharing*. Bekannte Protokolle entwickeln Feldman [35] und Pedersen [86]. Verifiable-Secret-Sharing-Schemata stellen sicher, dass alle Shares konsistent sind, auch wenn einige der Parteien bösartig sind [38]. Dabei legt sich der Sender durch ein sogenanntes *Commitment* auf einen bestimmten Wert fest. Ist der Sender bösartig, so muss dieser Wert nicht dem Geheimnis entsprechen. Das Schema gewährleistet, dass auch der von ehrlichen Parteien rekonstruierte Wert mit dem festgelegten Wert übereinstimmt oder der Angriff erkannt wird. Publicly-Verifiable-Secret-Sharing-Schemata [104, 100] erweitern die Überprüfbarkeit der Konsistenz der Shares auf Parteien, die weder Sender noch Empfänger sind.

3.9 Cut-and-Choose-Technik

Wir wollen im Kapitel 4 Secret-Sharing dazu einsetzen, einem Empfänger Shares von vielen Sendern zu übermitteln. Dabei soll der Empfänger die Shares eines Senders nicht anhand der Werte dieser Shares in Zusammenhang bringen können. Folglich müssen wir eine Überprüfung der Konsistenz eines Shares unabhängig von anderen Shares desselben Geheimnisses durchführen. Zudem möchten wir einen bösartigen Sender möglichst früh erkennen und daher die Verifikation der Shares vor der Rekonstruktion durchführen. Verifiable-Secret-Sharing-Schemata können wir nicht für diese Aufgabe einsetzen, da Broadcast-Nachrichten versendet werden, Verifikationsinformation von mehreren Shares zusammengebracht werden müssen oder die Verifikation der Konsistenz erst bei der Rekonstruktion stattfindet. Übernimmt Mallory beim Verifiable-Secret-Sharing die Rolle des Senders, so kann er das Geheimnis ändern, ohne dass die Empfänger dieses feststellen können.

Durch ein dem Secret-Sharing vorgeschaltetes *Cut-and-Choose* können wir mit großer Sicherheit den Sender auf eine Eingabe und entsprechende Shares festlegen. Diese Technik setzt Chaum et al. [23] ein, um einen Signierer bei blinden Signaturen vor einem bösartigen Benutzer zu schützen. Dazu bitten wir den Benutzer, sich auf m Eingaben festzulegen und uns blinde (verschlüsselte) Werte vorzulegen. Alle Shares bis auf einen (beziehungsweise einen Teil der blinden Werte) lassen wir uns öffnen, das heißt entschlüsseln. Sind diese Werte korrekt, so unterschreiben wir den verschlossenen Wert blind. Mit einer Wahrscheinlichkeit von mindestens $1 - \frac{1}{m}$ sind wir sicher, dass der unterschriebene Wert nicht gefälscht ist. Später vertrauen wir anhand der Unterschrift dem vorgelegten Wert. Wir können auf diese Weise

einen Sender auf konsistente Shamir Shares festlegen, indem wir ihn mehrere Polynome erzeugen lassen und durch Cut-and-Choose die entsprechenden Shares testen. Gegebenenfalls können wir überprüfen, ob das Geheimnis der Shares bestimmte Eigenschaften hat, zum Beispiel, ob es die korrekte Identität des Senders enthält. Der Sender verfügt anschließend über konsistente Shares, die blind unterschrieben sind. Erhalten wir ein Share, können wir die Signatur unabhängig von anderen Shares auf Konsistenz überprüfen und uns sehr sicher sein, dass das Geheimnis des Shares zulässig ist.

3.10 Bit-Commitment

Bit-Commitment-Schemata erlauben es, dass sich Alice gegenüber Bob auf ein Bit festlegt, dieses Bit Bob aber nicht verrät. Erst nach der Festlegung veröffentlicht Alice den Wert des Bits. Bob kann dann verifizieren, dass dieser Wert dem festgelegten entspricht. Bit-Commitment-Schemata werden beispielsweise für Zero-Knowledge-Protokolle [46, 14] oder Mehr-Parteien-Protokolle [47, 22] verwendet.

3.10.1 Definition

Wir halten uns bei der Definition von Bit-Commitment-Schemata an Naor [82], dessen Protokoll wir im nächsten Abschnitt kurz vorstellen werden. Ein *Bit-Commitment-Schema* ist ein Protokoll für zwei Parteien Alice und Bob und besteht aus zwei Phasen.

Commitment-Phase:
Alice hat ein Bit b, auf welches sie sich gegenüber Bob festlegen möchte. Alice und Bob tauschen in dieser Phase Nachrichten aus. Am Ende der Phase hat Bob Information, die das Bit b repräsentiert.

Öffnungsphase:
Am Ende dieser Phase kennt Bob den Wert von b.

Dabei erfüllt das Protokoll folgende Sicherheitseigenschaften:

Für alle probabilistischen Polynomialzeitalgorithmen für Bob, für alle Polynome p und alle genügend großen Werte des Sicherheitsparameters n gilt, dass

1. die Wahrscheinlichkeit, dass Bob den Wert von b richtig bestimmen kann, nicht größer als $\frac{1}{2} + \frac{1}{p(n)}$ ist. Sie ist also vernachlässigbar.

2. Alice kann nur einen möglichen Wert veröffentlichen. Wenn sie nicht den richtigen Wert veröffentlicht, so wird der Betrug mit einer Wahrscheinlichkeit von mindestens $1 - \frac{1}{p(n)}$ erkannt. Die Erfolgswahrscheinlichkeit eines Betrugs ist folglich vernachlässigbar.

3.10.2 Naors Bit-Commitment-Schema

Die Idee bei diesem Schema ist es, den Wert des Bits b mithilfe eines sicheren Pseudozufallszahlengenerators zu verschlüsseln und sich damit auf b festzulegen. Für Pseudozufallszahlengeneratoren ist es schwer, Kollisionen zu finden, das heißt, zwei Seeds zu finden, deren Pseudozufallszahlen an bestimmten Stellen übereinstimmen. Diese Eigenschaft nutzt Naor aus, um Bob vor einem Betrug von Alice zu schützen.

Sei G ein sicherer Pseudozufallszahlengenerator. Der Sicherheitsparameter n bezeichnet die Seedlänge. Mit $G_\ell(s)$ bezeichnen wir hier die ersten ℓ Bits der Ausgabe von G auf Eingabe des Seeds $s \in \{0,1\}^n$. Der Wert $B_i(s)$ sei das i-te Bit der Ausgabe von G auf s.

1. Commitment-Phase:
   (a) Bob wählt einen zufälligen String $R = r_1, \ldots, r_{3n}$ und sendet ihn an Alice.
   (b) Alice wählt einen Seed $s \in \{0,1\}^n$ und sendet Bob den String $D = d_1, \ldots, d_{3n}$ mit

$$d_i = \begin{cases} B_i(s) & \text{falls } r_i = 0, \\ B_i(s) \oplus b & \text{falls } r_i = 1. \end{cases}$$

2. Öffnungsphase: Alice sendet den Seed s an Bob. Bob verifiziert für alle $1 \leq i \leq 3n$, dass $d_i = B_i(s)$ für $r_i = 0$ und $d_i = B_i(s) \oplus b$ für $r_i = 1$ gilt.

Wählt Bob mindestens für ein Bit in R den Wert 1, so erhält er in der Öffnungsphase den Wert b. Da R unabhängig von b gewählt ist und D durch einen Pseudozufallsstring verschlüsselt ist, hat Bob am Ende der Commitment-Phase kein Wissen über B. Die Wahrscheinlichkeit, dass es zu R zwei Seeds s_1 und s_2 gibt, für die $G_{3n}(s_1)$ und $G_{3n}(s_2)$ auf allen Positionen i mit $r_i = 0$ übereinstimmen und auf allen Positionen i mit $r_i = 1$ nicht übereinstimmen, ist höchstens 2^{-n}. Somit kann Alice Bob nur mit vernachlässigbarer Wahrscheinlichkeit während der Öffnungsphase von einem falschen Wert b' überzeugen.

3.11 Quellen echten Zufalls

Die Sicherheit der meisten Primitive in diesem Kapitel basiert darauf, dass wir Zugriff auf uniform verteilte Zufallszahlen haben und dass dieser in ausreichendem Maß vorhanden ist. Wie in Abschnitt 3.3 dargestellt, können wir Pseudozufallszahlengeneratoren dazu verwenden, den Bedarf an uniformen Zufallszahlen in einem sicheren System gering zu halten. Für den Seed eines Pseudozufallszahlengenerators benötigen wir weiterhin uniforme Zufallsbits. In der Praxis ist allerdings eine Quelle von uniformen Zufallsbits schwer zu finden. Da die Entropie einer Quelle von uniformen Zufallsbits maximal ist, versucht man zur Gewinnung von Zufall in Computern Quellen hoher Entropie zu verwenden. Dazu gehören zum Beispiel die Messung der exakten Zeitpunkte der Tastatureingabe, der Bewegung der Maus, des Rauschens an einem Mikrofon oder der Fluktuation von Zugriffszeiten von Festplatten [36]. Ob die Bits, die aus diesen Daten extrahiert werden, tatsächlich echt zufällig oder zumindest hinreichend zufällig sind, ist schwer feststellbar. Geübte Benutzer können einen solch regelmäßigen Tastaturanschlag haben, dass die Rhythmusunterschiede unterhalb der Messauflösung des Eingabegerätes liegen. Damit treten fast keine zufälligen Abweichungen auf. Hat ein Angreifer Kenntnis von solchen Eigenschaften der Verteilung der generierten Bits, kann dies den Angriff sehr erleichtert. Kann der Angreifer zudem physikalisch auf einen Rechner zugreifen, so besteht zudem die Möglichkeit, dass er die Generierung der Zufallsbits und damit ihre Verteilung beeinflusst. Moderne Rechner haben teilweise Krypto-Chips mit eingebauten Zufallszahlengeneratoren, die solchen Angriffen widerstehen sollen.

Viele physikalischen Prozesse, insbesondere Quantenprozesse, verhalten sich echt zufällig. Es gibt Hoffnung, dass aus Messungen solcher Prozesse echte Zufallszahlen gewonnen

werden können. Kürzlich konnten dies beispielsweise Pironio et al. [88] für bestimmte Quanteneigenschaften („nonlocal correlations of entangled quantum particles“) experimentell nachweisen.

4

Schwellwert-Schema zur privaten Vorratsdatenspeicherung

Bevor wir auf die Vorratsdatenspeicherung mit Schwellwert-Bedingung eingehen, werden wir zunächst im Abschnitt 4.1 die Sicherheitsanforderungen an die private Vorratsdatenspeicherung im Allgemeinen betrachten. In den Abschnitten 4.2 bis 4.7 wird das Schwellwert-Schema zur privaten Vorratsdatenspeicherung vorgestellt. Der Nachweis der Sicherheitseigenschaften des Schemas erfolgt in Abschnitt 4.8. Anschließend wird in Abschnitt 4.9 die Effizienz des Schemas anhand von Laufzeitergebnissen untersucht. Zum Ende des Kapitels werden in Abschnitt 4.10 mögliche Erweiterungen des Schemas diskutiert.

4.1 Anforderungen an die private Vorratsdatenspeicherung

In dieser Arbeit beschäftigen wir uns mit der Privatheit von gespeicherten Daten. Wir betrachten nicht die Korrektheit und Sicherheit des Aufzeichnens. Wir gehen davon aus, dass die Datenintegrität der angefallenen Daten, das heißt, der Schutz vor falsch aufgezeichneten Daten sowie deren Plausibilitätsprüfung, Verifikation, Fehlererkennung und Fehlerkorrektur sichergestellt ist.

Im Folgenden fassen wir die Sicherheits- und Privatheitsanforderungen an ein Schema zur Vorratsdatenspeicherung zusammen:

Minimierung des Datenaufkommens
Das Schema legt fest, welche Art von Daten aufzuzeichnen und zu speichern sind. Dabei ist darauf zu achten, dass so wenig Daten wie möglich gespeichert werden.

Schutz und Anonymität
Die Daten werden verschlüsselt gespeichert. Es muss darauf geachtet werden, dass die verschlüsselten Daten so wenig Wissen wie möglich über die Personen enthalten, auf die sich die Daten beziehen. Personen müssen zudem vor ungerechtfertigtem Verdacht geschützt werden.

Kontrolle des Zugriffs
Die Regeln für den Zugriff auf die gespeicherten Daten sind vor der Aufzeichnung und Speicherung der Daten fest- und offenzulegen. Nur bei gerechtfertigtem Verdacht, beziehungsweise, wenn eine regelkonforme Erlaubnis vorliegt, können die Daten innerhalb der Vorhaltezeit geöffnet werden.

Trennung
Die räumliche Trennung der Speicherung von der Aufzeichnung und damit auch von der Verschlüsselung trägt zur Verhinderung des missbräuchlichen Zugriffs auf die Daten bei.

Rückwärtssicherheit/Löschung
Die Nützlichkeit alter Daten, das heißt Daten außerhalb der Vorhaltezeit, ist zu beschränken. Ist es nicht möglich, Daten innerhalb der Vorhaltezeit zu entschlüsseln, dann sollen diese Daten später, wenn sie außerhalb der Vorhaltezeit liegen, ebenfalls nicht entschlüsselt werden können. Im Idealfall werden alte Daten gelöscht.

Schreibt ein Schema vor, dass zusätzlich zu verschlüsselten Vorratsdaten noch persönlich zuordenbare Daten, wie zum Beispiel Name oder Telefonnummer, gespeichert werden, dann können wir ein Verhaltensmuster der Personen erkennen oder die Person identifizieren. Anstatt dessen sollten alle persönlich zuordenbaren Daten verschlüsselt oder in anonymisierter Form gespeichert werden. Falls der Zusammenhang der Daten sichergestellt werden soll, sollten Pseudonyme verwendet werden.

Wir verhindern den Zugriff auf Daten außerhalb der Vorhaltezeit idealerweise, indem wir diese sofort löschen. Leider wird die vollständige Löschung dadurch erschwert, dass auch die Entfernung aller Sicherungen der Daten und gegebenenfalls der Ergebnisse von Analysen erzwungen werden muss. Eine Löschung von Daten ist unmöglich, wenn das Speicherungssystem kompromittiert wurde und Daten kopiert wurden. Ein System, das trotzdem die Nützlichkeit von alten Daten beschränkt, ist zu bevorzugen. Es gewährleistet den Datenschutz auch in diesen Fällen.

Wie bei der Vorratsdatenspeicherung von Kommunikationsdaten ist es häufig so, dass der Aufzeichnende kein Interesse an den gespeicherten Daten hat. Die Internet Service Provider empfinden es eher als Last, wenn sie diese Daten vorhalten müssen. Der Aufzeichnungsprozess des Providers kann zertifiziert und jederzeit durch eine Aufsicht kontrolliert werden. Wir können also davon ausgehen, dass der Provider die Daten ehrlich aufzeichnet und weiterleitet. Auf der anderen Seite zeigt uns die Vergangenheit, dass Provider schnell ihren Eigentümer wechseln können. Dieses führt oft zur Veränderungen der Interessen und eventuell zum Verlust der Vertrauenswürdigkeit. Daher sollte ein Provider die aufgezeichneten Daten nicht speichern. Es ist daher sinnvoll, die Daten getrennt vom Aufzeichnungsort vorzuhalten. Für ein sicheres Schema fordern wir, dass die Daten sogar bei einer nach Zugriff verlangenden Partei gespeichert werden können und diese Partei ohne Zugriffserlaubnis kein Wissen aus den Daten extrahieren kann.

4.2 Vorratsdatenspeicherung mit Schwellwert-Zugriff

Sowohl die Datenspeicherung im Verkehrszentralregister als auch der Transaktionen beim Schutz vor Geldwäsche können wir als ein System zur Vorratsdatenspeicherung mit *Schwellwert* betrachten. Es ist ein bestimmter Zeitraum vorgegeben, in dem die Daten vorgehalten

werden müssen. Diesen Zeitraum nennen wir *Vorhaltezeit*. Alle Daten, die zum aktuellen Zeitpunkt vorgehalten werden müssen, bezeichnen wir als *Daten innerhalb der Vorhaltezeit* oder als *neue Daten*. Daten, die nicht mehr vorgehalten werden müssen, bezeichnen wir dementsprechend als *Daten außerhalb der Vorhaltezeit* oder als *alte Daten*. Übersteigen die Anzahl der Punkte oder die Höhe der Transaktionen in diesem Zeitraum einen bestimmten Wert, dann wird dem Kraftfahrer der Führerschein entzogen oder der Geldwäscher aus dem Verkehr gezogen. Wird der Schwellwert nicht erreicht, so müssen wir keine Aktion vornehmen. Die entsprechenden Personen sollen weder identifiziert werden, noch sollen Personen mit vielen von einer Person mit wenigen Punkten/Transaktionen zu unterscheiden sein.

Wir wollen die Vorratsdatenspeicherung mit Schwellwert-Zugriff anhand der Speicherung von Kommunikationsdaten darstellen und entsprechende Bezeichnungen benutzen. Vorabversionen dieses Ansatzes zur privaten Vorratsdatenspeicherung werden in den Arbeiten [57, 58] vorgestellt. Bei der Aufzeichnung der Vorratsdaten gehen wir davon aus, dass nur unbedingt benötigte Daten auf Vorrat gespeichert. Das Datenaufkommen wird somit minimiert. Durch den Einsatz einer *Blacklist* und des *Schwellwertes* erfüllt unser Schema die Anforderungen des kontrollierten Zugriffs. Zudem werden die Daten getrennt von ihrem Aufzeichnungsort gespeichert. In Abschnitt 4.8 weisen wir nach, dass unser Schema Schutz, Anonymität und Rückwärtssicherheit gewährleistet.

4.2.1 Teilnehmende Parteien

Wir betrachten drei Arten von Parteien.

Provider
: Ein Provider stellt einen Dienst zur Verfügung und zeichnet ehrlich Daten über die Benutzung des Dienstes auf.

Benutzer
: Ein Benutzer nimmt den Dienst eines Providers in Anspruch.

Sammelstelle
: Die Sammelstelle speichert die von den Providern aufgezeichneten Daten und erlangt gegebenenfalls Zugriff auf die gespeicherten Daten.

Die Provider sind ehrliche Parteien. Sie speichern oder manipulieren Vorratsdaten nicht. Ein Provider kann einen Benutzer identifizieren, der seinen Dienst in Anspruch nimmt. Manipuliert ein Benutzer Daten, kann dies der entsprechende Provider feststellen und den Benutzer vom Dienst ausschließen. Die Sammelstelle betrachten wir als einen aktiven Angreifer, der möglichst viel Wissen über die Benutzer gewinnen möchte.

Als Sammelstelle können wir das Kraftfahrt-Bundesamt oder die Finanzaufsicht sehen, als Benutzer den Kraftfahrer oder einen Bankkunden sowie als Provider einen Polizisten oder eine Bank.

4.2.2 Typen von Daten

Wir unterscheiden zwei Typen von Daten, die bei der Benutzung eines Dienstes auftreten können: *Kritische Daten* stehen bei der Vorratsdatenspeicherung im Interesse der Sammelstelle. Sie tragen zur Erreichung des festgelegten Schwellwertes bei. Kritische Daten werden

durch *kritische Aktionen* eines Benutzers ausgelöst. Kritische Aktionen sind beispielsweise das Überfahren einer roten Ampel oder die Überweisung eines großen Geldbetrages. Als *unkritische Daten* bezeichnen wir alle aufgezeichneten Daten, die von *unkritischen Aktionen* ausgelöst werden und nicht zum Erreichen des Schwellwertes beitragen.

4.2.3 Festlegung der aufzuzeichnenden Daten

Die Sammelstelle veranlasst in unserem Szenario die Vorratsdatenspeicherung, speichert alle Daten und fordert Zugriff auf die Daten. Der Zugriffsmechanismus muss vorab geregelt werden und kontrollierbar sein. Wir schlagen daher vor, dass die Sammelstelle zunächst öffentlich festlegt, welche Daten auf Vorrat gespeichert werden sollen. Dies könnten je nach Anwendung beispielsweise die Identität des Benutzers, die IP-Adresse, die Kontonummer oder das Aktenzeichen eines Verkehrsdeliktes sein. Dabei ist zu beachten, dass dies so wenig Daten wie nötig sind, um den Zweck der Vorratsdatenspeicherung zu erfüllen. Nur die als kritisch einzustufenden Daten begründen den Zugriff und damit die Entschlüsselung der Vorratsdaten. Daher muss die Sammelstelle festlegen, welche Aktionen kritisch sind. Dies kann sie den Providern durch eine *Blacklist* mitteilen oder den Providern einen Klassifikator zur Verfügung stellen. Der Provider ist nun in der Lage, die Daten als kritisch oder unkritisch richtig einzustufen. Die Sammelstelle kann durch Veröffentlichung der Merkmale von kritischen Aktionen, der Blacklist oder des Klassifikators der Transparenz gegenüber den Benutzern Rechnung tragen. Ferner kann der Provider unter Umständen einem Benutzer beim Auslösen einer kritischen Aktion dies mitteilen, zum Beispiel vor Betreten einer kritischen Webseite.

4.2.4 Zugriff auf die gespeicherten Daten

In den bisherigen Ansätzen zur Vorratsdatenspeicherung benötigte man eine Legitimation, um Zugriff auf die Daten zu bekommen. Dies reicht vom vorab genehmigten Vollzugriff einer Behörde über den Richterbeschluss bis hin zum Mehr-Augen-Prinzip zur Freigabe der Entschlüsselungsschlüssel für die Vorratsdaten. Für alle diese weichen Zugriffskriterien muss zum Zeitpunkt des Zugriffs auf die Daten ein interaktives Verfahren durchgeführt werden, das die Legitimation prüft, beziehungsweise, die gespeicherten Daten freigibt. Diese Zugriffskriterien sind weich, denn durch die Interaktion besteht Auslegungsspielraum. Damit ist die sichere Umsetzung des Verfahrens und der Ausschluss von Missbrauch schwierig. Wir können in unserem Schema hingegen ein hartes Zugriffskriterium einsetzen, das kein interaktives Verfahren zum Zeitpunkt des Zugriffs benötigt. Die Sammelstelle erhält den Zugriff auf die Daten, wenn ein Benutzer den *Schwellwert an kritischen Aktionen* überschreitet. Sie muss dazu nicht mit anderen Parteien kommunizieren. Die gespeicherten Daten geben selbst bei Erreichen des Schwellwertes den Schlüssel zu ihrer Entschlüsselung frei. Zur technischen Umsetzung verwenden wir dabei Secret-Shares des Entschlüsselungsschlüssels.

4.3 Struktur der Nachrichten

Im Folgenden beschreiben wir die grundlegende Struktur der Daten, die ein Provider aufzeichnet und die Sammelstelle speichert. Da der Provider die aufgezeichneten Daten verschlüsselt und an die Sammelstelle sendet, sprechen wir im weiteren Verlauf von *(kriti-*

schen/unkritischen) Nachrichten. Somit können wir zwischen den aufgezeichneten Daten (Daten) und den gespeicherten Daten (Nachrichten) unterscheiden.

Eine Nachricht M beschreiben wir durch ein 4-Tupel

$$M = (\text{time}, \text{ID}, \text{share}, \text{load}).$$

Der Wert $\text{time}(M)$ bezeichnet dabei die *Runde* (das heißt den Zeitpunkt), in der die Nachricht M generiert wird und $\text{ID}(M)$ eine *ID* der Nachricht, zum Beispiel ein Pseudonym des Senders. $\text{share}(M)$ beschreibt ein oder mehrere *Secret-Shares*, die zu dieser Nachricht gehören. Die *Nutzlast* $\text{load}(M)$ der Nachricht besteht aus weiteren Daten, die wir mit der Nachricht M verbinden. Mit $\text{subj}(M)$ bezeichnen wir den sogenannten *Betreff* oder auch das *Subjekt* der Nachricht M. Das Subjekt ist der Gegenstand des Interesses der Sammelstelle an dieser Nachricht, zum Beispiel Sender und Empfänger einer Mail oder Inhaltsdaten. Im Folgenden nehmen wir an, dass $\text{load}(M)$ das Subjekt der Nachricht in verschlüsselter Form enthält. Wenn die Nachricht M unkritisch ist, wählen wir $\text{ID}(M) = \text{share}(M) = \emptyset$. Die unkritischen Nachrichten erhalten keine Identifikationsmerkmale und tragen nicht zum Zugriff bei, da die Shares fehlen. Um eine unkritische Nachricht einem Benutzer zuzuordnen, muss die Nutzlast einen Identifikator des entsprechenden Benutzers enthalten.

4.4 Übersicht des Schemas

Wir geben nun einen Überblick über das Schema. Auf die genaue Berechnung der Schlüssel, der Nutzlast und der Shares gehen wir danach in den Abschnitten 4.5, 4.6 und 4.7 ein.

Für $\tau \in \mathbb{N}$ bezeichnen wir mit $\mathcal{T}_\tau \subset \mathbb{N}$ den *τ-ten Zeitabschnitt* (Abschnitt von Runden). Dabei können mehrere Zeitabschnitte überlappen. Die Länge eines Zeitabschnitts entspricht der Länge der Vorhaltezeit. Wir nehmen im Folgenden an, dass jeder Zeitabschnitt $\mathcal{T}_\tau$ aus Δ aufeinander folgenden Runden besteht, das heißt,

$$\mathcal{T}_\tau = \{\tau, \ldots, \tau + \Delta - 1\}.$$

Mit geringfügiger Modifikation unseres Schemas können wir auch beliebige Teilmengen der natürlichen Zahlen für $\mathcal{T}_\tau$ verwenden. Π_t bezeichnet *die Menge aller Zeitabschnitte* $\mathcal{T}_\tau$, so dass $t \in \mathcal{T}_\tau$. Außerdem sei $t_{\min}(\mathcal{T}_\tau) = \min_{t \in \mathcal{T}_\tau} t$ die kleinste Nummer einer Runde im Zeitabschnitt $\mathcal{T}_\tau$. Wir nennen daher Runde $t_{\min}(\mathcal{T}_\tau)$ die *erste Runde von* $\mathcal{T}_\tau$.

Initialisierung

1. Die Parteien vereinbaren:
   (a) Kriterien für kritische Aktionen
   (b) welche Daten aufzuzeichnen sind.
   (c) die Länge Δ der Vorhaltezeit/Zeitabschnitte; Vorratsdaten sind für Δ Runden zu speichern.
   (d) den Wert des Schwellwertes d.
2. Für jeden Benutzer generiert sein Provider zufällig seinen Seed S und sein Pseudonym P. Das Pseudonym ist eindeutig. Zwei unterschiedliche Benutzer haben unterschiedliche Pseudonyme. Zudem wird das Pseudonym unabhängig von der Identität des Benutzers gewählt, so dass der Wert keine Information über den Benutzer enthält.

Aufzeichnung von Daten und Generierung von Nachrichten

1. Ein Benutzer führt eine Aktion in Runde t aus.
2. Der Provider zeichnet die Daten der Aktion auf.
3. Der Provider klassifiziert die Aktion als kritisch oder unkritisch.
4. Der Provider generiert eine Nachricht M.
   (a) Falls die Aktion kritisch ist, generiert der Provider eine kritische Nachricht. Dazu setzt er
   $$\begin{aligned} \text{time}(M) &= t, \\ \text{ID}(M) &= P \end{aligned}$$
   und berechnet $\text{share}(M)$ sowie $\text{load}(M)$ (siehe Abschnitte 4.6 und 4.7).
   (b) Falls die Aktion unkritisch ist, so generiert der Provider eine unkritische Nachricht. Dazu setzt er
   $$\begin{aligned} \text{time}(M) &= t, \\ \text{ID}(M) &= \emptyset, \\ \text{share}(M) &= \emptyset \end{aligned}$$
   und berechnet $\text{load}(M)$ (siehe Abschnitt 4.6).
5. Der Provider sendet M an die Sammelstelle zur Speicherung.

Speicherung und Entschlüsselung von Nachrichten

1. Die Sammelstelle erhält eine Nachricht M zur Speicherung.
2. Wenn die Sammelstelle innerhalb eines Zeitabschnittes $\mathcal{T}_j$ von Δ Runden d oder mehr kritische Nachrichten mit der $\text{ID}(M)$ erhalten hat, kann sie folgende Schritte ausführen:
   (a) Durch Lagrange-Interpolation berechnet sie aus d dieser kritischen Nachrichten den Wert $\text{seed}^{(t_{\min}(\mathcal{T}_j))}(S)$ (siehe Abschnitt 4.5) sowie die Identität $\mathcal{I}_i$ des Benutzers, der die entsprechenden kritischen Aktionen ausgeführt hat (siehe Abschnitt 4.7).
   (b) Aus $\text{seed}^{(t_{\min}(\mathcal{T}_j))}(S)$ generiert die Sammelstelle die Entschlüsselungsschlüssel aller Runden $t' \geq t_{\min}(\mathcal{T}_j)$ (siehe Abschnitte 4.5 und 4.6).
   (c) Mithilfe der Identität $\mathcal{I}_i$ und eines Fingerabdrucks der Nachrichten identifiziert die Sammelstelle alle unkritischen Nachrichten des korrespondierenden Benutzers (siehe Abschnitt 4.6).
   (d) Die Sammelstelle entschlüsselt den Inhalt $\text{subj}(M')$ für alle Nachrichten M' des Benutzers, die in Runde $t_{\min}(\mathcal{T}_j)$ oder später generiert wurden.

4.5 Bestimmung der Schlüssel

Wir wählen die Länge einer Runde derart, dass ein Benutzer pro Runde höchstens eine Aktion ausführt. Der Provider muss daher in einer Runde höchstens eine Nachricht für einen Benutzer generieren. Für jede Runde bestimmen wir die Schlüssel zur Verschlüsselung aus den Werten der Vorrunde. Dazu wählt der Provider für jeden Benutzer zu Beginn der Durchführung des Schemas einen zufälligen Seed $S \in \{0,1\}^\ell$.

Sei $\widehat{G}\colon \{0,1\}^\ell \to \{0,1\}^{3\ell}$ ein Pseudozufallszahlengenerator. Wir definieren die *Seedgenerierungsfunktion* $\text{seed}^{(i)}\colon \{0,1\}^\ell \to \{0,1\}^\ell$ und die *pseudozufällige Stringfunktion* $\text{rand}^{(i)}\colon \{0,1\}^\ell \to \{0,1\}^{2\ell}$ durch

1. $\text{seed}^{(0)}(S) = S$.
2. $\text{seed}^{(i)}(S) = \widehat{G}^{(i)}(S)_{\{1,\ldots,\ell\}}$ für $i \geq 1$.
3. $\text{rand}^{(i)}(S) = \widehat{G}^{(i+1)}(S)_{\{\ell+1,\ldots,3\ell\}}$ für $i \geq 0$.

Für die Definition von $\widehat{G}^{(i)}$ verweisen wir auf Abschnitt 3.3. Bei deren Kenntnis können wir von $\text{seed}^{(i)}(S)$ die Werte $\text{seed}^{(i+1)}(S)$ und $\text{rand}^{(i)}(S)$ durch

$$\text{seed}^{(i+1)}(S) \circ \text{rand}^{(i)}(S) = \widehat{G}(\text{seed}^{(i)}(S))$$

berechnen. Somit können wir beginnend mit einem Seed S die Sequenzen $\{\text{seed}^{(i)}(S)\}_{i=0}^{\tau}$ und $\{\text{rand}^{(i)}(S)\}_{i=0}^{\tau}$ effizient berechnen. Für die Runde t werden wir $\text{rand}^{(t)}(S)$ als Schlüssel zur Verschlüsselung benutzen. Mit wachsendem τ nimmt die Wahrscheinlichkeit zu, zwischen echt zufälligen und pseudozufälligen Strings unterscheiden zu können. Wir können dem entgegentreten, indem wir den Wert des Sicherheitsparameters ℓ erhöhen. Für den Fall, dass τ sehr groß wird, können wir den Seed $\text{seed}^{(t)}(S)$ in einem vorgegebenen Turnus durch einen neuen Zufallsstring ersetzen.

Obwohl wir die Strings $\text{seed}^{(j)}(S)$ und $\text{rand}^{(j)}(S)$ für $j \geq i$ auf Eingabe $\text{seed}^{(i)}(S)$ effizient generieren können, kann kein probabilistischer Polynomialzeitalgorithmus auf Eingabe $\text{seed}^{(i)}(S)$ oder $\text{rand}^{(i)}(S)$ Wissen über die Strings $\text{seed}^{(k)}(S)$ und $\text{rand}^{(k-1)}(S)$ für $k < i$ mit mehr als vernachlässigbarer Wahrscheinlichkeit gewinnen. Dies folgt aus der kryptographischen Sicherheit des Pseudozufallszahlengenerators $\widehat{G}$. Die dem zugrundeliegende Pseudozufälligkeit von $\{\text{rand}^{(i)}(S)\}_{i=0}^{\tau}$ weisen wir in Theorem 4.8 nach.

Zunächst folgen einige, teilweise bekannte Vorüberlegungen. Da uns keine Quellen mit entsprechenden Beweisen der Eigenschaften aus der Literatur bekannt sind, geben wir eigene Beweise an.

Lemma 4.1 *Sei $J \subseteq \{0,\ldots,\ell_1 - 1\}$ und $G\colon \{0,1\}^{\ell_0} \to \{0,1\}^{\ell_1}$ ein Pseudozufallszahlengenerator. Dann ist die Ausgabe von G_J, das heißt, die Bits der Ausgabe von G mit Index in J, pseudozufällig.*

BEWEIS Es seien U_{ℓ_0}, U_{ℓ_1} und $U_{|J|}$ unabhängige und uniforme Zufallsvariablen über den Strings der Längen ℓ_0, ℓ_1 und $|J|$. Angenommen, die Ausgabe von G_J ist nicht pseudozufällig. Dann gibt es einen probabilistischen Polynomialzeitalgorithmus $\mathcal{D}$ und ein Polynom q, so dass für genügend große ℓ_0 gilt:

$$|\Pr[\mathcal{D}(1^{\ell_0}, G_J(U_{\ell_0})) = 1] - \Pr[\mathcal{D}(1^{\ell_0}, U_{|J|})]| \geq \frac{1}{q(\ell_0)}.$$

Wir konstruieren den probabilistischen Polynomialzeitalgorithmus $\mathcal{D}'$ als Unterscheider für G. Für einen String $\alpha \in \{0,1\}^{\ell_1}$ sei

$$\mathcal{D}'(1^{\ell_0}, \alpha) = \mathcal{D}(1^{\ell_0}, \alpha_J).$$

Es ist leicht zu sehen, dass $\mathcal{D}'$ in Polynomialzeit berechenbar ist. Zudem gilt:

$$\Pr[\mathcal{D}'(1^{\ell_0}, G(U_{\ell_0})) = 1] = \Pr[\mathcal{D}(1^{\ell_0}, G(U_{\ell_0})_J) = 1] = \Pr[\mathcal{D}(1^{\ell_0}, G_J(U_{\ell_0})) = 1]$$

und

$$\Pr[\mathcal{D}'(1^{\ell_0}, U_{\ell_1}) = 1] = \Pr[\mathcal{D}(1^{\ell_0}, (U_{\ell_1})_J) = 1] = \Pr[\mathcal{D}(1^{\ell_0}, U_{|J|}) = 1].$$

Daraus können wir schließen, dass

$$|\Pr[\mathcal{D}'(1^{\ell_0}, G(U_{\ell_0})) = 1] - \Pr[\mathcal{D}'(1^{\ell_0}, U_{\ell_1}) = 1]| \geq \frac{1}{q(\ell_0)}.$$

Dies ist ein Widerspruch zur Annahme, dass G ein Pseudozufallszahlengenerator ist. ∎

Lemma 4.2 *Seien $G\colon \{0,1\}^{\ell_0} \to \{0,1\}^{\ell_1}$ und $G'\colon \{0,1\}^{\ell_1} \to \{0,1\}^{\ell_2}$ Pseudozufallszahlengeneratoren. Dann ist auch $G \circ G'$ ein Pseudozufallszahlengenerator.*

BEWEIS Es seien U_{ℓ_0}, U_{ℓ_1} und U_{ℓ_2} unabhängige und uniforme Zufallsvariablen über den Strings der Längen ℓ_0, ℓ_1 und ℓ_2. Angenommen $G \circ G'$ ist kein Pseudozufallszahlengenerator. Dann gibt es einen probabilistischen Polynomialzeitalgorithmus $\mathcal{D}$ und ein Polynom q, so dass für genügend große ℓ_0 gilt:

$$|\Pr[\mathcal{D}(1^{\ell_0}, G'(G(U_{\ell_0}))) = 1] - \Pr[\mathcal{D}(1^{\ell_0}, U_{\ell_2}) = 1]| \geq \frac{1}{q(\ell_0)}.$$

Für den probabilistischen Polynomialzeitalgorithmus $\mathcal{D}'$, definiert durch

$$\mathcal{D}'(1^{\ell_0}, \alpha) = \mathcal{D}(1^{\ell_0}, G'(\alpha))$$

für alle $\alpha \in \{0,1\}^{\ell_1}$, erhalten wir

$$\Pr[\mathcal{D}'(1^{\ell_0}, G(U_{\ell_0})) = 1] = \Pr[\mathcal{D}(1^{\ell_0}, G'(G(U_{\ell_0}))) = 1] \text{ und}$$
$$\Pr[\mathcal{D}'(1^{\ell_0}, U_{\ell_1}) = 1] = \Pr[\mathcal{D}(1^{\ell_0}, G'(U_{\ell_1})) = 1]$$

Da G und G' Pseudozufallszahlengeneratoren sind, gilt für alle probabilistischen Polynomialzeitalgorithmen $\mathcal{A}$, alle Polynome p und alle genügend großen ℓ_0:

$$|\Pr[\mathcal{A}(1^{\ell_0}, G(U_{\ell_0})) = 1] - \Pr[\mathcal{A}(1^{\ell_0}, U_{\ell_1}) = 1]| \leq \frac{1}{p(\ell_0)}$$

und

$$|\Pr[\mathcal{A}(1^{\ell_0}, G'(U_{\ell_1})) = 1] - \Pr[\mathcal{A}(1^{\ell_0}, U_{\ell_2}) = 1]| \leq \frac{1}{p(\ell_0)}.$$

Die zweite Ungleichung gilt, weil ℓ_1 polynomiell in ℓ_0 ist und somit ℓ_1 von einem effizienten Algorithmus auf Eingabe ℓ_0 berechnet werden kann. Insbesondere gelten diese Aussagen

auch für $p = 2q$ und für $\mathcal{A} = \mathcal{D}'$. Folglich erhalten wir mithilfe der Dreiecksungleichung, dass

$$\begin{aligned}
&|\Pr[\mathcal{D}'(1^{\ell_0}, G(U_{\ell_0})) = 1] - \Pr[\mathcal{D}'(1^{\ell_0}, U_{\ell_1}) = 1]| \\
&= |\Pr[\mathcal{D}(1^{\ell_0}, G'(G(U_{\ell_0}))) = 1] - \Pr[\mathcal{D}(1^{\ell_0}, G'(U_{\ell_1})) = 1]| \\
&= |\Pr[\mathcal{D}(1^{\ell_0}, G'(G(U_{\ell_0}))) = 1] - \Pr[\mathcal{D}(1^{\ell_0}, U_{\ell_2}) = 1] \\
&\qquad + \Pr[\mathcal{D}(1^{\ell_0}, U_{\ell_2}) = 1] - \Pr[\mathcal{D}(1^{\ell_0}, G'(U_{\ell_1})) = 1]| \\
&\geq |\Pr[\mathcal{D}(1^{\ell_0}, G'(G(U_{\ell_0}))) = 1] - \Pr[\mathcal{D}(1^{\ell_0}, U_{\ell_2}) = 1]| \\
&\qquad - |\Pr[\mathcal{D}(1^{\ell_0}, U_{\ell_2}) = 1] - \Pr[\mathcal{D}(1^{\ell_0}, G'(U_{\ell_1})) = 1]| \\
&\geq \frac{1}{q(\ell_0)} - \frac{1}{2q(\ell_0)} \\
&= \frac{1}{2q(\ell_0)}.
\end{aligned}$$

Also ist G kein Pseudozufallszahlengenerator. Dies widerspricht der Voraussetzung des Lemmas. ∎

Durch polynomiell häufige Anwendung des vorigen Lemmas erhalten wir:

Korollar 4.3 *Seien $G_1, \ldots, G_\tau$ Pseudozufallszahlengeneratoren, wobei τ polynomiell in ℓ ist und für $1 \leq j \leq \tau$ gilt, dass $G_j \colon \{0,1\}^{\ell_{j-1}} \to \{0,1\}^{\ell_j}$ und $\ell_0 = \ell$. Dann ist*

$$G = G_1 \circ \ldots \circ G_\tau$$

ein Pseudozufallszahlengenerator.

Dabei ist es wichtig darauf zu achten, dass τ und somit auch ℓ_τ polynomiell in ℓ sind. Andernfalls erreicht man durch Verlängerung der Eingabe, dass ein Algorithmus zum Brechen von G_1 einen nicht vernachlässigbaren Vorteil erhalten könnte.

Lemma 4.4 *Sei $G_1 \colon \{0,1\}^{\ell_0} \to \{0,1\}^{\ell_2}$ ein Pseudozufallszahlengenerator und ℓ_1 sowie ℓ_3 polynomiell in ℓ_0. Dann ist*

$$G(S) = S_{1,\ldots,\ell_1} \circ G_1(S_{\ell_1+1,\ldots,\ell_1+\ell_0}) \circ S_{\ell_1+\ell_0+1,\ldots,\ell_1+\ell_0+\ell_3}$$

ein Pseudozufallszahlengenerator.

BEWEIS Angenommen, G ist kein Pseudozufallszahlengenerator. Da ℓ_1, ℓ_2 und ℓ_3 polynomiell in ℓ_0 sind, gibt es einen probabilistischen Polynomialzeitalgorithmus $\mathcal{D}$ und ein Polynom q, so dass für unendlich viele Werte ℓ_0 gilt:

$$|\Pr[\mathcal{D}(1^{\ell_0}, G(U_{\ell_1+\ell_0+\ell_3})) = 1] - \Pr[\mathcal{D}(1^{\ell_0}, U_{\ell_1+\ell_2+\ell_3}) = 1]| \geq \frac{1}{q(\ell_0)},$$

wobei auch hier für alle $j \in \mathbb{N}$ die Zufallsvariablen U_j unabhängig und uniform über den Strings der Länge j verteilt sind. Da G_1 ein Pseudozufallszahlengenerator ist, gilt für alle probabilistischen Polynomialzeitalgorithmen $\mathcal{A}$, alle Polynome p und genügend große Werte ℓ_0:

$$|\Pr[\mathcal{A}(1^{\ell_0}, G_1(U_{\ell_0})) = 1] - \Pr[\mathcal{A}(1^{\ell_0}, U_{\ell_2}) = 1]| < \frac{1}{p(\ell_0)}.$$

Da ℓ_1 und ℓ_3 polynomiell in ℓ_0 sind, sowie U_{ℓ_1} und U_{ℓ_3} unabhängig von U_{ℓ_0} und $G_1(U_{\ell_0})$ sind, können wir folgern, dass

$$|\Pr[\mathcal{A}(1^{\ell_0}, U_{\ell_1} \circ G_1(U_{\ell_0}) \circ U_{\ell_3}) = 1] - \Pr[\mathcal{A}(1^{\ell_0}, U_{\ell_1} \circ U_{\ell_2} \circ U_{\ell_3}) = 1]| < \frac{1}{p(\ell_0)}.$$

Dies ist ein Widerspruch dazu, dass es einen Unterscheider $\mathcal{D}$ für G gibt und G kein Pseudozufallszahlengenerator ist. Somit folgt die Aussage des Lemmas. ∎

Korollar 4.5 *Seien $G_1\colon \{0,1\}^{\ell_0} \to \{0,1\}^{\ell_2}$ und $G_2\colon \{0,1\}^{\ell_1} \to \{0,1\}^{\ell_3}$ Pseudozufallszahlengeneratoren, wobei ℓ_1 polynomiell in ℓ_0 ist. Dann ist $G\colon \{0,1\}^{\ell_0+\ell_1} \to \{0,1\}^{\ell_2+\ell_3}$, gegeben durch*

$$G(S) = G_1(S_{1,\ldots,\ell_0}) \circ G_2(S_{\ell_0+1,\ldots,\ell_0+\ell_1})$$

für alle $S \in \{0,1\}^{\ell_0+\ell_1}$, ein Pseudozufallszahlengenerator.

BEWEIS Mit Lemma 4.4 ist es leicht zu sehen, dass

$$G_3(S_{1,\ldots,\ell_0+\ell_1}) = G_1(S_{1,\ldots,\ell_0}) \circ S_{\ell_0+1,\ldots,\ell_0+\ell_1}$$

und

$$G_4(S_{1,\ldots,\ell_2+\ell_1}) = S_{1,\ldots,\ell_2} \circ G_2(S_{\ell_2+1,\ldots,\ell_2+\ell_1})$$

Pseudozufallszahlengeneratoren sind. Damit ist auch $G = G_3 \circ G_4$ nach Lemma 4.2 ein Pseudozufallszahlengenerator. ∎

Durch polynomiell häufiges Anwenden von Lemma 4.4 erhalten wir analog:

Korollar 4.6 *Seien $G_1, \ldots, G_k$ Pseudozufallszahlengeneratoren mit $G_j\colon \{0,1\}^{\ell_j} \to \{0,1\}^{\ell'_j}$, wobei k sowie $\ell'_1, \ell_2, \ell'_2, \ldots, \ell_k, \ell'_k$ polynomiell in ℓ_1 sind. Dann ist $G\colon \{0,1\}^{\ell_1+\ldots+\ell_k} \to \{0,1\}^{\ell'_1+\ldots+\ell'_k}$, gegeben durch*

$$G(S) = G_1(S_{1,\ldots,\ell_1}) \circ G_2(S_{\ell_1+1,\ldots,\ell_1+\ell_2}) \circ \ldots \circ G_k(S_{(\ell_1+\ldots+\ell_{k-1}+1),\ldots,(\ell_1+\ldots+\ell_k)})$$

für alle $S \in \{0,1\}^{\ell_1+\ldots+\ell_k}$, ein Pseudozufallszahlengenerator.

Nun werden wir nachweisen, dass die Strings, aus denen wir später die Schlüssel generieren, pseudozufällig sind.

Sei $\widetilde{G}^{[\tau]}\colon \{0,1\}^{\ell+2\tau\ell} \to \{0,1\}^{\ell+2(\tau+1)\ell}$ definiert durch

$$\widetilde{G}^{[\tau]}(S) = \widehat{G}(S_{1,\ldots,\ell}) \circ S_{\ell+1,\ldots,2\tau\cdot\ell},$$

für alle $S \in \{0,1\}^{\ell+2\tau\cdot\ell}$, $\tau \in \mathbb{N}$ und ℓ der Sicherheitsparameter für $\widehat{G}$ (siehe Anfang dieses Abschnitts 4.5). Da $\widehat{G}$ ein Pseudozufallszahlengenerator ist, können wir direkt aus Lemma 4.4 schließen:

Korollar 4.7 *Sei τ polynomiell in ℓ. Dann ist $\widetilde{G}^{[\tau]}$ ein Pseudozufallszahlengenerator.*

Theorem 4.8 *Sei τ polynomiell in ℓ. Dann ist*

$$\{\mathrm{rand}^{(\tau)}(S)\}_{i=0}^{\tau} = \mathrm{rand}^{(\tau)}(S) \circ \ldots \circ \mathrm{rand}^{(0)}(S) \text{ mit } S \in \{0,1\}^{\ell}$$

ein Pseudozufallszahlengenerator.

BEWEIS Wir können $\{\text{rand}^{(\tau)}(S)\}_{i=0}^{\tau}$ durch Hintereinanderausführung von Instanzen von $\widetilde{G}^{[\cdot]}$ ausdrücken. Nach Konstruktion von $\text{seed}^{(\cdot)}(S)$ und $\text{rand}^{(\cdot)}(S)$ gilt:

$$\widetilde{G}^{[0]} \circ \ldots \circ \widetilde{G}^{[\tau]}(S) = \text{seed}^{(\tau+1)}(S) \circ \text{rand}^{(\tau)}(S) \circ \ldots \circ \text{rand}^{(0)}(S).$$

Da τ polynomiell in ℓ ist, ist $\widetilde{G}^{[0]} \circ \ldots \circ \widetilde{G}^{[\tau]}$ nach Korollar 4.3 ein Pseudozufallszahlengenerator. Folglich ist mit Lemma 4.1

$$\{\text{rand}^{(i)}(S)\}_{i=0}^{\tau} = \widetilde{G}^{[0]} \circ \ldots \circ \widetilde{G}^{[\tau]}(S)_{\ell+1,\ldots,\ell+2(\tau+1)\ell}$$

ein Pseudozufallszahlengenerator. ∎

4.6 Verschlüsselung der Nutzlast und Identifikation einer Nachricht

Sei n die *Anzahl aller Benutzer*. Die Identität eines Benutzers sei ein eindeutiges Codewort eines binären Blockcodes $\mathcal{I}$ mit Hammingabstand δ und mit mehr als n Codewörtern. Falls ein Provider eine Aktion des Benutzers mit Identität $\mathcal{I}_i$ aufzeichnet, so muss er eine Nachricht M zur Speicherung vorbereiten und die zu speichernden Daten verschlüsseln. Dabei wird $\text{subj}(M)$ in die Nutzlast $\text{load}(M)$ eingefügt. Anschließend sendet der Provider die Nachricht M an die Sammelstelle. Diese speichert M in einem Nachrichtenpool. Wenn die Sammelstelle den Entschlüsselungsschlüssel erhalten hat, soll sie alle Nachrichten identifizieren können, die mit diesem Schlüssel entschlüsselt werden können.

4.6.1 Aufbau und Verschlüsselung der Nutzlast

Für den Identifikationsmechanismus führen wir einen zufälligen *Indikatorstring* R, einen pseudozufälligen String L und einen verschlüsselten Fingerabdruck $\text{fp}_{R,L}(\mathcal{I}_i)$ ein. Die Implementierung des Fingerabdrucks basiert auf Naors Bit-Commitment-Schema (siehe Abschnitt 3.10 und [82]). Die Struktur der Nutzlast $\text{load}(M)$ ist

$$\text{load}(M) = (R,\ \text{fp}_{R,L}(\mathcal{I}_i),\ \text{enc}_K(\text{subj}(M) \circ \mathcal{I}_i)).$$

Dabei sei enc_K die Verschlüsselung mit einem pseudozufälligen String, das heißt,

$$\text{enc}_K(\text{subj}(M) \circ \mathcal{I}_i) = G'(K) \oplus (\text{subj}(M) \circ \mathcal{I}_i),$$

wobei G' ein Pseudozufallszahlengenerator ist, der einen String der Länge $|\text{subj}(M) \circ \mathcal{I}_i|$ erzeugt. Aus technischen Gründen nehmen wir an, dass für alle Nachrichten die Subjekte unabhängig sind, sowie dass für alle Nachrichten und Identitäten der String $|\text{subj}(M) \circ \mathcal{I}_i|$ die gleiche Länge hat. Für enc könnten wir auch die Verschlüsselungsfunktion eines sicheren symmetrischen Verschlüsselungsschemas verwenden. Allerdings vereinfacht die Verwendung der `XOR`-Operation mit einem pseudozufälligen String die Analyse. Wir nehmen an, dass $\text{subj}(M)$ unabhängig von der Identität $\mathcal{I}_i$ ist. Sei t die Runde, in der M gesendet beziehungsweise generiert wurde. Für den zufälligen Seed $S \in \{0,1\}^{\ell}$ des Benutzers erhalten wir die Schlüssel $K \in \{0,1\}^{\ell}$ und $L \in \{0,1\}^{\ell}$ aus $\text{rand}^{(t)}(S)$ durch

$$\text{rand}^{(t)}(S) = K \circ L.$$

Auf die Sicherheit der Verschlüsselung gehen wir in Abschnitt 4.8 ein.

4.6.2 Identifikationsmechanismus der Nachricht

Sei $m \in \mathbb{N}$ der Sicherheitsparameter des Fingerabdrucks und $G'' \colon \{0,1\}^{\ell} \to \{0,1\}^{m \cdot |\mathcal{I}_i|}$ ein Pseudozufallszahlengenerator. Es sei

$$G''(L) = B_1 \circ \ldots \circ B_{|\mathcal{I}_i|},$$

mit $B_j \in \{0,1\}^m$. Zudem gelte für den Indikatorstring $R = R_1 \circ \ldots \circ R_{|\mathcal{I}_i|} \in \{0,1\}^{m \cdot |\mathcal{I}_i|}$, wobei jeder Block $R_i \in \{0,1\}^m$ ein zufälliger String ist, der mindestens eine 1 enthält. Es bezeichne b_j das j-te Bit von $\mathcal{I}_i$. Wir definieren den Fingerabdruck

$$\mathrm{fp}_{R,L}(\mathcal{I}_i) = \tilde{B}_1 \circ \ldots \circ \tilde{B}_{|\mathcal{I}_i|},$$

so dass gilt:

$$\tilde{B}_j = \begin{cases} B_j, & \text{falls } b_j = 0, \\ B_j \oplus R_j, & \text{falls } b_j = 1. \end{cases}$$

Haben wir Zugriff auf die Nachricht M und die Schlüssel K and L, dann können wir $\mathcal{I}_i$ entschlüsseln und überprüfen, ob dieser Wert dem Fingerabdruck $\mathrm{fp}_{R,L}(\mathcal{I}_i)$ entspricht. Falls $\mathrm{seed}^{(t)}(S)$ bekannt ist, können wir die entsprechenden Schlüssel K und L berechnen.

Der initiale Wert des Seeds S wird für jeden Benutzer unabhängig gewählt. Seien K' und L' zwei Schlüssel, die von einem Seed $\mathrm{seed}^{(t)}(S')$ stammen. Dabei ist S' der Seed, den wir der Identität $\mathcal{I}_j \neq \mathcal{I}_i$ zuordnen. Dann sollten sich $\mathrm{fp}_{K,L}(\mathcal{I}_i)$ und $\mathrm{fp}_{K',L'}(\mathcal{I}_j)$ für jede Identität $\mathcal{I}_j \neq \mathcal{I}_i$ unterscheiden, sonst würden wir möglicherweise die Nachricht M dem falschen Benutzer zuordnen. Wenn dieser unerwünschte Fall eintritt, das heißt, wir assoziieren die Nachricht M mit der falschen Identität $\mathcal{I}_j$, dann sagen wir, dass eine *Kollision* auftritt. Eine Kollision kann nur dann auftreten, wenn die Werte R, R', $\mathrm{fp}_{R,L}(\mathcal{I}_i)$, $\mathrm{fp}_{R',L'}(\mathcal{I}_j)$ sowie Schlüssel L und L' für eine Runde t auftreten und

$$\mathrm{fp}_{R,L'}(\mathcal{I}_i) = \mathrm{fp}_{R',L'}(\mathcal{I}_j),$$

wobei $K'' \circ L'' = \mathrm{rand}^{(t'-t'')}(S_h)$. Gibt es für einen Fingerabdruck keine Kollision und sind die Schlüssel K und L bekannt, dann können wir den Benutzer einer Nachricht korrekt identifizieren.

Wie bereits beschrieben, gehen wir davon aus, dass jeder Benutzer höchstens eine Aktion in einer Runde ausführt. Zudem nehmen wir an, dass mit Wahrscheinlichkeit $(1-q)$ die Tupel $(t, K, G''(L))$ mit $K \circ L = \mathrm{rand}^{(t)}(S)$ für alle Nachrichten unterschiedlich sind. Wir können durch Verlängerung der Seedlänge sicherstellen, dass q sehr klein ist.

Lemma 4.9 *Sei $m \geq \ell + 1$. Dann beträgt die Wahrscheinlichkeit, dass eine Kollision für eine Nachricht M auftritt, höchstens $2^{-(\delta-1)\ell}$.*

BEWEIS Sei M eine Nachricht die von Identität $\mathcal{I}_i$ in Runde t ausgelöst wird. Die Nutzlast $\mathrm{load}(M)$ wurde mithilfe von $\mathrm{seed}^{(t)}(S)$ und $\mathrm{rand}^{(t)}(S) = K \circ L$ verschlüsselt:

$$\mathrm{load}(M) = (R, \mathrm{fp}_{R,L}(\mathcal{I}_i), \mathrm{enc}_K(\mathrm{subj}(M) \circ \mathcal{I}_i)).$$

Mit Wahrscheinlichkeit höchstens q stimmt $(t, G''(L))$ mit dem entsprechenden Paar einer anderen Nachricht überein und wir können die Identität $\mathcal{I}_j$ entschlüsseln. Für $\mathcal{I}_i \neq \mathcal{I}_j$ tritt

keine Kollision auf. Im Folgenden betrachten wir daher den Fall, dass die entsprechenden Tupel $(t, K, G''(L))$ ungleich sind.

Falls $G''(L) \neq G''(L')$, verfahren wir analog zum Beweis von Naor [82]. Sei $K \circ L = \text{rand}^{(t)}(S)$ und $K' \circ L' = \text{rand}^{(t)}(S')$, wobei S der Seed von $\mathcal{I}_i$ und S' ein beliebiger Seed ist, den wir $\mathcal{I}_j$ zuordnen. Ferner sei $G'(L) = B_1 \circ \ldots \circ B_{|\mathcal{I}|}$ und $G'(L') = B'_1 \circ \ldots \circ B'_{|\mathcal{I}|}$. Wir werden nun die Kollisionswahrscheinlichkeit in diesem Fall abschätzen, das heißt, die Wahrscheinlichkeit, dass M der falschen Identität $\mathcal{I}_j$ zugeordnet werden kann. Wir betrachten zunächst den Fall, dass $\mathcal{I}_j$ für S' gegeben ist.

Da die Codewörter des Codes $\mathcal{I}$ einen Hammingabstand von δ haben, gilt für alle Paare von Codewörtern $\mathcal{I}_i \neq \mathcal{I}_j$, dass es mindestens δ Positionen $\kappa = \{k_1, \ldots, k_\delta\}$ gibt, an denen sich diese beiden Codewörter unterscheiden. Eine Kollision tritt nur auf, falls für alle $k \in \kappa$ gilt: $\tilde{B}_k = \tilde{B}'_k$ und somit $B_k = R_k \oplus B'_k$. Die Bits in B_k and B'_k unterscheiden sich genau an den Positionen, an denen R_k den Wert 1 hat. An allen anderen Positionen müssen die beiden Blöcke übereinstimmen. Hätten wir also einen anderen Wert für R_k gewählt, dann wäre keine Kollision aufgetreten beziehungsweise die Identifikation wäre fehlgeschlagen, da wir einen Fehler in der Codierung erkannt hätten.

Da jeder pseudozufällige String L vom ursprünglichen Seed S abhängt, gibt es höchstens 2^ℓ unterschiedliche Wahlmöglichkeiten für L'. Für jedes R_k mit $k \in \kappa$ gibt es $2^m - 1 > 2^{m-1}$ mögliche Strings. Folglich gibt es mehr als $2^{\delta(m-1)}$ unterschiedliche Wahlmöglichkeiten für $R_{k_1}, \ldots, R_{k_\delta}$.

Somit beträgt die Kollisionswahrscheinlichkeit höchstens

$$\begin{aligned} &\Pr[G''(L) \neq G''(L')] \cdot \frac{2^\ell}{2^{\delta(m-1)}} \\ &\leq \frac{2^\ell}{2^{\delta(m-1)}} \\ &\leq 2^{-(\delta-1)\ell} \quad \text{für } m \geq \ell + 1. \end{aligned}$$

Wir müssen nun noch die Wahrscheinlichkeit für eine Kollision in dem Fall untersuchen, wenn wir durch eine schlechte Wahl eines Seeds die Nachricht M einem Benutzer zuordnen, der vor der Analyse von M noch nicht näher bestimmt wurde. Hierbei ergibt sich die Wahl von $\mathcal{I}_j$ durch die Entschlüsselung von $\text{enc}_K(\text{subj}(M) \circ \mathcal{I}_i)$ mithilfe von K'. Wir unterscheiden die folgenden Fälle:

1. $\mathcal{I}_i = \mathcal{I}_j$, das heißt, es liegt keine Kollision vor.
2. $\mathcal{I}_i \neq \mathcal{I}_j$. Hier erhalten wir die Wahrscheinlichkeit analog zu unserer Analyse für ein gegebenes $\mathcal{I}_j$. ∎

Demnach ist es sehr unwahrscheinlich, dass eine Nachricht M mit einem falschen Benutzer assoziiert wird. Eine Kollision unter den Nachrichten ist nur von Interesse, wenn die Nachrichten innerhalb der Vorhaltezeit liegen. Sei a die Anzahl der gespeicherten Nachrichten in diesem Zeitraum. Die Wahrscheinlichkeit, dass keine Kollision unter allen Nachrichten auftritt, können wir durch $a \cdot 2^{-(\delta-1)\ell}$ nach oben abschätzen. Somit reichen Werte für δ und ℓ im Bereich von $\Omega(\sqrt{\log(a)})$ aus.

4.7 Aufbau der Shares

In den bisherigen Ansätzen zur Vorratsdatenspeicherung werden die Daten bei den Providern gespeichert. Die Sammelstelle erlangt Zugriff auf die Vorratsdaten, indem sie die Daten von

dem Provider anfordert. Idealerweise geschieht dies erst nach juristischer Legitimation. Wir schlagen vor, dass die kritischen Nachrichten selbst zu ihrer Entschlüsselung beitragen. Die Daten erlauben ihre Entschlüsselung nur dann, wenn der Provider zu viele kritische Aktionen eines Benutzers innerhalb eines bestimmten Zeitraums aufgezeichnet und entsprechende Nachrichten an die Sammelstelle geschickt hat. Daher kann die Sammelstelle die Daten selber speichern und muss zur Entschlüsselung nicht mit dem Provider in Kontakt treten.

Damit bei Erreichen des Schwellwertes die Nachrichten ohne weitere Hilfe entschlüsselt werden können, weisen wir jeder Nachricht entsprechende Shamir-Shares der Schlüssel zu: Für jeden Benutzer und jeden Zeitabschnitt $\mathcal{T}_\tau$ generieren wir uniform und unabhängig ein zufälliges Polynom p_τ vom Grad $d-1$ über einem genügend großen endlichen Körper $\mathcal{F}$, so dass

$$p_\tau(0) = \text{seed}^{(t_{\min}(\mathcal{T}_\tau))}(S) \circ \mathcal{I}_i.$$

Dazu wählen wir $d-1$ Koeffizienten a_j mit $1 \leq j \leq d-1$ uniform und unabhängig aus $\mathcal{F}$ und berechnen:

$$p_\tau(x) = (\text{seed}^{(t_{\min}(\mathcal{T}_\tau))}(S) \circ \mathcal{I}_i) + \sum_{j=1}^{d-1} a_j x^j.$$

Wir interpretieren die Konkatenation aus $\text{seed}^{(t_{\min}(\mathcal{T}_\tau))}(S)$ und der Identität $\mathcal{I}_i$ des Benutzers als ein entsprechendes Element in $\mathcal{F}$. Diese Zuordnung ist einfach zu realisieren, zum Beispiel durch die Interpretation als Binärzahl in einem Restklassenkörper modulo einer genügend großen Primzahl.

Für jede kritische Nachricht M sendet der Provider neben der verschlüsselten Nachricht noch eine Sammlung von Shares an die Sammelstelle. Diese Shares $\text{share}(M)$ berechnen wir wie folgt:

Wir initialisieren $\text{share}(M) = \emptyset$. Wird M in Runde t generiert, dann nehmen wir für jedes $\mathcal{T}_\tau \in \Pi_t$ das Share

$$p_\tau(t - t_{\min}(\mathcal{T}_\tau) + 1)$$

in $\text{share}(M)$ auf. Somit entspricht der Wert $p_\tau(j)$ mit $1 \leq j \leq \Delta$ gerade der j-ten Runde in $\mathcal{T}_\tau$ und der t-ten Runde im Protokoll für den Seed S. Da $t \in \mathcal{T}_\tau$ für alle $\tau \in \{t-(\Delta-1), \ldots, t\}$, ergibt sich

$$\text{share}(M) = (p_{t-(\Delta-1)}(\Delta), p_{t-(\Delta-2)}(\Delta-1), \ldots, p_t(1)).$$

Veranschaulicht wird dies in Abbildung 4.1. Nach dem Beginn der Ausführung des Schemas benötigen wir für die ersten Zeitpunkte Werte für nicht existente Zeitabschnitte $\mathcal{T}_j$ mit $j < 0$. Für diese ersetzen wir wir die Shares durch den leeren Eintrag $\emptyset$. Folglich können wir aus solchen Einträgen kein Wissen über $\text{ID}(\mathcal{I}_i)$ gewinnen.

Hat die Sammelstelle d Nachrichten aus einem Zeitabschnitt $\mathcal{T}_\tau$ gespeichert, dann kennt sie d Stellen des Polynoms p_τ und kann dann $p_\tau(0) = \text{seed}^{(t_{\min}(\mathcal{T}_\tau))}(S)$ durch Lagrange-Interpolation berechnen. Anschließend kann sie für alle $t \geq t_{\min}(\mathcal{T}_\tau)$ die Werte $\text{seed}^{(t)}(S)$ und $\text{rand}^{(t)}(S)$ bestimmen. Entsprechend der benutzten Verschlüsselung der Nutzlast und des Fingerabdrucks der Nachrichten können wir alle Nachrichten (kritische und unkritische) des entsprechenden Benutzers identifizieren und entschlüsseln.

4.8 Analyse der Sicherheitseigenschaften des Schemas

Wir wollen nun untersuchen, ob die Sammelstelle unerlaubt Wissen aus den gespeicherten Nachrichten extrahieren kann. Dies bedeutet die Beantwortung der folgenden vier Fragen:

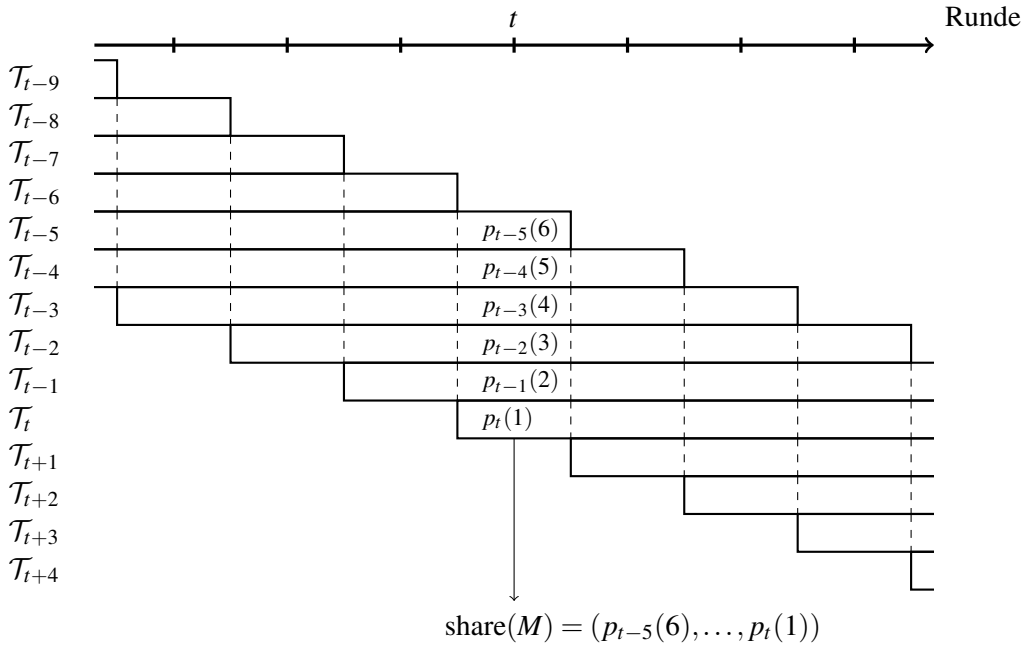


Abbildung 4.1: Struktur von $\text{share}(M)$ der Nachricht M aus Runde t für $\Delta = 6$

Zugriffssicherheit:
Kann die Sammelstelle Nachrichten eines Benutzers entschlüsseln, der in jedem Zeitabschnitt weniger als d kritische Aktionen ausgeführt hat?

Rückwärtssicherheit:
Kann die Sammelstelle Nachrichten eines Benutzers entschlüsseln, wenn sie nur für spätere Zeitabschnitte Entschlüsselungsschlüssel berechnen kann?

Anonymität:
Kann die Sammelstelle Nachrichten von Benutzern, die in jedem Zeitabschnitt weniger als d kritische Aktionen ausgeführt haben, einem Benutzer zuordnen?

Sicherheit vor falscher Verdächtigung:
Ordnet die Sammelstelle eine Nachricht einem falschen Benutzer zu?

Für unser Schema können wir diese Fragen mit „nein“ beantworten.

4.8.1 Zugriffssicherheit und Anonymität

Im Folgenden werden wir zeigen, dass unser Schema sowohl Zugriffssicherheit als auch Anonymität gewährleistet. In Theorem 4.8 haben wir bewiesen, dass die Sequenz $\{\text{rand}^{(\tau)}(S)\}_{i=0}^{\tau}$ pseudozufällig ist. Ferner können wir das folgende Lemma zeigen:

Lemma 4.10 *Sei $\ell' = |\text{subj}(M) \circ \mathcal{I}_i|$ die Länge der in der Nutzlast der Nachrichten zu verschlüsselnden Daten und seien τ sowie ℓ' polynomiell in ℓ. Sei die Abbildung* $\text{keys}^{[\tau]}\colon \{0,1\}^{\ell} \to \{0,1\}^{2(\tau+1)\ell'}$ *definiert durch*

$$\text{keys}^{[\tau]}(S) = G'(K_\tau) \circ G''(L_\tau) \circ G'(K_{\tau-1}) \circ G''(L_{\tau-1}) \circ \ldots \circ G'(K_0) \circ G''(L_0),$$

wobei $S \in \{0,1\}^{\ell}$ und $\text{rand}^{(t)}(S) = K_t \circ L_t$ für alle t. Dann ist $\text{keys}^{[\tau]}$ *ein Pseudozufallszahlengenerator.*

BEWEIS Da G', G'' sowie $\{\text{rand}^{(\tau)}(S)\}_{i=0}^{\tau}$ Pseudozufallszahlengeneratoren sind, τ und ℓ' polynomiell in ℓ sind und da

$$\{\text{rand}^{(\tau)}(S)\}_{i=0}^{\tau} = K_\tau \circ L_\tau \circ K_{\tau-1} \circ L_{\tau-1} \circ \ldots \circ K_0 \circ L_0,$$

können wir direkt aus Korollar 4.6 schließen, dass $\text{keys}^{[\tau]}$ ein Pseudozufallszahlengenerator ist. ∎

Für $\tau \in \mathbb{N}$ sei $\mathcal{B}_\tau$ die Menge der Identitäten der Benutzer, die in jedem der Zeitabschnitte $\mathcal{T}_0, \ldots, \mathcal{T}_\tau$ weniger als d kritische Aktionen ausgeführt haben. Sei $\mathcal{M}_\tau$ die Menge aller Nachrichten der Benutzer aus $\mathcal{B}_\tau$, die die Sammelstelle für die Zeitabschnitte $\mathcal{T}_0, \ldots, \mathcal{T}_\tau$ gespeichert hat. Für alle Nachrichten $M \in \mathcal{M}_\tau$ gilt folglich, dass $\text{time}(M) \in \mathcal{T}_j$. Wir bezeichnen die Menge aller möglichen Subjekte der Nachrichten von Benutzern aus $\mathcal{B}_\tau$ mit $\mathcal{SU}_\tau$. Mit $\mathcal{I}(M)$ bezeichnen wir die Identität des Benutzers der Nachricht M.

Lemma 4.11 *Für jeden probabilistischen Polynomialzeitalgorithmus $\mathcal{A}$, für alle $M \in \mathcal{M}_\tau$, für alle Polynome p und für alle genügend großen Werte des Sicherheitsparameters ℓ gilt:*

1. $$\Pr[\mathcal{A}(1^\ell, \text{load}(M)) = \mathcal{I}(M)] < \frac{1}{|\mathcal{B}_\tau|} + \frac{1}{p(\ell)},$$
2. $$\Pr[\mathcal{A}(1^\ell, \text{load}(M)) = \text{subj}(M)] < \frac{1}{|\mathcal{SU}_\tau|} + \frac{1}{p(\ell)}.$$

BEWEIS Die erste Ungleichung folgt analog zum Beweis der Sicherheit von Naors Bit-Commitment-Schemas [82]: R und $\text{fp}_{R,L}(\mathcal{I}_i)$ geben nicht mehr als vernachlässigbares Wissen über $\mathcal{I}_i$. Der String R wird zufällig gewählt und $\text{fp}_{R,L}(\mathcal{I}_i)$ ist nach Lemma 4.10 und analog zum Beweis von Lemma 4.4 pseudozufällig.

Da der Seed eines Benutzers zufällig gewählt ist, sind auch K und $G'(K)$ nach Theorem 4.8 und Lemma 4.10 pseudozufällig. Damit ist auch der String $G'(K) \oplus \text{subj}(M) = \text{enc}_K(\text{subj}(M) \circ \mathcal{I}_i)$ pseudozufällig.

Angenommen, wir könnten mit nicht vernachlässigbarer Wahrscheinlichkeit Wissen über $\text{subj}(M)$ aus $\text{load}(M)$ gewinnen. Dann gibt es einen probabilistischen Polynomialzeitalgorithmus $\mathcal{A}$ und ein Polynom q, so dass

$$\Pr[\mathcal{A}(1^\ell, \text{enc}_K(\text{subj}(M) \circ \mathcal{I}(M))) = \text{subj}(M)] > \frac{1}{|\mathcal{SU}_\tau|} + \frac{1}{q(\ell)},$$

wobei die Wahrscheinlichkeit über alle Subjekte und alle Seeds S, die K zugrunde liegen, zu nehmen ist. Sei U eine uniform verteilte Zufallsvariable über die Strings der Länge $|\text{subj}(M) \circ \mathcal{I}(M)|$. Dann gilt für alle Algorithmen $\mathcal{A}'$, dass

$$\Pr[\mathcal{A}'(1^\ell, U \oplus (\text{subj}(M) \circ \mathcal{I}(M))) = \text{subj}(M)] = \frac{1}{|\mathcal{SU}_\tau|}.$$

Für $\mathcal{A}' = \mathcal{A}$ erhalten wir dann, dass

$$\begin{aligned} &|\Pr[\mathcal{A}'(1^\ell, G'(K) \oplus (\text{subj}(M) \circ \mathcal{I}(M))) = 1] \\ &\quad - \Pr[\mathcal{A}'(1^\ell, U \oplus (\text{subj}(M) \circ \mathcal{I}(M))) = 1]| > \frac{1}{q(\ell)}. \end{aligned}$$

Also kann $\mathcal{A}$ mit nicht vernachlässigbarer Wahrscheinlichkeit sagen, ob der Pseudozufallszahlengenerator G' zur Verschlüsselung verwendet wurde. Damit können wir allerdings zwischen einem zufälligen und einem aus S generierten pseudozufälligen String unterscheiden. Dieses ist ein Widerspruch zur Eigenschaft des Pseudozufallszahlengenerators. Also ist $\text{enc}_K(\text{subj}(M) \circ \mathcal{I}_i)$ pseudozufällig und es gilt die zweite Ungleichung von Lemma 4.11. ∎

Korollar 4.12 *Für jeden probabilistischen Polynomialzeitalgorithmus $\mathcal{A}$, für alle $M \in \mathcal{M}_\tau$, für alle Polynome p und für alle genügend großen Werte ℓ gilt:*

1. $\Pr[\mathcal{A}(1^\ell, \text{load}(M)) = \text{subj}(M) \circ \mathcal{I}(M)] < \min\left(\frac{1}{|\mathcal{SU}_\tau|}, \frac{1}{|\mathcal{B}_\tau|}\right) + \frac{1}{p(\ell)}$.
2. *Falls $\mathcal{I}(M)$ und* subj(M) *unabhängig voneinander sind, gilt:*

$$\Pr[\mathcal{A}(1^\ell, \text{load}(M)) = \text{subj}(M) \circ \mathcal{I}(M)] < \frac{1}{|\mathcal{SU}_\tau|} \cdot \frac{1}{|\mathcal{B}_\tau|} + \frac{1}{p(\ell)}.$$

BEWEIS Nach dem Beweis von Lemma 4.11 ist load(M) pseudozufällig. Die zweite Ungleichung folgt aus der Unabhängigkeit von $\mathcal{I}(M)$ und subj(M). Allgemein gilt, dass

$$\Pr[\mathcal{A}(1^\ell, \text{load}(M)) = \text{subj}(M) \circ \mathcal{I}(M)] \leq \Pr[\mathcal{A}(1^\ell, \text{load}(M)) = \mathcal{I}(M)]$$

und $$\Pr[\mathcal{A}(1^\ell, \text{load}(M)) = \text{subj}(M) \circ \mathcal{I}(M)] \leq \Pr[\mathcal{A}(1^\ell, \text{load}(M)) = \text{subj}(M)].$$

Nach Lemma 4.11 folgt somit auch die erste Ungleichung. ∎

Wenn wir kein Wissen über die Entschlüsselungsschlüssel K und L haben, dann ist also die Chance, dass wir $\mathcal{I}_i$ und subj(M) aus load(M) für eine Nachricht M bestimmen können, vernachlässigbar.

Nun wollen wir untersuchen, ob die Gesamtheit der gespeicherten Nachrichten nicht zur Identifikation und Entschlüsselung von Nachrichten beiträgt.

Lemma 4.13 *Sei $G\colon \{0,1\}^{\ell_0} \to \{0,1\}^{\ell_1}$ ein Pseudozufallszahlengenerator. Dann gilt für alle $Z \in \{0,1\}^{\ell_1}$, für alle probabilistischen Polynomialzeitalgorithmen $\mathcal{A}$, alle Polynome p und alle genügend großen Werte ℓ_0, dass*

$$|\Pr[\mathcal{A}(1^{\ell_0}, G(U_{\ell_0}) \oplus Z) = 1] - \Pr[\mathcal{A}(1^{\ell_0}, U_{\ell_1}) = 1]| < \frac{1}{p(\ell_0)}.$$

BEWEIS Angenommen es gibt ein $Z' \in \{0,1\}^{\ell_1}$, einen probabilistischen Polynomialzeitalgorithmus $\mathcal{D}$ und ein Polynom q, so dass für unendlich viele ℓ_0

$$|\Pr[\mathcal{D}(1^{\ell_0}, G(U_{\ell_0}) \oplus Z') = 1] - \Pr[\mathcal{A}(1^{\ell_0}, U_{\ell_1}) = 1]| \geq \frac{1}{q(\ell_0)}$$

gilt. Für $W \in \{0,1\}^{\ell_1}$ sei der probabilistische Algorithmus $\mathcal{D}_W$ auf Eingabe $\alpha \in \{0,1\}^{\ell_1}$ wie folgt definiert:

$$\mathcal{D}_W(1^{\ell_0}, \alpha) = \mathcal{D}(1^{\ell_0}, \alpha \oplus W).$$

$\mathcal{D}_W$ ist ein probabilistischer Polynomialzeitalgorithmus, da ℓ_1 polynomiell in ℓ_0 ist und $\mathcal{D}$ ein probabilistischer Polynomialzeitalgorithmus ist. Es folgt, dass

$$\begin{aligned}
&|\Pr[\mathcal{D}_{Z'}(1^{\ell_0}, G(U_{\ell_0})) = 1] - \Pr[\mathcal{D}_{Z'}(1^{\ell_0}, U_{\ell_1}) = 1]| \\
&= |\Pr[\mathcal{D}(1^{\ell_0}, G(U_{\ell_0}) \oplus Z') = 1] - \Pr[\mathcal{A}(1^{\ell_0}, U_{\ell_1}) = 1]| \\
&\geq \frac{1}{q(\ell_0)}.
\end{aligned}$$

Dies ist ein Widerspruch dazu, dass G ein Pseudozufallszahlengenerator ist. ∎

Verschlüsseln wir folglich einen String Z durch XOR mit einem pseudozufälligen String, ist die Verschlüsselung sicher. Dies gilt allerdings nur, wenn wir den pseudozufälligen String zur Verschlüsselung von einem einzigen Plaintext benutzen.

Sei $\mathcal{AM}_t$ die Menge aller Nachrichten, die die Sammelstelle bis zur Runde $t \in \mathbb{N}$ gespeichert hat. Wir sagen, *ein Polynomialzeitalgorithmus $\mathcal{A}$ hat beliebigen Zugriff auf $\mathcal{AM}_\tau$*, notiert mit $\mathcal{A}^{\mathcal{AM}_\tau}$, wenn er auf Bits der Elemente von $\mathcal{AM}_\tau$ beliebig zugreifen kann. Die Anzahl dieser Bits ist nur durch seine Laufzeit beschränkt, das heißt, er kann auf höchstens polynomiell in der Eingabelänge viele Bits aus $\mathcal{AM}_\tau$ zugreifen.

Theorem 4.14 *Sei $\tau \in \mathbb{N}$. Für alle probabilistischen Polynomialzeitalgorithmen $\mathcal{A}^{\mathcal{AM}_\tau}$, für alle $M \in \mathcal{M}_\tau$, für alle Polynome p und alle genügend großen Werte des Sicherheitsparameters ℓ gilt:*

$$\Pr[\mathcal{A}^{\mathcal{AM}_\tau}(1^\ell, M) = \mathrm{subj}(M) \circ \mathcal{I}(M)] < \min\left(\frac{1}{|\mathcal{SU}_\tau|}, \frac{1}{|\mathcal{B}_\tau|}\right) + \frac{1}{p(\ell)}.$$

BEWEIS Die Zufallsvariable O beschreibe die Menge der Nachrichten aus $\mathcal{AM}_\tau$, auf deren Bits $\mathcal{A}^{\mathcal{AM}_\tau}$ während eines Berechnungslaufs auf Eingabe $(1^\ell, M)$ zugreift. Da die Laufzeit von $\mathcal{A}^{\mathcal{AM}_\tau}$ polynomiell durch ℓ beschränkt ist, ist $|O|$ polynomiell in ℓ. Nach Definition von $\mathcal{M}_\tau$ gibt es in $\mathcal{M}_\tau$ und damit in O für jeden Zeitabschnitt höchstens $d-1$ kritische Nachrichten eines Benutzers in $\mathcal{B}_\tau$. Folglich gibt es in O für jeden Zeitabschnitt höchstens $d-1$ kritische Nachrichten mit derselben ID. Da für jeden Benutzer sein Seed unabhängig von den anderen Benutzern gewählt wird, sind auch die Strings $\mathrm{seed}^{(\cdot)}(\cdot)$ und $\mathrm{rand}^{(\cdot)}(\cdot)$ dieses Benutzers unabhängig von den Seeds und den entsprechenden Strings der anderen Benutzer. Damit sind die Shares der Nachrichten eines Benutzers unabhängig von den Shares, den Seeds und den entsprechenden Strings $\mathrm{seed}^{(\cdot)}(\cdot)$ und $\mathrm{rand}^{(\cdot)}(\cdot)$ der anderen Benutzer. Durch die perfekten Sicherheit von Shamirs Secret-Sharing-Schema im Fall von kritischen Nachrichten und durch den leeren share-Eintrag im Fall von unkritischen Nachrichten geben die Werte $\mathrm{share}(M')$ aller Nachrichten $M' \in \mathcal{AM}_\tau$ keine Information über andere Werte. Wir können sie als unabhängige und uniforme Zufallsvariablen über den Strings entsprechender Länge betrachten. Da zudem $\mathcal{A}^{\mathcal{AM}_\tau}$ keine weitere Information über das Verhalten der Benutzer enthält, sind die Werte time und ID aller Nachrichten unabhängig von $\mathrm{subj}(M) \circ \mathcal{I}(M)$ und von $\mathrm{keys}^{[\tau]}(S)$. Dabei sei S der Seed des Benutzers, zu dem die Nachricht M gehört. Folglich kann $\mathcal{A}^{\mathcal{AM}_\tau}$ nur auf die Werte $\mathrm{load}(M')$ von Nachrichten aus O zugreifen. Seien $S_1, \ldots, S_o$ die Seeds für alle Benutzer aus $\mathcal{B}_\tau$ mit Nachrichten in O. Dann ist $o \leq |O|$ polynomiell in ℓ. Aus Lemma 4.10, Korollar 4.6 und der zufälligen Generierung der Seeds können wir folgern, dass der String

$$\mathrm{keys}^{[\tau]}(S_1) \circ \ldots \circ \mathrm{keys}^{[\tau]}(S_o)$$

pseudozufällig ist. Weil für jeden Benutzer jeder Substring von $\mathrm{rand}^{(\cdot)}(\cdot)$ als Schlüssel höchstens in einem Fingerabdruck oder einer Verschlüsselung benutzt wird, ist nach Lemma 4.13 auch

$$\mathrm{load}(M_{a_1}) \circ \ldots \circ \mathrm{load}(M_{a_{|O_{\mathcal{B}_\tau}|}})$$

pseudozufällig, wobei $O_{\mathcal{B}_\tau} = O \cap \mathcal{M}_\tau = \{M_{a_1}, \ldots, M_{a_{|O_{\mathcal{B}_\tau}|}}\}$. Somit folgt die Behauptung des Theorems analog zu Korollar 4.12. ∎

Folglich hat die Sammelstelle nur einen vernachlässigbaren Vorteil, um Wissen über den Benutzer und den Inhalt einer beliebigen Nachricht aus allen gespeicherten Nachrichten zu gewinnen. Damit sind die Zugriffssicherheit und die Anonymität des Schemas gewährleistet.

4.8.2 Rückwärtssicherheit und Sicherheit vor falscher Verdächtigung

Um die Nutzlast einer Nachricht zu entschlüsseln, ist es durch die Zugriffssicherheit des Schemas notwendig, dass die Sammelstelle für einen Benutzer mindestens d kritische Nachrichten innerhalb eines Zeitabschnitts erhalten hat.

Nach Theorem 4.14 kann die Sammelstelle für alle Runden τ aus den rekonstruierten Schlüsseln von identifizierten Benutzern kein Wissen über Identität und Inhalt von Nachrichten von Benutzern gewinnen, für die bis Runde τ in jedem Zeitabschnitt weniger als d kritische Nachrichten gespeichert wurden. Gibt es für einen Benutzer einen Zeitabschnitt $\mathcal{T}_\tau$, innerhalb dessen mindestens d kritische Nachrichten dieses Benutzers gespeichert wurden, kann die Sammelstelle aus den share-Einträgen dieser Nachrichten die Werte $\text{seed}^{(t_{\min}(\mathcal{T}_\tau))}(S)$ und $\mathcal{I}_i$ für diesen Benutzer durch Lagrange-Interpolation bestimmen. Damit kann die Sammelstelle für alle $t \geq t_{\min}(\mathcal{T}_\tau)$ die Werte $\text{rand}^{(t)}(S)$ berechnen und erhält so alle Schlüssel zur Identifikation und Entschlüsselung der Nachrichten dieses Benutzers, die in der Runde $t_{\min}(\mathcal{T}_\tau)$ oder später empfangen und gespeichert wurden.

Ein wesentliches Merkmal von pseudozufälligen Strings ist die Unvorhersagbarkeit des nächsten Bits:

Lemma 4.15 *$G\colon \{0,1\}^{\ell_0} \to \{0,1\}^{\ell_1}$ ist genau dann ein Pseudozufallszahlengenerator, wenn für alle $1 \leq k < \ell_1$, für jeden probabilistischen Polynomialzeitalgorithmus $\mathcal{A}$, für alle Polynome p und für hinreichend große ℓ_0 gilt:*

$$\Pr[\mathcal{A}(1^{\ell_0}, G(U_{\ell_0})_{1,\ldots,k}) = G(U_{\ell_0})_{k+1}] < \frac{1}{2} + \frac{1}{p(\ell_0)}.$$

Diese Beobachtung geht auf Yao [111] zurück und wird für einen allgemeineren Fall von Goldreich et al. in [40] bewiesen. Die Unvorhersagbarkeit gilt nicht nur für das nächste Bit, sondern auch für beliebige Teilstrings:

Lemma 4.16 *Sei $G\colon \{0,1\}^{\ell_0} \to \{0,1\}^{\ell_1}$ ein Pseudozufallszahlengenerator. Dann gilt für alle Teilmengen $J \subset \{1,\ldots,\ell_1\}$ und $I \subseteq \overline{J} = \{1,\ldots,\ell_1\} \setminus J$, für alle probabilistischen Polynomialzeitalgorithmen $\mathcal{A}$, für alle Polynome p und für hinreichend große ℓ_0:*

$$\Pr[\mathcal{A}(1^{\ell_0}, G(U_{\ell_0})_J) = G(U_{\ell_0})_I] < \frac{1}{2^{|I|}} + \frac{1}{p(\ell_0)},$$

wobei die Zufallsvariable U_{ℓ_0} uniform über $\{0,1\}^{\ell_0}$ verteilt ist.

BEWEIS Wir nehmen an, dass es einen probabilistischen Polynomialzeitalgorithmus $\mathcal{A}'$, Teilmengen $J \subset \{1,\ldots,\ell_1\}$ und $I \subseteq \overline{J}$ sowie ein Polynom q gibt, so dass für unendlich viele ℓ_0 gilt:

$$\Pr[\mathcal{A}'(1^{\ell_0}, G(U_{\ell_0})_J = G(U_{\ell_0})_I] \geq \frac{1}{2^{|I|}} + \frac{1}{q(\ell_0)}.$$

Wir betrachten nun den folgenden probabilistischen Polynomialzeitalgorithmus $\mathcal{D}$ auf Eingabe $(1^{\ell_0}, \alpha)$ mit $\alpha \in \{0,1\}^{\ell_1}$, der den Algorithmus $\mathcal{A}'$ als Subroutine verwendet:

$$\mathcal{D}(1^{\ell_0}, \alpha) = \begin{cases} 1, & \text{falls } \mathcal{A}'(1^{\ell_0}, \alpha_J) = \alpha_I \text{ und} \\ \beta & \text{sonst, wobei } \beta \text{ uniform aus } \{0,1\} \text{ gewählt wird.} \end{cases}$$

Für die uniform über $\{0,1\}^{\ell_1}$ verteilte Zufallsvariable U_{ℓ_1} gilt, dass

$$\begin{aligned}
\Pr[\mathcal{D}(1^{\ell_0}, U_{\ell_1}) = 1] &= \sum_{x \in \{0,1\}^{|I|}} \left(1 \cdot \Pr[\mathcal{A}'(1^{\ell_0}, (U_{\ell_1})_J) = x] + \frac{1}{2} \cdot \Pr[\mathcal{A}'(1^{\ell_0}, (U_{\ell_1})_J) \neq x] \right) \\
&\qquad \cdot \Pr[(U_{\ell_1})_I = x] \\
&= \sum_{x \in \{0,1\}^{|I|}} \left(1 \cdot \Pr[\mathcal{A}'(1^{\ell_0}, (U_{\ell_1})_J) = x] + \frac{1}{2} \cdot (1 - \Pr[\mathcal{A}'(1^{\ell_0}, (U_{\ell_1})_J) = x]) \right) \\
&\qquad \cdot \Pr[(U_{\ell_1})_I = x] \\
&= \sum_{x \in \{0,1\}^{|I|}} \frac{1}{2}(1 + \Pr[\mathcal{A}'(1^{\ell_0}, (U_{\ell_1})_J) = x]) \\
&\qquad \cdot 2^{-|I|} \\
&= \frac{1}{2} + 2^{-|I|-1} \cdot \sum_{x \in \{0,1\}^{|I|}} \Pr[\mathcal{A}'(1^{\ell_0}, (U_{\ell_1})_J) = x] \\
&= \frac{1}{2} + 2^{-|I|-1} \cdot 1.
\end{aligned}$$

Für die Eingabe eines pseudozufälligen Strings $G(U_{\ell_0})$ gilt, dass

$$\begin{aligned}
\Pr[\mathcal{D}(1^{\ell_0}, G(U_{\ell_0})) = 1] &= \frac{1}{2} \Pr[\mathcal{A}'(1^{\ell_0}, G(U_{\ell_0})_J) \neq G(U_{\ell_0})_I] \\
&\quad + 1 \cdot \Pr[\mathcal{A}'(1^{\ell_0}, G(U_{\ell_0})_J) = G(U_{\ell_0})_I] \\
&\geq \frac{1}{2} \cdot \left(1 - \left(2^{-|I|} + \frac{1}{q(\ell_0)} \right) \right) + 1 \cdot \left(2^{-|I|} + \frac{1}{q(\ell_0)} \right) \\
&= \frac{1}{2} + \frac{1}{2} \cdot \left(2^{-|I|} + \frac{1}{q(\ell_0)} \right).
\end{aligned}$$

Folglich ist $\mathcal{D}$ ein Unterscheider für G mit

$$\begin{aligned}
&|\Pr[\mathcal{D}(1^{\ell_0}, G(U_{\ell_0})) = 1] - \Pr[\mathcal{D}(1^{\ell_0}, U_{\ell_1}) = 1]| \\
&= \left| \frac{1}{2} + 2^{-|I|-1} - \left(\frac{1}{2} + \frac{1}{2} \cdot \left(2^{-|I|} + \frac{1}{q(\ell_0)} \right) \right) \right| \\
&= \frac{1}{2q(\ell_0)}.
\end{aligned}$$

Dieses ist ein Widerspruch zu der Voraussetzung, dass G ein Pseudozufallszahlengenerator ist. ∎

Theorem 4.17 *Sei $\tau \in \mathbb{N}$ polynomiell in dem Sicherheitsparameter ℓ. Sei $\mathcal{T}_{\tau^*}$ der erste Zeitabschnitt für einen Benutzer mit $\tau^* \leq \tau$, in dem die Sammelstelle mindestens d Nachrichten dieses Benutzers empfangen hat. Dann gilt für alle Nachrichten M dieses Benutzers mit* $\mathrm{time}(M) < t_{\min}(\mathcal{T}_{\tau^*})$*, für alle probabilistischen Polynomialzeitalgorithmen $\mathcal{A}^{\mathcal{AM}_\tau}$, für alle Polynome p und genügend große Werte des Sicherheitsparameters ℓ, dass*

$$\Pr[\mathcal{A}^{\mathcal{AM}_\tau}(1^\ell, M) = \mathrm{subj}(M)] < \frac{1}{|\mathcal{SU}_t|} + \frac{1}{p(\ell)}.$$

BEWEIS Analog zum Beweis von Theorem 4.14 können wir folgern, dass M und somit die Verschlüsselung von load(M) unabhängig von den Nachrichten aus $\mathcal{AM}_\tau$ anderer Benutzer sind. Da die Einträge time und ID unabhängig von subj(M) sind, die Sammelstelle in allen Zeitabschnitten $\mathcal{T}_{\tau'}$ mit $\tau' < \tau^*$ höchstens $d-1$ kritische Nachrichten des Benutzers empfangen und gespeichert hat und Shamirs Secret-Sharing-Schema perfekt sicher ist, kann die Sammelstelle nur Information über subj(M) aus den Nutzlasten load(M') der Nachrichten M' desselben Benutzers mit time$(M') \geq t_{\min}(\mathcal{T}_{\tau^*})$ gewinnen. Da wir annehmen, dass subj(M') unabhängig von subj(M) ist, kann die Sammelstelle nur aus den Shares der Nachrichten des entsprechenden Benutzers ab Runde $t_{\min}(\mathcal{T}_{\tau^*})$ und damit aus $\text{seed}^{(t_{\min}(\mathcal{T}_{\tau^*}))}(S)$ sowie aus den daraus erzeugbaren Strings Wissen über subj(M) gewinnen. Aus Lemma 4.1 und den Beweisen von Theorem 4.8 und Lemma 4.10 folgt, dass für alle Zeitpunkte $t \geq 1$, die polynomiell in ℓ sind,

$$\text{seed}^{(t)}(S) = \widehat{G}(\text{seed}^{(t-1)}(S))_{1,\ldots,\ell} = (\tilde{G}^{[0]} \circ \ldots \circ \tilde{G}^{[t-1]}(S))_{1,\ldots,\ell}$$

und

$$\text{seed}^{(t)}(S) \circ \text{keys}^{[t-1]}(S)$$

pseudozufällig sind. Nach Lemma 4.16 kann $\text{keys}^{[t-1]}(S)$ oder ein Teilstring von $\text{keys}^{[t-1]}(S)$ höchstens mit vernachlässigbarer Wahrscheinlichkeit vorhergesagt werden. Somit folgt analog zu dem Beweis von Theorem 4.14 die Aussage. ∎

Die Sammelstelle erhält somit kein Wissen über den Inhalt der Nachrichten eines Benutzers, die vor der Runde $t_{\min}(\mathcal{T}_{\tau^*})$ generiert wurden. Damit ist das Schema rückwärtssicher. Nach den Theoremen 4.14 und 4.17 sowie nach Lemma 4.9 lässt sich aus den Nachrichten eines Benutzers kein Wissen über die Nachrichten anderer Benutzer gewinnen. Zudem ist nach Lemma 4.9 die Wahrscheinlichkeit einer Kollision vernachlässigbar klein. Folglich schützt das Schema vor falscher Verdächtigung. Damit können wir schließen:

Theorem 4.18 *Das Schwellwert-Schema zur privaten Vorratsdatenspeicherung erfüllt die Anforderungen der Zugriffssicherheit, Rückwärtssicherheit, Anonymität und Sicherheit vor falscher Verdächtigung.*

4.9 Implementierung

Um Laufzeitmessungen des vorgestellten Schwellwert-Schemas durchzuführen, verwenden wir den von Stockhusen implementierten JAVA-Prototypen des Systems [105].

Grundsätzlich erwarten wir einen linearen Anstieg der Laufzeit zur Generierung einer Nachricht in Abhängigkeit der Seedlänge ℓ. Da eine kritische Nachricht Shares enthält, erwarten wir für eine kritische Nachricht außerdem einen linearen Anstieg der Laufzeit zu ihrer Generierung im Schwellwert d und in der Länge Δ der Zeitabschnitte.

Falls nicht anders angegeben, verwenden wir standardmäßig folgende Werte:

Seedlänge	$\ell = 1.000$
Schwellwert	$d = 20$
Länge der Zeitabschnitte	$\Delta = 500$
Länge der Identitätsstrings	$\lvert\mathcal{I}_i\rvert = 100$
Blockgröße des Fingerabdrucks	$m = 50$
Größe der Subjekte	$\lvert\text{subj}(\cdot)\rvert$ =1kB

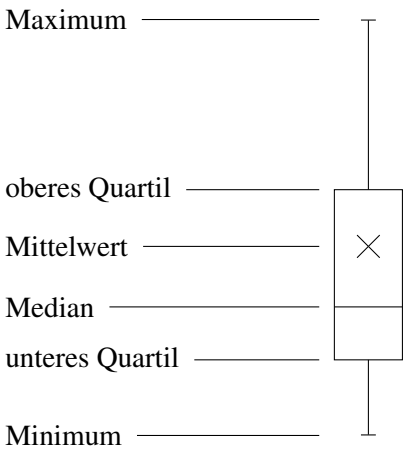


Abbildung 4.2: Aufbau eines Box-Whisker-Plots zur Darstellung der Verteilung eines Datensatzes.

Zur Darstellung eines Datensatzes (Stichprobe) nutzen wir Box-Whisker-Plots (siehe Abbildung 4.2). Die Kurven in den weiteren Abbildungen verbinden die Mediane der verschiedenen Stichproben. Die Messungen wurden auf einem Rechner mit einer 2,66 GHz Dual-Core CPU mit 4 GB Arbeitsspeicher durchgeführt. Die Generierung der Nachrichten erfolgte seriell. Um Messungenauigkeiten entgegenzuwirken, wurde der JAVA-Garbage-Collector während der Zeitnahme nach Möglichkeit unterdrückt.

Für eine Stichprobe wird das System für 100.000 Runden simuliert. Insgesamt werden dabei 1.000 kritische beziehungsweise unkritische Nachrichten generiert. Wir betrachten im Folgenden den Zeitabstand zwischen der Generierung von aufeinander folgenden kritischen Nachrichten. Wenn die Abstände zwischen den Generierungszeitpunkten der kritischen Nachrichten stark unterscheiden, variieren die Generierungszeiten ebenfalls stark. In Abbildung 4.3 ist die Abhängigkeit der Laufzeit vom Abstand zu erkennen. Bis zu einem Abstand von $500 = \Delta$ Runden müssen proportional zum Abstand viele neue Polynome generiert werden. Für Abstände, die größer als Δ Runden sind, steigt die Laufzeit nur sehr langsam an. Der geringe Anstieg erklärt sich dadurch, dass die Anzahl der neu zu bestimmenden Polynome durch Δ gedeckelt ist und dass die Schlüssel über die iterierte Anwendung eines Pseudozufallszahlengenerators konstruiert werden müssen.

Da für eine unkritische Nachricht keine Polynome und damit Shares berechnet werden müssen, liegt ihre Generierungszeit unter der Generierungszeit einer kritischen Nachricht. Der lineare Anstieg der Laufzeit erklärt sich durch die Schlüsselberechnung. Damit die Abhängigkeit der Laufzeit vom Abstand der Zeitpunkte für verschiedene Nachrichten nicht die weiteren Untersuchungen stört, werden die Nachrichten im Folgenden in einem regelmäßigen Abstand von 100 Runden generiert.

Die Laufzeit zur Verschlüsselung für eine Nachricht hängt hauptsächlich von der Länge des Sicherheitsparameters ℓ, also der Seedlänge, ab. In Abbildung 4.4 ist der Einfluss der Seedlänge auf die Generierung der Nachrichten dargestellt. Da bei kritischen Nachrichten die Seedlänge in die Polynom- und Shareberechnung eingeht, ist bei kritischen Nachrichten der Anstieg in Abhängigkeit von der Seedlänge größer als bei unkritischen Nachrichten. Die Seedlänge bestimmt dabei die Größe der Koeffizienten der Polynome. Wie erwartet wächst die Laufzeit linear in der Seedlänge.

Die Zeit zur Berechnung der Polynome und damit der Shares für die kritischen Nach-

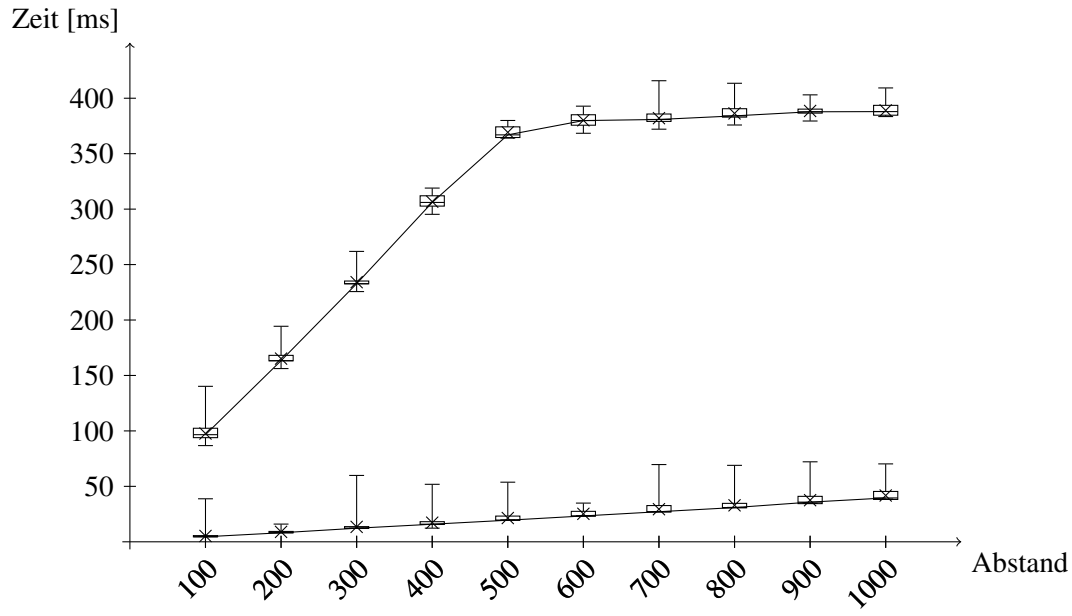


Abbildung 4.3: Laufzeit [ms] zur Generierung einer kritischen Nachricht (obere Kurve) und einer unkritischen Nachricht (untere Kurve) in Abhängigkeit des Abstandes der Generierungszeitpunkte von aufeinander folgenden Nachrichten

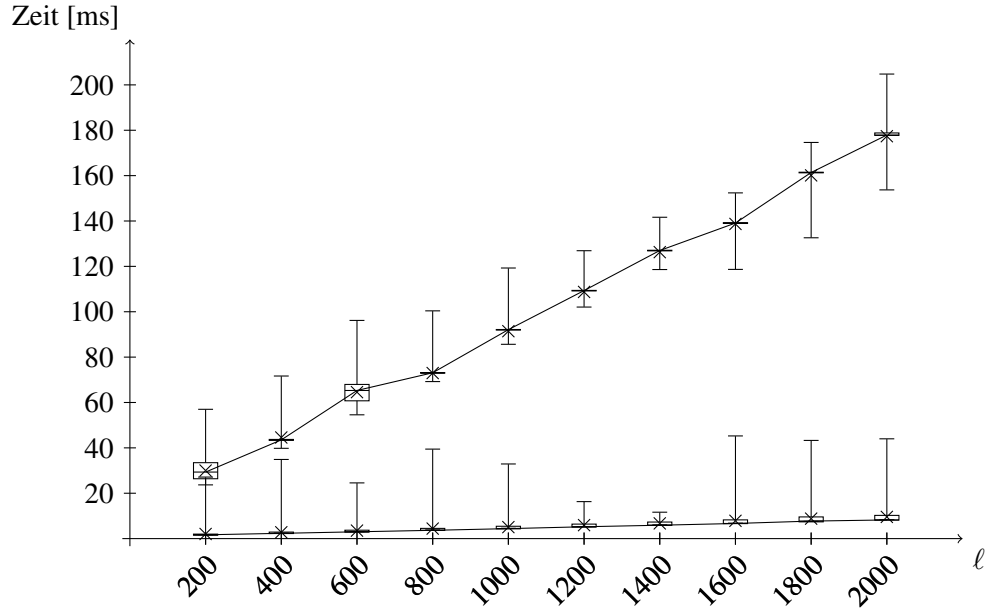


Abbildung 4.4: durchschnittliche Laufzeit [ms] zur Generierung einer kritischen Nachricht (obere Kurve) und einer unkritischen Nachricht (untere Kurve) in Abhängigkeit der Seedlänge ℓ

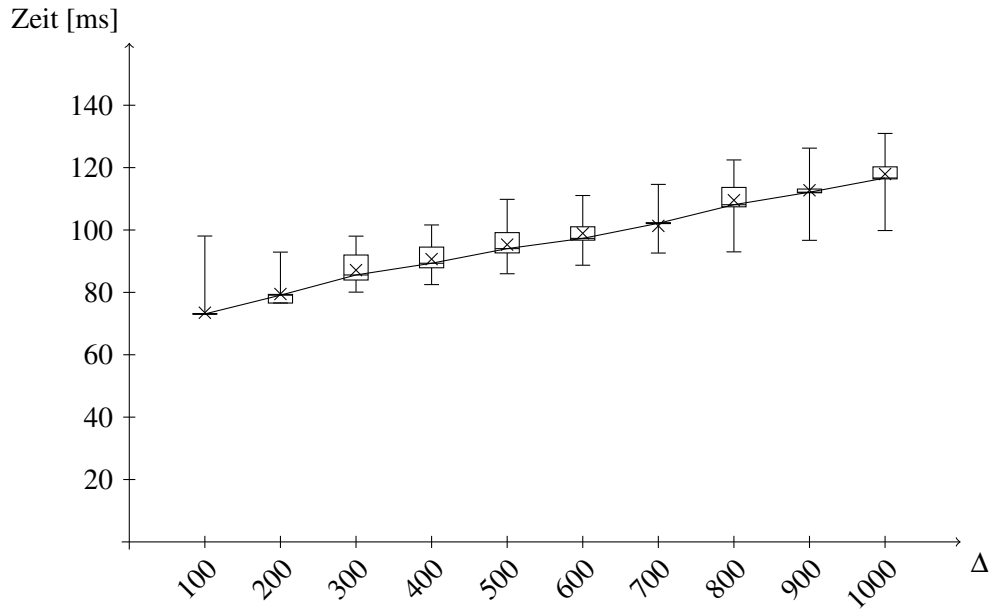


Abbildung 4.5: durchschnittliche Laufzeit [ms] zur Generierung einer kritischen Nachricht in Abhängigkeit der Zeitabschnittslänge Δ

richten ist des Weiteren von der Länge Δ des Vorhaltezeitraums und dem Schwellwert d abhängig. Für diese Parameter erwarten wir ebenfalls einen linearen Anstieg der Laufzeit. Dieses Verhalten ist in den Abbildungen 4.5 und 4.6 deutlich zu erkennen.

Die Generierungszeit für kritische Nachrichten in dieser Implementierung liegt für moderate Werte der Parameter im Bereich von 100 Millisekunden und für größere Werte unter einer Sekunde. Für unkritische Nachrichten ist die Laufzeit deutlich geringer. Damit können wir auch experimentell anhand dieser Beispielimplementierung feststellen, dass das vorgestellte Schwellwert-Schema zur Vorratsdatenspeicherung effizient ist und praktischen Einsatzcharakter hat. Bei der vorliegenden JAVA-Implementierung stand die Performanz des Systems nicht im Vordergrund. Durch die Wahl einer anderen Programmiersprache und eine Optimierung der Implementierung bezüglich der Laufzeit ist eine Verbesserung der Ergebnisse zu erwarten.

4.10 Diskussion des Schemas

In diesem Abschnitt gehen wir auf zwei Punkte ein, die zu einer Erweiterung des Ansatzes führen und die bei einem Einsatz des Schemas wichtig sein können. Zunächst betrachten wir den Einfluss von Zusatzinformation der Sammelstelle über das Benutzerverhalten. Anschließend diskutieren wir die Rolle des Providers im Schema und den Einsatz einer vertrauenswürdigen Partei.

4.10.1 Externe Informationen der Sammelstelle

In den Analysen zur Sicherheit des Schemas in Abschnitt 4.8 sind wir davon ausgegangen, dass die Sammelstelle aus den gespeicherten Daten Wissen über die Benutzer gewinnen kann und dass sie keine weitere Information über die Benutzer aus externer Quelle erhält.

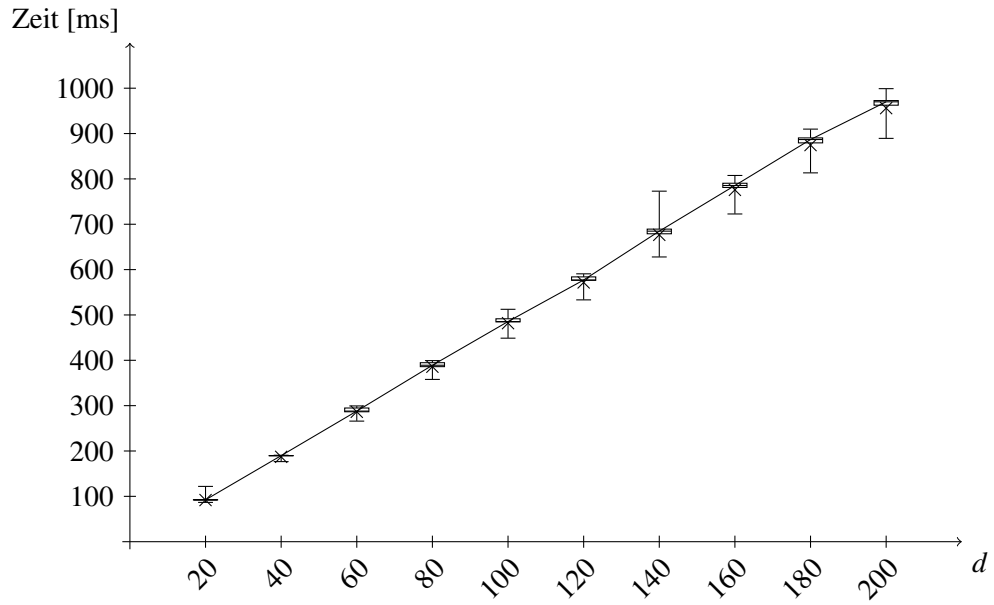


Abbildung 4.6: durchschnittliche Laufzeit [ms] zur Generierung einer kritischen Nachricht in Abhängigkeit des Schwellwertes d

Bei ehrlichen Providern kann die Sammelstelle kein Wissen über die generierten Werte gewinnen und die Verschlüsselung der Nachrichten nicht brechen. Jedoch wäre es denkbar, dass die Sammelstelle aus anderer Quelle lernt, wie das typische Verhalten eines bestimmten Nutzers ist. Beispielsweise könnte die Sammelstelle wissen, dass ein Benutzer nur abends nach der Arbeit den Dienst seines Providers nutzt. Ist nun die zeitliche Auflösung des Schemas so genau, dass die Sammelstelle zum Beispiel morgens und abends unterscheiden kann, dann entsprechen die morgens generierten Nachrichten nicht diesem Benutzer. Die Nachrichten sind in diesem Fall nicht mehr anonym bezüglich aller Benutzer, sondern nur noch ununterscheidbar unter allen Benutzern, die ein vergleichbares typisches Verhalten zeigen.

Müssen wir weitere Informationsquellen der Sammelstelle befürchten, so kann der Wissensgewinn für die Sammelstelle begrenzt werden, indem wir eine gröbere zeitliche Auflösung des Schemas wählen. Dadurch kann es notwendig werden, dass ein Benutzer mehrere Aktionen in einer Runde durchführt. Technisch gesehen, können wir den Versatz der Zeitabschnitte $\mathcal{T}_i$ von 1 auf einen Wert $\theta > 1$ vergrößern, das heißt, $t_{\min}(\mathcal{T}_{i+1}) = t_{\min}(\mathcal{T}_i) + \theta$. Wir interpretieren dann einen Block von θ Schritten als eine Runde. Somit kann ein Benutzer θ Aktionen in einer Runde ausführen. In unserem Beispiel kann der Provider für eine kritische Aktion eines Benutzers dann ein beliebiges und unbenutztes der θ Shares verwenden, um die Nachricht zu generieren. Am Ende der Runde kann der Provider alle Nachrichten der Runde gemeinsam an die Sammelstelle schicken. Folglich kann die Sammelstelle keinen Unterschied mehr zwischen unterschiedlichen Tagesabschnitten feststellen und damit das Wissen über das typische Verhalten dieses Benutzers nicht nutzen, um Daten des Benutzers zu erkennen.

Diese Technik zur Erhöhung des Versatzes zwischen den Zeitabschnitten können wir ebenfalls zur Effizienzsteigerung des Schemas einsetzen. Bei Länge Δ der Zeitabschnitte und einem Versatz von θ Schritten, verringert sich durch die Vergrößerung der Runden die

Anzahl der Runden pro Zeitabschnitt von Δ Runden auf $\frac{\Delta}{\theta}$. Somit werden nur $\frac{\Delta}{\theta}$ Shares je kritischer Nachricht benötigt und es verringern sich die Laufzeiten zur Erstellung der Shares einer Nachricht entsprechend um den Faktor $\frac{1}{\theta}$.

Wir müssen beachten, dass wir die Runden nur soweit vergrößern können, wie wir mehrere Nachrichten pro Runde und damit eine gröbere zeitliche Auflösung akzeptieren können. Reicht die Vergröberung der zeitlichen Auflösung nicht aus, um die Anonymität bei externem Wissen der Sammelstelle sicherzustellen, muss im Einzelfall der Anwendung entschieden werden, ob der Wissensgewinn der Sammelstelle tolerierbar ist. Ist das nicht der Fall, müssen die Speicherung der Daten bei einer Trusted Third Party in Betracht ziehen. Diese löscht Daten ordnungsgemäß und stellt diese bei möglicher Öffnung der Sammelstelle zur Verfügung.

4.10.2 Rolle einer vertrauenswürdigen Partei

Wir gehen in dieser Analyse des Schemas davon aus, dass der Provider eine vertrauenswürdige Partei ist. Bei der Speicherung von Kommunikationsdaten, der Verkehrsüberwachung und der Geldwäsche lässt sich die Arbeit der Provider kontrollieren. Die Konsequenzen eines Missbrauchs für den Provider können so gewählt werden, dass sie die Vorteile einer Verfehlung deutlich überwiegen. Betrachten wir ein Szenario, in dem der Provider, in seiner Rolle als Schnittstelle zwischen Benutzer und Sammelstelle, Daten von kritischen Nachrichten fälschen kann, um dadurch die Identifikation eines Benutzers zu verhindern. Dann können wir bei der Generierung von kritischen Nachrichten den Provider zusammen mit dem Benutzer als Teil einer nicht vertrauenswürdigen Partei betrachten. Ein Beispiel für dieses Szenario wäre, dass die Sammelstelle den Benutzern den Zugang zu einem Dienst gewährt, solange diese nicht zu häufig auf den Dienst zugreifen. Dabei soll die Gewährung des Zugangs anonym erfolgten. Dazu könnte die Sammelstelle dem Benutzer ein bisher unbenutztes Token blind unterschreiben und im Gegenzug eine kritische Nachricht vom Benutzer erhalten. Das unterschriebene Token dient somit als Einmal-Eintrittskarte zum Dienst. Der Dienst kann durch Überprüfung der Tokens eine Doppelbenutzung feststellen. Allerdings können Dienst und Sammelstelle weder einzeln noch gemeinsam feststellen, wann der Zugang beantragt wurde. Können wir sicherstellen, dass die Shares einer kritischen Nachricht der korrekten Identität entsprechen, dann kann ein im obigen Sinne nicht vertrauenswürdiger Provider die Identität nicht fälschen.

Um in diesem Fall die Korrektheit des Schwellwertmechanismus unseres Schemas sicherzustellen, können wir die Identitäten und die Shares durch eine Trusted Third Party überprüfen und signieren lassen. Diese Trusted Third Party muss sich nicht an der Schnittstelle zur Sammelstelle befinden. Die Signatur kann an einem beliebigen Zeitpunkt vor der Generierung einer kritischen Nachricht stattfinden. Die Sammelstelle kann sich anhand der Korrektheit einer Signatur vergewissern, dass die Shares und die gegebenenfalls aus den Shares rekonstruierte Identität korrekt sind. Lassen es die Parameter des Schemas zu, dass die Trusted Third Party für alle Benutzer und für alle Shares eines Zeitabschnitts gleichzeitig Unterschriften leisten kann sowie dass die Benutzer einwilligen, regelmäßig ihre Shares unterschreiben zu lassen, dann kann die Sammelstelle die Rolle der unterschreibenden Trusted Party übernehmen. Dadurch, dass alle Benutzer zu bestimmten Terminen ihre Shares unterschreiben lassen, sind die Zeitpunkte, an denen ein Benutzer unterschriebene Shares bekommt, unabhängig von den Zeitpunkten, an denen ein Benutzer kritische Nachrichten sendet. Folglich gewinnt die Sammelstelle durch die Signatur der Shares keine weitere Infor-

mation über das typische Verhalten des Benutzers. Durch diese Cut-and-Choose-Technik kann die Sammelstelle sicherstellen, dass alle Benutzer korrekte Shares benutzen. Die Benutzer können auf der anderen Seite durch die Verwendung von blinden Signaturen sicher sein, dass die Sammelstelle vorab beim Signieren kein Wissen über die zu benutzenden Shares gewinnt.

5 Historie der Nachrichten

Im vorherigen Kapitel haben wir gezeigt, dass es möglich ist, Vorratsdaten bei der Sammelstelle zu speichern und gleichzeitig den Inhalt der Daten vor unbefugtem Zugriff zu schützen. Dies gilt insbesondere für Daten, die außerhalb der aktuellen Vorhaltezeit liegen, das heißt, für sogenannte *alte* Daten. Man könnte meinen, dass die Sammelstelle diese Daten nun von allein löschen würde, da sie wegen der Rückwärtssicherheit deren Inhalt nicht mehr entschlüsseln kann. Interessanterweise erlauben es diese auf den ersten Blick nutzlosen Daten, zusätzliche Informationen über den Benutzer zu sammeln. Diese zusätzliche Informationen nennen wir die *Historie der Nachrichten*. Sie spiegelt den Verlauf der Interaktion der Benutzer mit dem System wider. Die Historie ist kein spezielles Phänomen unseres Schemas und lässt sich auf die meisten Szenarien verallgemeinern, bei denen Daten im zeitlichen Verlauf gespeichert werden.

Zunächst betrachten wir anhand unseres Schemas zur privaten Vorratsdatenspeicherung, welches zusätzliche Wissen wir aus alten Daten gewinnen können und warum sie die Privatheit gefährden. Anschließend wird im Abschnitt 5.2 der Begriff der Historie formal definiert und ein Überblick über den Rest des Kapitels gegeben.

5.1 Gewinn von zusätzlichem Wissen aus der Historie bei der Entschlüsselung von Vorratsdaten

Die Sammelstelle kann bei Verwendung des Schwellwert-Schemas zur privaten Vorratsdatenspeicherung für jedes Pseudonym ein Profil der Zeitpunkte erstellen, an denen sie die mit diesem Pseudonym versehenen kritischen Nachrichten erhalten hat. Dieses Profil erstreckt sich auf alle Daten und somit auch auf alte Daten. Die Sammelstelle erhält dadurch ein Profil der Zeitpunkte, an denen ein Benutzer kritische Aktionen durchgeführt hat. Kann ein Benutzer durch Übertritt des Schwellwertes an kritischen Aktionen identifiziert werden, kann die Sammelstelle den Benutzer mit seinem Pseudonym in Verbindung bringen und damit dem Benutzer das richtige Profil zuordnen. Das Profil kann dann durch den Identifikationsmechanismus des Fingerabdrucks um alle unkritischen Nachrichten erweitert werden. Dadurch enthält das Profil alle gespeicherten Daten des Benutzers, das heißt, seine Historie.

Insbesondere beinhaltet dies auch die Daten, die außerhalb der Vorhaltezeit liegen. Im Allgemeinen können wir nicht davon ausgehen, dass dieses Profilwissen aus den entschlüsselbaren Nachrichten hervorgeht. In unseren Anwendungsbeispielen könnten die Behörden folglich feststellen, ob ein Verkehrsteilnehmer in der Vergangenheit ein notorischer Sünder war oder wie groß das Transaktionsvolumen eines Bankkunden gewesen sein könnte. Bei einer idealen Vorratsdatenspeicherung wäre dieses zusätzliche Wissen um die Benutzerprofile durch die Löschung zerstört worden. Dieses Löschen können wir bei einer digitalen Speicherung nicht garantieren. Einerseits hat die Sammelstelle ein Interesse an jedem Wissen über die Daten und andererseits ist die Löschung der Daten, der notwendigen Backups und eventuell gemachter Kopien schwer sicherzustellen. Ein Schema zur privaten Vorratsdatenspeicherung sollte die Realisierung der idealen Vorratsdatenspeicherung sein. Da wir das Löschen der Daten bei einer potentiell bösartigen Sammelstelle nicht garantieren können, sollte ein Schema dem Idealfall möglichst nahe kommen und die Historie der Daten schützen.

5.2 Definition der Historie und Überblick der Ansätze zu ihrem Schutz

Die *Historie eines Benutzers bis zu einem Zeitpunkt t* ist die Menge aller Nachrichten dieses Benutzers, die bis zum Zeitpunkt t gespeichert wurden. Die *Historie einer Nachricht M* ist die Historie des entsprechenden Benutzers der Nachricht M bis zum Zeitpunkt $\text{time}(M)$. Diese Definition können wir analog in allen Szenarien verwenden, in denen Daten im zeitlichen Verlauf auftreten oder gespeichert werden.

Ist die Historie von Daten zu schützen, aber ihre Löschung nicht zu garantieren, dann muss durch das verwendete Protokoll die Zuordnung der Daten zu einer Historie verhindert oder zumindest erschwert werden. Im Fall der privaten Vorratsdatenspeicherung müssen wir darüber hinaus den Schutz der Daten ihrem Alter entsprechend anpassen. Es soll nur die Historie der Daten geschützt werden, die außerhalb der aktuellen Vorhaltezeit liegen. Der Zusammenhang der Daten innerhalb der Vorhaltezeit soll bei Entschlüsselung voll erkennbar sein.

Wir wollen im Folgenden Ansätze zum Schutz der Historie betrachten und ihre Eigenschaften analysieren. Dabei werden wir das besondere Augenmerk auf die Anwendbarkeit der Ansätze auf die private Vorratsdatenspeicherung richten. Wir werden beleuchten, wie man „Erschwerung der Zuordnung“ fassen kann. Bei allen Ansätzen müssen wir zum Schutz der Historie Kompromisse bei anderen Parametern machen. Es werden in dieser Arbeit mehrere Protokolle vorgestellt und bezüglich ihrer Parameter diskutiert. Vorversionen von Verfahren, die in diesem Kapitel beschrieben werden, finden wir in [58].

Zunächst wollen wir in Abschnitt 5.3 eine einfache Erweiterung des Schwellwert-Schemas untersuchen. Anschließend betrachten wir in Abschnitt 5.4 Ansätze, in denen der Schutz der Historie durch eine Lockerung der Schwellwertbedingung herbeigeführt wird. Anstatt des konstanten Schwellwertes d von kritischen Nachrichten innerhalb eines Zeitabschnitts betrachten wir einen Toleranzbereich, in dem sich der Schwellwert bewegen kann. Liegt die Anzahl der kritischen Nachrichten in einem Abschnitt unterhalb des Toleranzbereichs, so ist die Öffnung der Secret Shares nicht möglich, liegt sie oberhalb, so wird die Öffnung garantiert. Innerhalb des Toleranzbereichs hängt die Öffnung von der Historie der Daten ab. Dieser Ansatz gewährleistet, dass für jeden Benutzer die Historien aller Nachrichten geschützt sind, die vor dem ersten Zeitabschnitt liegen, für den Nachrichten entschlüsselt

werden können. In Abschnitt 5.5 untersuchen wir einen Ansatz, bei dem die Zuordnung der Daten einer Historie prinzipiell möglich ist. Für alte Daten wird die Zuordnung aber zunehmend unpraktikabel. Dafür bleibt der Schwellwert konstant bei d. Favorisieren wir eine effiziente Zuordnung von Daten und möchten wir zudem mit großer Sicherheit, dass d kritische Nachrichten innerhalb eines Zeitintervalls zur Entschlüsselung ausreichen, dann können wir die in den Kapiteln 6 und 7 beschriebenen probabilistischen Ansätze verwenden. Bei diesem Verfahren können wir die *Entkopplung der Historie* mit großer Sicherheit nach einer gewissen Anzahl von empfangenen kritischen Nachrichten garantieren, das heißt, die Nachrichten können nicht mehr mit den folgenden in Verbindung gebracht werden. Dieser Ansatz eignet sich besonders für Anwendung außerhalb der Vorratsdatenspeicherung, beispielsweise dem privaten Marketing, siehe Abschnitte 1.2.5 und 1.4.

5.3 Eine einfache Erweiterung des Schwellwert-Schemas

Zunächst betrachten wir eine einfache Erweiterung des Schwellwert-Schemas als Ansatz zur Entkopplung der Historie. Die Historie des Benutzers wäre entkoppelt, wenn die Sammelstelle bei erstmaliger Überschreitung des Schwellwertes dem Benutzer nur Nachrichten des entsprechenden Zeitabschnitts sowie der folgenden Abschnitte zuordnen kann. Die Nachrichten aus früheren Zeitabschnitten soll die Sammelstelle nicht mit dem Benutzer in Verbindung bringen können. Dadurch, dass bisher alle Nachrichten eines Benutzers mit derselben ID versehen werden, können wir die Nachrichten derselben Historie zuordnen. Wir benötigen die IDs hauptsächlich, um Nachrichten desselben Zeitabschnitts zu erkennen und um bei Übertreten des Schwellwertes die Entschlüsselung durchführen zu können. Konnte die Sammelstelle einen Benutzer nach Erreichen des Schwellwertes identifizieren, dann kann sie mittels des Identifikationsmechanismus alle Nachrichten des Benutzers erkennen. Anstatt einer einzigen ID für alle Zeitabschnitte können wir jedem Zeitabschnitt und damit jedem Polynom eine eigene ID geben. Werden die Shares für eine kritische Nachricht gemeinsam übertragen, dann kann die Sammelstelle immer noch feststellen, welche IDs zu demselben Benutzer gehören. Empfängt die Sammelstelle zwei oder mehr kritische Nachrichten in jedem Zeitabschnitt, dann kann die Sammelstelle sogar alle IDs eines Benutzers in Zusammenhang bringen und damit die Historie berechnen.

Betrachten wir daher als Erweiterung des Schemas aus Abschnitt 4.4 die Möglichkeit, dass der Provider die Shares für die Zeitabschnitte nicht gemeinsam sendet. Dazu generiert der Provider anstatt einer kritischen Nachricht M eine unkritische Nachricht mit Inhalt load(M) und für jeden entsprechenden Zeitabschnitt eine *Share-Nachricht* ohne load-Eintrag. Um zu verhindern, dass bei einer ähnlichen Nutzlast eine Verbindung zwischen den Share-Nachrichten entstehen kann, beschränkt sich der Inhalt einer Share-Nachricht lediglich auf den Identifikationsmechanismus. Dieses verhindert zudem, dass der Kommunikationsaufwand unnötig anwächst.

Erweiterte Generierung von kritischen Nachrichten

Nun wird beschrieben, wie sich die Generierung der kritischen Nachrichten gegenüber dem Schwellwert-Schema ändert. Für jeden Benutzer und jeden Zeitabschnitt $\mathcal{T}_j$ generiert sein Provider in der Initialisierungsphase zufällig ein eindeutiges Pseudonym P_j.

1. Ein Benutzer führt eine kritische Aktion in Runde t aus.

2. Für jeden Zeitabschnitt $\mathcal{T}_j \in \Pi_t$ generiert der Provider eine Share-Nachricht M_j mit

$$\begin{aligned} \text{time}(M_j) &= t, \\ \text{ID}(M_j) &= P_j, \\ \text{share}(M_j) &= p_j((t \bmod \Delta) + 1) \end{aligned}$$

   und berechnet load(M_j), wobei subj$(M_j) = \emptyset$.

3. Zudem generiert der Provider eine unkritische Nachricht M_u mit

$$\begin{aligned} \text{time}(M_u) &= t, \\ \text{ID}(M_u) &= \emptyset, \\ \text{share}(M_u) &= \emptyset \end{aligned}$$

   und setzt für load(M_u) den Inhalt load(M) der ursprünglich einzelnen kritischen Nachricht M des Schemas ein.

4. Der Provider sendet alle Share-Nachrichten und alle unkritischen Nachrichten aller Benutzer am Ende der Runde t in zufälliger Reihenfolge an die Sammelstelle.

Erhält die Sammelstelle innerhalb eines Zeitabschnitts d oder mehr Share-Nachrichten eines Benutzers mit derselben ID, dann kann die Sammelstelle Schlüssel und Identität rekonstruieren und alle (unkritischen) Nachrichten des Benutzers für diesen und folgende Zeitabschnitte identifizieren und entschlüsseln.

Man beachte die abweichende Wahl der Stützstelle $x = (t \bmod \Delta) + 1$ für die Polynome im Vergleich zum ursprünglichen Schema. Da jeder Zeitabschnitt die Länge Δ hat, wird für jedes Polynom der Wert x nur einmal als Stützstelle benutzt und wir können die so berechnete Stützstelle x für jedes Polynom anstelle der ursprünglichen Stützstellen verwenden.

Da die Share-Nachrichten gemischt versendet werden, gibt es für die Sammelstelle keine direkt zu erkennende natürliche Reihenfolge der Shares mehr. Jedoch treten die IDs von überlappenden Zeitabschnitten eines Benutzers gleichzeitig auf, wenn dieser eine kritische Aktion ausführt. Führt ein Benutzer in einem Zeitabschnitt mehr als eine kritische Aktion aus, so kann die Sammelstelle das gemeinsame Auftreten von IDs feststellen und gewinnt somit Information über den Zusammenhang der IDs. Im Allgemeinen können wir erwarten, dass sich das Verhalten von Benutzern unterscheidet und damit auch die Reihe der Zeitpunkte, an denen die Benutzer kritische Aktionen ausführen. Dieses erleichtert der Sammelstelle das Erkennen von zusammengehörigen IDs, insbesondere, wenn das Gesamtaufkommen von kritischen Aktionen in einer Runde niedrig ist. Führt ein Benutzer regelmäßig kritische Aktionen aus, dann wird die Rekonstruktion der Historie durch das getrennte Senden der Share-Nachrichten erschwert. Je mehr kritische Aktionen ein Benutzer begeht, desto einfacher wird es für die Sammelstelle zusammengehörende IDs zu erkennen und die Historie des Benutzers zu bestimmen.

Gibt es einen Zeitabschnitt $\mathcal{T}_j$, in dem der Benutzer keine kritische Aktion ausführt, dann gibt es keine Runde, in der die Sammelstelle die IDs $P_{j'}$ und $P_{j''}$ für beliebige j' und j'' mit $j' < j < j''$ gemeinsam empfängt. In Abbildung 5.1 werden alle mit $\mathcal{T}_j$ überlappenden Zeitabschnitte dargestellt. Wir können leicht erkennen, dass $\mathcal{T}_{j'} \cap \mathcal{T}_{j''} \subseteq \mathcal{T}_j$. Empfängt die Sammelstelle in $\mathcal{T}_j$ keine kritische Nachricht, kann sie folglich die IDs des Benutzers vor und nach diesem Zeitabschnitt nicht in Verbindung bringen. Somit können $P_{j'}$ und $P_{j''}$ in keiner Runde gemeinsam auftreten. Die Historie des Benutzers wird an der Stelle $t_{\min}(\mathcal{T}_j)$

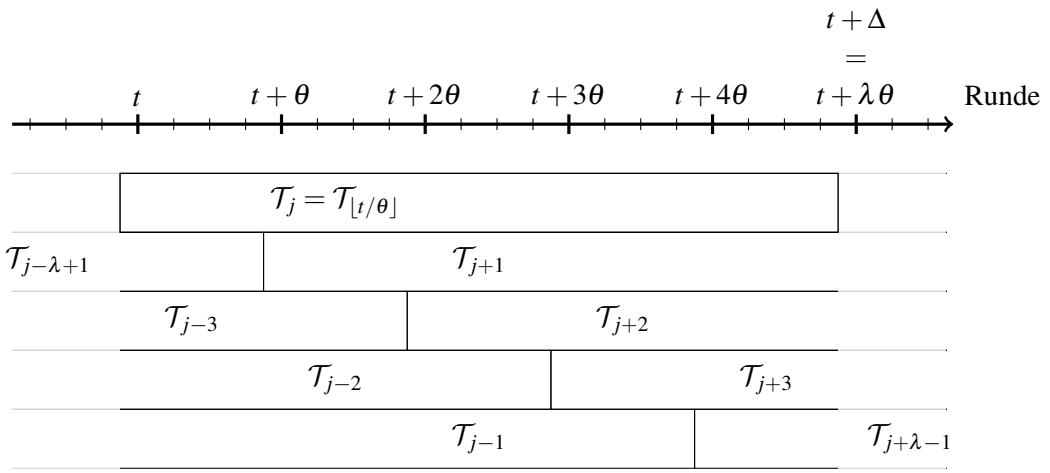


Abbildung 5.1: überlappende Zeitabschnitte mit $\mathcal{T}_j = \mathcal{T}_{\lfloor t/\theta \rfloor}$ für $\lambda = 5$, Versatz $\theta = 4$ und Zeitabschnittslänge $\Delta = \lambda\theta = 20$

entkoppelt, denn als weitere Fortsetzung der Historie ist jede Teilhistorie eines anderen Benutzers möglich, die vor einem Zeitabschnitt $\mathcal{T}_{j'}$ mit $j' \leq j$ liegt, so dass der andere Benutzer keine kritische Aktionen in $\mathcal{T}_{j'}$ durchgeführt hat.

5.4 Entkopplung der Historie durch Lockerung der Schwellwertbedingung

Im letzten Abschnitt haben wir gesehen, dass die Sammelstelle auch bei getrenntem Versand der Share-Nachrichten durch ihr gemeinsames Aufkommen in einer Runde die Historie rekonstruieren kann, wenn regelmäßig kritische Aktionen von einem Benutzer ausgeführt werden. Man beachte, dass eine kleine Anzahl von kritischen Aktionen je Zeitabschnitt ausreichen. Nun wollen wir untersuchen, ob die Generierung einer Share-Nachricht für nur einen anstatt für viele Zeitabschnitte bei der Entkopplung der Historie helfen kann. Dadurch treten keine IDs für eine Aktion eines Benutzers gleichzeitig auf. Die Sammelstelle kann also nicht durch das gleichzeitige Vorkommen der IDs die Historie eines Benutzers rekonstruieren. Jedoch erhält die Sammelstelle nicht mehr für jeden Zeitabschnitt, in dem die Runde einer kritischen Aktion liegt, ein Share. Wir können nicht damit rechnen, dass die Sammelstelle nach d kritischen Aktionen eines Benutzers die Identität und die Schlüssel rekonstruieren kann. Im Folgenden wollen wir die Ausprägung dieser Lockerung untersuchen.

5.4.1 Randomisierte Auswahl einer Share-Nachricht

Für die Analyse der Auswirkung einer zufälligen Wahl einer Share-Nachricht wollen wir das allgemeinere Szenario aus Abschnitt 4.10.1 verwenden. Innerhalb eines Zeitintervalls $\mathcal{T}_j$ beginnen insgesamt $\lambda \in \mathbb{N}$ Zeitintervalle mit einem *Versatz* von $\theta \in \mathbb{N}$. Wir gehen davon aus, dass $\lambda\theta = \Delta$. Damit gilt für jede Runde t, dass $|\Pi_t| = \lambda$, siehe Abbildung 5.1. Für $\lambda = \Delta$ und $\theta = 1$ betrachten wir das ursprüngliche Schwellwert-Schema.

Im Folgenden beschreiben wir die Generierung einer kritischen Nachricht bei randomisierter Auswahl.

Wie in Abschnitt 5.3 generiert ein Provider für jeden seiner Benutzer und jeden Zeitabschnitt $\mathcal{T}_j$ in der Initialisierungsphase zufällig ein eindeutiges Pseudonym P_j.

1. Ein Benutzer führt eine kritische Aktion in Runde t aus.
2. Der Provider wählt uniform, das heißt mit Wahrscheinlichkeit $\frac{1}{\lambda}$, ein Polynom p_j, so dass $\mathcal{T}_j \in \Pi_t$ und generiert die kritische Nachricht M mit

$$\begin{aligned} \text{time}(M) &= t, \\ \text{ID}(M) &= P_j, \\ \text{share}(M) &= p_j((t \bmod \Delta) + 1) \end{aligned}$$

   und berechnet load(M) entsprechend dem Schwellwert-Schema.
3. Der Provider sendet M an die Sammelstelle.

5.4.2 Analyse der benötigten Anzahl von kritischen Nachrichten bis zur Entschlüsselung bei randomisierter Auswahl

Im günstigsten Fall für die Sammelstelle und damit im ungünstigsten Fall für den Benutzer werden für d Nachrichten in Δ Runden immer Shares desselben Polynoms p_j gewählt. Dann wird der Schwellwert nach d kritischen Nachrichten erreicht. Dieser Fall wird mit wachsenden d und λ unwahrscheinlicher.

Betrachten wir nun einen beliebigen Zeitabschnitt $\mathcal{T}_j$. Es überlappen sich $2(\lambda-1)+1 = 2\lambda-1$ Zeitabschnitte während der Runden von $\mathcal{T}_j$. Damit können während der Runden von $\mathcal{T}_j$ Shares von den entsprechenden $2\lambda-1$ Polynomen empfangen werden. Dies kann man in Abbildung 5.1 erkennen. Hat ein Benutzer nun $(d-1)\cdot(2\lambda-1)+1$ kritische Aktionen in dem Zeitabschnitt $\mathcal{T}_j$ ausgeführt, dann hat die Sammelstelle nach dem Taubenschlagprinzip für mindestens ein Polynom $(d-1)+\frac{1}{2\lambda-1} > d-1$ Shares empfangen. Es gibt also mindestens ein Polynom, für das die Anzahl der empfangenen Shares den Schwellwert d überschreitet. Daraus können wir direkt folgern:

Fakt 5.1 *Empfängt die Sammelstelle $(d-1)\cdot(2\lambda-1)+1$ kritische Nachrichten für einen Benutzer innerhalb eines Zeitabschnitts $\mathcal{T}_j$, dann hat die Sammelstelle mit Wahrscheinlichkeit 1 für mindestens ein Polynom, dessen Zeitabschnitt sich mit $\mathcal{T}_j$ überlappt, d Shares empfangen und kann somit Identität und Schlüssel für diesen Zeitabschnitt rekonstruieren.*

Damit haben wir eine scharfe obere Schranke für die Anzahl der benötigten kritischen Nachrichten. Nun werden wir untersuchen, wie viele Nachrichten mit großer Sicherheit mindestens benötigt werden, um den Schwellwert d für die Shares zu überschreiten.

Sei z die Gesamtanzahl kritischer Nachrichten, die die Sammelstelle in Zeitabschnitt $\mathcal{T}_j$ für einen Benutzer erhalten hat. Für $\tau \in \{j-\lambda+1,\ldots,j+\lambda-1\}$ beschreibe die Zufallsvariable $X_\tau^{(z)}$ die Anzahl der Nachrichten M, die in Zeitabschnitt $\mathcal{T}_j$ von der Sammelstelle empfangen wurden und dem Polynom p_τ zuzuordnen sind, wenn wir $\mathcal{T}_\tau$ auf den gesamten Zeitabschnitt $\mathcal{T}_j$ ausdehnen. Da bei der Generierung einer kritischen Nachricht das Polynom, für das die kritische Nachricht ein Share enthält, uniform und mit Wahrscheinlichkeit $\frac{1}{\lambda}$ gewählt wird, seien alle $X_\tau^{(z)}$ binomialverteilt mit Parametern z und $\frac{1}{\lambda}$ Sie haben den Erwartungswert

$$\mathbb{E}\left(X_\tau^{(z)}\right) = \frac{z}{\lambda}.$$

Indem wir die Zeitabschnitte ausdehnen, aber gleichzeitig den Parameter $\frac{1}{\lambda}$ beibehalten, vergrößern wir höchstens die Anzahl der kritischen Nachrichten, die einem Zeitabschnitt $\mathcal{T}_\tau$ zugerechnet werden. Somit beschreibt $X_\tau^{(z)}$ eine obere Schranke für die Anzahl der Nachrichten, die p_τ tatsächlich zuzuordnen sind und die innerhalb des Zeitabschnitts $\mathcal{T}_j$ empfangen wurden. Dadurch können wir eine untere Schranke für die Anzahl der Nachrichten herleiten, bis in einem der Abschnitte $\mathcal{T}_\tau$ mindestens d Nachrichten mit großer Wahrscheinlichkeit vorkommen.

Lemma 5.2 *Für $c > 1$ und $z \leq \frac{d\cdot\lambda}{c}$ gilt:*

$$\Pr\left[\max_{\tau\in\{j-\lambda+1,\ldots,j+\lambda-1\}} X_\tau^{(z)} \geq d\right] \leq e^{-\frac{\min(c-1,(c-1)^2)}{3c}d+\ln(2\lambda-1)}.$$

BEWEIS Sei $\delta = c-1$ und damit $\delta > 0$. Dann gilt für $\tau \in \{j-\lambda+1,\ldots,j+\lambda-1\}$ unter Verwendung der Chernoff-Ungleichung, dass

$$\begin{aligned}
\Pr\left[X_\tau^{(z)} \geq d\right] &= \Pr\left[X_\tau^{(z)} \geq c\cdot\frac{d\cdot\lambda}{c}\cdot\frac{1}{\lambda}\right] \\
&= \Pr\left[X_\tau^{(z)} \geq (1+\delta)\frac{z}{\lambda}\right] \\
&= \Pr\left[X_\tau^{(z)} \geq (1+\delta)\mathbb{E}\left(X_\tau^{(z)}\right)\right] \\
&\leq e^{-\frac{\min(\delta,\delta^2)}{3}\mathbb{E}\left(X_\tau^{(z)}\right)} \\
&= e^{-\frac{\min(c-1,(c-1)^2)}{3}\cdot\frac{z}{\lambda}} \\
&= e^{-\frac{\min(c-1,(c-1)^2)}{3c}d}.
\end{aligned}$$

Allgemein gilt für beliebige Ereignisse $A_1,\ldots,A_\kappa$ die Abschätzung

$$\Pr\left[\bigcup_i A_i\right] \leq \sum_i \Pr[A_i].$$

Sei nun A_τ das Ereignis, dass $X_\tau^{(z)} \geq d$. Folglich ist

$$\begin{aligned}
\Pr\left[\max_{\tau\in\{j-\lambda+1,\ldots,j+\lambda-1\}} X_\tau^{(z)} \geq d\right] &= \Pr\left[\bigcup_{\tau\in\{j-\lambda+1,\ldots,j+\lambda-1\}} A_\tau\right] \\
&\leq \sum_{\tau\in\{j-\lambda+1,\ldots,j+\lambda-1\}} \Pr[A_\tau] \\
&= (2\lambda-1)e^{-\frac{\min(c-1,(c-1)^2)}{3c}d} \\
&= e^{-\frac{\min(c-1,(c-1)^2)}{3c}d+\ln(2\lambda-1)}.
\end{aligned}$$

■

Damit erhalten wir:

Theorem 5.3 *Sei $z \leq \frac{d\cdot\lambda}{2}$. Mit einer Sicherheit von $1-e^{-\frac{d}{6}+\ln(2\lambda-1)}$ müssen $\Theta(\lambda d)$ kritische Nachrichten eines Benutzers innerhalb von Δ Runden übertragen werden, damit die Sammelstelle Identität und Schlüssel des Benutzers aus den empfangenen Shares rekonstruieren kann.*

BEWEIS Setzen wir $c = 2$, so folgt aus Lemma 5.2, dass die Sammelstelle mit einer Sicherheit von $1-e^{-\frac{d}{6}+\ln(2\lambda-1)}$ mindestens $\frac{1}{2}d\cdot\lambda$ kritische Nachrichten eines Benutzers innerhalb eines Zeitabschnitts empfangen muss, damit mindestens d Shares für ein Polynom vorliegen. Nach Fakt 5.1 müssen dazu höchstens $(d-1)(2\lambda-1)+1 \leq 2d\cdot\lambda$ kritische Nachrichten übertragen werden. Daraus folgt die Aussage. ■

Die Werte für d und λ sind nach oben durch Δ beschränkt. Aus Effizienzgründen wird λ keine großen Werte annehmen, auch wenn Δ wächst. Daher können wir annehmen, dass $\frac{d}{10} > \ln(2\lambda)$. Damit können wir sagen, dass mit großer Sicherheit $\Theta(\lambda d)$ kritische Nachrichten eines Benutzers innerhalb von Δ Runden übertragen werden müssen, damit Identifikation und Entschlüsselung möglich werden.

Bei kleineren Werten für d kann es allerdings vorkommen, dass die Auswahl der Shares ungünstig für den Benutzer erfolgt. Dies kann zur Folge haben, dass er bereits nach wenig mehr als d kritischen Aktionen identifiziert werden kann. Auch bei größeren Werten für d bedeutet der Einsatz dieses randomisierten Verfahrens ein Risiko, früh erkannt zu werden, auch wenn dieses Risiko mit wachsendem d abnimmt. Unter dieser Unsicherheit kann die Akzeptanz des Benutzers für das randomisierte Schema leiden. Die Akzeptanz wird zunehmen, wenn er genauer einschätzen kann, wie viele kritische Aktionen er in einem Zeitabschnitt noch durchführen kann. Aus diesem Grund werden wir im nächsten Abschnitt eine deterministische Variante des Verfahrens betrachten, die immer die für den Benutzer günstigste Wahl trifft.

5.4.3 Deterministische Auswahl einer Share-Nachricht

Der Benutzer ist daran interessiert, seine Identifizierung und das Öffnen der Schlüssel lang hinauszuzögern, um möglichst viele kritische Aktionen durchführen zu können. Wenn der Benutzer die Auswahl der Polynome für die Share-Nachrichten beeinflussen könnte, dann würde er, höchstens $(d-1)$-mal dasselbe Polynom auswählen. Damit er in der Zukunft möglichst viele kritische Aktionen durchführen kann, wird der Benutzer immer das „älteste“ Polynom auswählen. Das älteste Polynom entspricht dem Zeitabschnitt mit dem kleinstmöglichen Index, so dass die aktuelle Runde zu diesem Zeitabschnitt gehört und das Polynom bisher weniger als $(d-1)$-mal ausgewählt wurden. Ist für jeden Zeitabschnitt, zu dem der aktuelle Zeitpunkte gehört, das entsprechende Polynom bereits $(d-1)$-mal ausgewählt worden, gibt es kein ältestes Polynom. Bei jeder Auswahl eines Polynoms der Schwellwert überschritten. Ein Benutzer wird in diesem Fall das „jüngste“ Polynom für die Share-Nachricht wählen, das heißt das Polynom mit dem größtmöglichen Index. Damit wird die Anzahl der Nachrichten minimiert, die die Sammelstelle öffnen kann. Für unser Schema bedeutet dies:

1. Ein Benutzer führt eine kritische Aktion in Runde t aus.

2. Der Provider wählt p_j mit minimalem j, so dass p_j bisher weniger als $(d-1)$-mal ausgewählt wurde und $\mathcal{T}_j \in \Pi_t$. Ansonsten wählt der Provider p_j mit maximalem j, so dass $\mathcal{T}_j \in \Pi_t$. Er erzeugt die kritische Nachricht M mit

$$\begin{aligned} \text{time}(M) &= t, \\ \text{ID}(M) &= P_j, \\ \text{share}(M) &= p_j((t \bmod \Delta) + 1) \end{aligned}$$

   und berechnet $\text{load}(M)$ entsprechend dem Schwellwert-Schema.

3. Der Provider sendet M an die Sammelstelle.

5.4.4 Analyse der benötigten Anzahl von kritischen Nachrichten bis zur Entschlüsselung bei deterministischer Auswahl

Wie in der Analyse der randomisierten Auswahl betrachten wir einen beliebigen Zeitabschnitt $\mathcal{T}_j$. Der günstigste Fall für den Benutzer liegt dann vor, wenn er in keinem der mit $\mathcal{T}_j$ überlappenden Zeitabschnitte in den Runden vor $t_{\min}(\mathcal{T}_j)$ eine kritische Aktion durchgeführt hat. Im schlechtesten Fall hat der Benutzer in diesen Zeitabschnitten bereits $d-1$ kritische Aktionen ausgeführt.

Theorem 5.4 *Sei* $c' = \lfloor \frac{(\lambda-1)\theta}{d-1} \rfloor$ *und* $c'' = (\lambda-1)\theta \bmod (d-1)$. *Darf ein Benutzer pro Runde eine kritische Aktion durchführen und werden die Share-Nachrichten deterministisch ausgewählt, dann kann ein Benutzer*

1. *im besten Fall* $\min(\Delta,\ (\lambda-1)\cdot\min(\theta,d-1)+\lambda(d-1))$ *kritische Aktionen,*
2. *im schlechtesten Fall für* $\theta < d-1$

$$\min\left(\Delta,\ \min\left((\lambda-1)\theta,(\lambda-c')(d-1)-c''\right)+\lambda(d-1)\right)$$

und für $\theta \geq d-1$

$$\min(\Delta,\lambda(d-1))$$

kritische Aktionen in einem Zeitabschnitt durchführen, ohne dass die Sammelstelle aus den in diesem Zeitabschnitt empfangenen Shares die Identität und die Entschlüsselungsschlüssel des Benutzers rekonstruieren kann.

BEWEIS Sei $\mathcal{T}_j = \{j\theta,\ldots,(j+\lambda)\theta-1\}$ ein beliebiger Zeitabschnitt. Führt ein Benutzer eine kritische Aktion aus, wird ein Polynom für die zu generierende Share-Nachricht ausgewählt. Wenn ein Benutzer nur eine kritische Aktion pro Runde durchführen darf, dann werden im Zeitabschnitt $\mathcal{T}_j$ insgesamt höchstens Δ-mal Polynome ausgewählt. Es gilt für $i \in \{1,\ldots,\lambda\}$ nach Definition der Zeitabschnitte, dass

$$\begin{aligned}\mathcal{T}_{j-(\lambda-i)} &= \{(j-(\lambda-i))\theta,\ldots,(j+i)\theta-1\} \quad \text{und}\\ \mathcal{T}_{j+(\lambda-i)} &= \{(j+i)\theta,\ldots,(j+(\lambda-i))\theta-1\}.\end{aligned}$$

Folglich gilt, dass

$$|\mathcal{T}_j \cap \mathcal{T}_{j-(\lambda-i)}| = |\mathcal{T}_j \cap \mathcal{T}_{j+(\lambda-i)}| = i\theta.$$

Siehe dazu auch 5.1. Somit dürfen die Polynome $p_{j-(\lambda-i)}$ und $p_{j+(\lambda-i)}$ jeweils höchstens $\min(i\theta,d-1)$-mal für eine Share-Nachricht ausgewählt werden, ohne dass die Sammelstelle eine Entschlüsselung herbeiführen kann. Da $\left|\mathcal{T}_j \cap \bigcup_{i=1}^{\lambda-1}\mathcal{T}_{j-(\lambda-i)}\right| = (\lambda-1)\theta$, dürfen in der Summe die Polynome $p_{j-(\lambda-1)},\ldots,p_{j-1}$ höchstens $((\lambda-1)\cdot\min(\theta,d-1))$-mal ausgewählt werden. Die Polynome $p_j,\ldots,p_{j+(\lambda-1)}$ können jeweils höchstens $(d-1)$-mal ausgewählt werden, ohne dass die Sammelstelle eine Entschlüsselung herbeiführen kann. Damit können im besten Fall

$$\min(\Delta,(\lambda-1)\cdot\min(\theta,d-1)+\lambda(d-1))\text{-mal}$$

Polynome ausgewählt, beziehungsweise kritische Aktionen von einem Benutzer durchgeführt werden.

Im schlechtesten Fall wurden einige Polynome, deren Zeitabschnitte sich mit $\mathcal{T}_j$ überlappen, vor der Runde $t_{\min}(\mathcal{T}_j)$ bereits $(d-1)$-mal ausgewählt. Dieses betrifft die Zeitabschnitte $\mathcal{T}_{j-(\lambda-1)},\ldots,\mathcal{T}_{j-1}$. In der Summe können diese in maximal $(\lambda-1)\theta$ Runden ausgewählt worden sein. Sei $c' = \lfloor \frac{(\lambda-1)\theta}{d-1} \rfloor$. Da immer das älteste Polynom ausgewählt wird, sind im schlechtesten Fall die Polynome $p_{j-(\lambda-1)},\ldots,p_{j-(\lambda-c')}$ bereits $(d-1)$-mal ausgewählt worden.

Falls $\theta \geq d-1$, dann sind alle Polynome $p_{j-(\lambda-1)},\ldots,p_{j-1}$ demnach $(d-1)$-mal ausgewählt worden und in $\mathcal{T}_j$ können $p_j,\ldots,p_{j+(\lambda-1)}$ noch jeweils $(d-1)$-mal ausgewählt werden. Folglich kann der Benutzer in diesem Fall höchstens

$$\min(\Delta, \lambda(d-1))$$

kritische Aktionen durchführen.

Falls $\theta < d-1$ und damit $c' < \lambda-1$ können nur die Polynome $p_{j-(\lambda-1)},\ldots,p_{j-(\lambda-c')}$ nicht mehr in $\mathcal{T}_j$ ausgewählt werden. Das Polynom $p_{j-(\lambda-c')+1}$ ist demnach höchstens $c'' = ((\lambda-1)\theta \bmod (d-1))$-mal vor der Runde $t_{\min}(\mathcal{T}_j)$ ausgewählt worden.

Zusammengefasst heißt das für den Zeitabschnitt $\mathcal{T}_j$:

1. Die Polynome $p_{j-(\lambda-1)},\ldots,p_{j-1}$ können insgesamt höchstens $((\lambda-1)\theta)$-mal ausgewählt werden.
2. Die Polynome $p_{j-(\lambda-1),\ldots,j-(\lambda-c')}$ können nicht ausgewählt werden.
3. Das Polynom $p_{j-(\lambda-c')+1}$ kann höchstens $((d-1)-c'')$-mal ausgewählt werden.
4. Die Polynome $p_{j-(\lambda-c')+2},\ldots,p_{j-1}$ können insgesamt höchstens $((\lambda-c'-1)\cdot(d-1))$-mal ausgewählt werden.
5. Die Polynome $p_j,\ldots,p_{j+(\lambda-1)}$ können insgesamt höchstens $\lambda(d-1)$-mal ausgewählt werden.
6. Insgesamt dürfen höchstens Δ kritische Aktionen durchgeführt werden.

Daraus ergibt sich für den schlechtesten Fall, dass

$$\min\big(\Delta,\ \min\big((\lambda-1)\theta, (\lambda-c')(d-1)-c''\big)+\lambda(d-1)\big)$$

kritische Aktionen in Zeitabschnitt $\mathcal{T}_j$ durchgeführt werden können. ∎

Im Folgenden betrachten wir, wie sich diese Anzahl an kritischen Aktionen ändert, falls zugelassen ist, dass der Benutzer bis zu λ Aktionen pro Runde ausführen darf. Somit dürfen bis zu λ Polynome pro Runde ausgewählt werden, für die jeweils die entsprechende Share-Nachricht gesendet wird. Da die Shares in einer Runde durch die Auswertung der Polynome an derselben Stützstelle berechnet werden, kann ein Polynom höchstens einmal pro Runde ausgewählt werden.

Theorem 5.5 *Sei $c' = \lfloor \frac{d-1}{\theta} \rfloor$. Darf ein Benutzer pro Runde mehrere kritische Aktionen durchführen und werden die Share-Nachrichten deterministisch ausgewählt, dann kann ein Benutzer*

1. *im besten Fall $2\theta\frac{c'(c'+1)}{2} + (2(\lambda-c')-1)(d-1))$ kritische Aktionen,*
2. *im schlechtesten Fall $\lambda(d-1)$ kritische Aktionen*

in einem Zeitabschnitt durchführen, ohne dass die Sammelstelle aus den in diesem Zeitabschnitt empfangenen Shares die Identität und die Entschlüsselungsschlüssel des Benutzers rekonstruieren kann.

BEWEIS Sei $\mathcal{T}_j = \{j\theta, \ldots, (j+\lambda)\theta - 1\}$ ein beliebiger Zeitabschnitt. Für $i \in \{1, \ldots, \lambda\}$ gilt, dass der Benutzer die Polynome $p_{j-(\lambda-i)}$ und $p_{j+(\lambda-i)}$ jeweils höchstens $\min(i\theta, d-1)$-mal für eine Share-Nachricht auswählen darf, ohne dass die Sammelstelle eine Entschlüsselung herbeiführen kann. Da $c' = \lfloor \frac{d-1}{\theta} \rfloor$ ist $i = c'$ die größtmögliche natürliche Zahl, so dass $i\theta < d-1$. Daraus ergibt sich:

$$\min(i\theta, d-1) = \begin{cases} i\theta, & \text{falls } i \leq c' \text{ und} \\ d-1 & \text{sonst.} \end{cases}$$

Im besten Fall hat der Benutzer in keinem der mit $\mathcal{T}_j$ überlappenden Zeitabschnitte vor der Runde $t_{\min}(\mathcal{T}_j)$ eine kritische Aktion durchgeführt. Somit ist die größte Anzahl b von kritischen Aktionen, die ein Benutzer in $\mathcal{T}_j$ durchführen kann, ohne eine Identifizierung und Entschlüsselung zu riskieren, gegeben durch:

$$\begin{aligned} b &= \min(\lambda\theta, d-1) + 2\textstyle\sum_{i\in\{1,\ldots,\lambda-1\}} \min(i\theta, d-1) \\ &= \min(\Delta, d-1) + 2\textstyle\sum_{i\in\{1,\ldots,\lambda-1\}} \min(i\theta, d-1) \\ &= (d-1) + 2\left(\sum_{i=1}^{c'} i\theta + \sum_{i=c'+1}^{\lambda-1} (d-1)\right) \\ &= (d-1) + 2\theta\frac{c'(c'+1)}{2} + 2(\lambda - 1 - c')(d-1) \\ &= 2\theta\frac{c'(c'+1)}{2} + (2(\lambda - c') - 1)(d-1). \end{aligned}$$

Im schlechtesten Fall hat der Benutzer für alle Polynome, deren Zeitabschnitte sich mit $\mathcal{T}_j$ überlappen, vor der Runde $t_{\min}(\mathcal{T}_j)$ bereits die maximal mögliche Anzahl von Share-Nachrichten generiert. Für $i \in \{1, \ldots, \lambda - 1\}$ gilt $|\mathcal{T}_{j-i} \setminus \mathcal{T}_j| = i\theta$. Folglich kann der Benutzer das Polynom p_{j-i} höchstens $((d-1) - \min(i\theta, d-1))$-mal nach Runde $t_{\min}(\mathcal{T}_j)$ auswählen, ohne dass die Sammelstelle seine Identität und die Entschlüsselungsschlüssel aus den Shares dieser Polynome rekonstruieren kann. Damit ist in diesem Fall die größte Anzahl s an kritischen Aktionen, die ein Benutzer in $\mathcal{T}_j$ durchführen kann, ohne eine Identifizierung und Entschlüsselung zu riskieren, gegeben durch:

$$\begin{aligned} s &= \sum_{i=0}^{\lambda-1} \min(i\theta, d-1) + \sum_{i=1}^{\lambda-1} ((d-1) - \min(i\theta, d-1)) \\ &= (d-1) + \sum_{i=1}^{\lambda-1} \min(i\theta, d-1) + (\lambda-1)(d-1) - \sum_{i=1}^{\lambda-1} \min(i\theta, d-1) \\ &= \lambda(d-1). \end{aligned}$$

■

Ist $\theta \geq d-1$ und $\lambda(d-1) \leq \Delta$, dann sind die Grenzen der Theoreme 5.4 und 5.5 für den schlechten Fall identisch. Zudem sind unter diesen Voraussetzungen beide Grenzen für den besten Fall identisch und entsprechen genau der Grenze von Fakt 5.1. Durch die obigen Theoreme erhalten wir somit scharfe obere Schranken für die Anzahl der benötigten Share-Nachrichten bis zur Identifizierung und Entschlüsselung.

5.4.5 Entkopplung der Historie bei deterministischer und randomisierter Auswahl

Zunächst können wir wie bei der einfachen Erweiterung des Schwellwert-Schemas feststellen, dass die Historie eines Benutzers entkoppelt wird, wenn dieser Benutzer für Δ Runden keine kritische Aktion durchführt.

Die Analyse des gemeinsamen Auftretens von kritischen Nachrichten ist bei diesem Ansatz nicht möglich, da Pseudonyme eines Benutzers nicht gleichzeitig auftreten. In einer Runde werden niemals zwei Pseudonyme desselben Benutzers verwendet. Darf der Benutzer nur eine kritische Aktion pro Runde ausführen, dann kann die Sammelstelle für die Historie einer Share-Nachricht alle Nachrichten und deren Historien ausschließen, die gleichzeitig mit der Share-Nachricht empfangen wurden. Um die Analyse solcher gegenseitigen Ausschlüsse sowie die Analyse gleichzeitig auftretender IDs zu erschweren, könnte man anstelle von einem beziehungsweise aller Shares der Polynome nur einige Shares einer Auswahl von Polynomen senden. Darf der Benutzer mehrere kritische Aktionen in einer Runde durchführen, dann kann die Sammelstelle ohne weitere a-priori-Information über das Benutzerverhalten keine IDs ausschließen. Durch die Analyse des gemeinsamen Vorkommens von IDs sind die Historien von Benutzern eher angreifbar, die häufig kritische Aktionen durchführen.

Verwenden wir das Verfahren mit deterministischer Polynomauswahl zur Sharegenerierung, so kann die Sammelstelle ausschließen, dass zwei Pseudonyme P' und P'' für entsprechende Zeitabschnitte $\mathcal{T}'$ und $\mathcal{T}''$ zu demselben Benutzer gehören. Dazu muss die Sammelstelle feststellen, dass

1. $t_{\min}(\mathcal{T}') < t_{\min}(\mathcal{T}'')$, zum Beispiel anhand des zeitlichen Abstandes zwischen der ersten Share-Nachricht für $\mathcal{T}'$ und der letzten Share-Nachricht für $\mathcal{T}''$ beziehungsweise der entsprechenden Polynome.

2. bis zu irgendeinem Zeitpunkt des Überlappungsbereichs von $\mathcal{T}'$ und $\mathcal{T}''$ die Anzahl der empfangenen Share-Nachrichten für $\mathcal{T}'$ kleiner ist als die für $\mathcal{T}''$ beziehungsweise für die entsprechenden Polynome.

Würden die entsprechenden Nachrichten zu der Historie desselben Benutzers gehören, dann würde die Sammelstelle für das Polynom des früheren Zeitabschnitts bis zum letzten Zeitpunkt der Überlappung mindestens genauso viele Share-Nachrichten empfangen haben wie für das Polynom des späteren Zeitabschnitts.

5.5 Vermischung der Historien von Benutzern

Eine Gemeinsamkeit der vorherigen Ansätze ist die Eindeutigkeit der verwendeten Pseudonyme/IDs. Damit wird erreicht, dass die Nachrichten verschiedener Benutzer nicht falsch zugeordnet werden. In diesem Abschnitt werden wir untersuchen, wie der Verzicht auf die Eindeutigkeit der IDs bei der Entkopplung der Historie hilfreich sein kann. Dieser Verzicht auf die Eindeutigkeit der IDs ermöglicht es, dass kritische Nachrichten für unterschiedliche Benutzer dieselbe ID tragen. Somit wird es für die Sammelstelle schwieriger, die Nachrichten dem entsprechenden Benutzer richtig zuzuordnen.

5.5.1 Generierung einer kritischen Nachricht

Für diesen Ansatz wird die Struktur einer kritischen Nachricht M wie folgt erweitert:

$$M = (\text{time}, \text{ID}, \text{IDpre}, \text{share}, \text{load}).$$

Die Pseudonyme ID und IDpre sind Werte aus der Menge $\{1,\ldots,N\}$ mit $N \in \mathbb{N}$, so dass

$$\text{IDpre}(M) = \begin{cases} \text{ID}(M\text{pre}(M)), & \text{falls es die Nachricht } M\text{pre}(M) \text{ gibt und} \\ r \in \{1,\ldots,N\}, & \text{zufällig gewählt, andernfalls.} \end{cases}$$

Dabei bezeichnet $M\text{pre}(M)$ die kritische Vorgängernachricht von der kritischen Nachricht M, also die kritische Nachricht, die zuletzt vor M von demselben Benutzer initiiert wurde. Das Pseudonym $\text{ID}(M)$ wird für jede kritische Nachricht zufällig aus der Menge $\{1,\ldots,N\}$ gewählt und ist damit im Allgemeinen nicht eindeutig. Die Werte $\text{time}(M)$, $\text{share}(M)$ und $\text{load}(M)$ werden entsprechend dem Schwellwert-Schema aus Kapitel 4 generiert.

5.5.2 Test auf Erreichen des Schwellwertes

Nach einer bestimmten Anzahl von generierten kritischen Nachrichten gibt es mit großer Sicherheit mehrere Nachrichten, die dieselbe ID tragen. Die Vorgänger-ID $\text{IDpre}(M)$ einer kritischen Nachricht M eines Benutzers gibt der Sammelstelle einen Hinweis auf die Vorgängernachricht des Benutzers. Dieser Hinweis ist wie die IDs im Allgemeinen nicht eindeutig und verweist zunächst auf alle potentiellen Vorgängernachrichten, das heißt, auf alle vorher empfangenen kritischen Nachrichten mit entsprechender ID. Die Hinweise kann man als gerichtete Kanten eines Graphen mit der Menge der empfangenen kritischen Nachrichten als Knotenmenge interpretieren. Im Folgenden betrachten wir die von den Vorgänger-IDs erzeugte Graphenstruktur im Detail.

Sei $\mathcal{M}_{[t]}$ die Menge aller kritischen Nachrichten, die von der Sammelstelle bis zur Runde t empfangen wurden. Als *Nachrichtengraph* bezeichnen wir den gerichteten Graphen $G_{[t]} = (\mathcal{M}_{[t]}, E_{[t]})$. Für alle $M_1, M_2 \in \mathcal{M}_{[t]}$ gilt, dass (M_1, M_2) genau dann in $E_{[t]}$ ist, wenn

1. $\text{time}(M_1) > \text{time}(M_2)$ und
2. $\text{IDpre}(M_1) = \text{ID}(M_2)$,

das heißt, M_2 ist ein potentieller Vorgänger von M_1. Dabei gehen wir in diesem Abschnitt wieder davon aus, dass ein Benutzer höchstens eine kritische Aktion je Runde durchführt.

Der Graph $G_{[t]}$ ist ein azyklisch gerichteter Graph. Für jede kritische Nachricht M ist die auf kritische Nachrichten eingeschränkte Historie von M ein Pfad in $G_{[t]}$, der mit M beginnt. Will die Sammelstelle überprüfen, ob mit einer kritischen Nachricht M der entsprechende Benutzer den Schwellwert d an kritischen Nachrichten innerhalb eines Zeitabschnitts erreicht hat und sich seine Identität sowie der Entschlüsselungsschlüssel rekonstruieren lassen, benötigt die Sammelstelle die entsprechende Teilhistorie von M aus dem aktuellen Zeitabschnitt. Jeder Pfad in $G_{[t]}$ der Länge d, der mit M beginnt und nur Nachrichten der letzten Δ Runden enthält, ist ein *Kandidat* für die Teilhistorie von M. Sei $C_{[t]}(M)$ die Menge aller Kandidaten für die Teilhistorie der Nachricht M. Für jeden Pfad $\pi = M_1,\ldots,M_{d-1} \in C_{[t]}(M)$ in $G_{[t]}$ kann die Sammelstelle mit folgendem Algorithmus überprüfen, ob dieser der Teilhistorie von M entspricht:

Algorithmus $\text{Test}_{G_{[t]}}(\pi, M)$

1. Für einen Pfad $\pi = M_1, \ldots, M_{d-1}$ in $G_{[t]}$ mit $M_1 = M$ seien $\widehat{\mathcal{I}}$ der Identitätsstring und $\widehat{S}$ der Schlüssel, die aus den Shares in π rekonstruiert wurden.
2. Für jede Nachricht M_τ in π berechnet die Sammelstelle aus $\widehat{S}$ die Schlüssel $\widehat{K}_\tau$ und $\widehat{L}_\tau$ zur Entschlüsselung von $\text{load}(M_\tau)$ gemäß der Entschlüsselungsphase des Schwellwert-Schemas.
3. Mithilfe des Identitätsstrings $\widehat{\mathcal{I}}$ und den Schlüsseln $\widehat{K}_\tau$ und $\widehat{L}_\tau$ verifiziert die Sammelstelle die Korrektheit des Fingerabdrucks von M_τ und der in $\text{load}(M_\tau)$ verschlüsselten Identität.
4. Schlägt diese Verifikation für eine der Nachrichten in π fehl, so entspricht π nicht der Teilhistorie von M und wird verworfen. Kann jede Nachricht aus π verifiziert werden, so akzeptiert die Sammelstelle π als Teilhistorie von M und $\widehat{\mathcal{I}}$ als Identität des Benutzers von M.

Lemma 5.6 *Sei $M \in \mathcal{M}_{[t]}$ und $\pi = M_1, \ldots, M_{d-1} \in C_{[t]}(M)$. Dann akzeptiert die Sammelstelle den Pfad π nach Durchführung von $\text{Test}_{G_{[t]}}(\pi, M)$ mit großer Sicherheit genau dann, wenn π ein Teil der Historie von M ist.*

BEWEIS Führt die Sammelstelle diesen Test für einen Pfad π durch, der ein Teil der Historie von M ist, dann akzeptiert sie π, da aus den Shares der Nachrichten in π die Identität des Benutzers sowie alle Schlüssel korrekt bestimmt werden können und jede Nachricht in π über ihren Fingerabdruck korrekt verifiziert werden kann.

Nach Lemma 4.9 tritt für jede Nachricht mit großer Sicherheit keine Kollision des Fingerabdrucks auf. Daher schlägt mit großer Sicherheit für jede Nachricht die Verifikation mit einem anderen Schlüssel oder mit einem falschen Identitätsstring fehl. Daraus folgt für einen Pfad π, der nicht der Teilhistorie von M entspricht, dass mit großer Sicherheit für mindestens eine Nachricht in π die Verifikation des Fingerabdrucks mit der Identität $\widehat{\mathcal{I}}$ fehlschlägt und π somit nicht von der Sammelstelle als Teilhistorie akzeptiert wird. ∎

Zur Prüfung auf Erreichen des Schwellwertes führt die Sammelstelle für alle Kandidaten $\pi \in C_{[t]}(M)$ den Algorithmus $\text{Test}_{G_{[t]}}(\pi, M)$ durch. Akzeptiert die Sammelstelle einen Pfad, dann hat der Benutzer mit der Nachricht M den Schwellwert d von kritischen Aktionen innerhalb eines Zeitabschnitts überschritten.

5.5.3 Effizienzanalyse des Tests auf Erreichen des Schwellwertes

Die Laufzeit, um das Erreichen des Schwellwertes durch eine Nachricht M festzustellen, wächst im Worst-Case linear mit der Anzahl der unterschiedlichen Pfade π aus $C_{[t]}(M)$. Damit wächst die Anzahl der potentiellen Kandidaten für die Teilhistorie von M. Wir wollen im Folgenden die Anzahl dieser Pfade untersuchen.

Sei m_t die Anzahl aller Nachrichten M mit $\text{time}(M) = t$, sowie $m_{\min} = \min_t m_t$ und $m_{\max} = \max_t m_t$. Im Folgenden nehmen wir an, dass in jeder Runde jeder der n Benutzer mit Wahrscheinlichkeit p_{cm} eine kritische Aktion durchführt.

Lemma 5.7 *Sei $z \leq \frac{p_{\text{cm}} n + 12}{12 \ln(N)}$, dann gilt:*

$$\Pr\left[\frac{1}{2} \cdot p_{\text{cm}} n \leq m_t \leq \frac{3}{2} \cdot p_{\text{cm}} n\right] \geq 1 - N^{-z}.$$

BEWEIS Die Zufallsvariable m_t ist binomialverteilt mit den Parametern p_{cm} und n und Erwartungswert $p_{\text{cm}}n$. Unter Verwendung der Chernoff-Schranken gilt:

$$\Pr\left[m_t \leq \frac{1}{2}\cdot p_{\text{cm}}n\right] \leq e^{-\frac{p_{\text{cm}}n}{8}} \quad \text{und}$$
$$\Pr\left[m_t \geq \frac{3}{2}\cdot p_{\text{cm}}n\right] \leq e^{-\frac{p_{\text{cm}}n}{12}}.$$

Daraus folgt, dass

$$\begin{aligned}
&\Pr\left[\frac{1}{2}\cdot p_{\text{cm}}n \leq m_t \leq \frac{3}{2}\cdot p_{\text{cm}}n\right] \\
&\geq 1-\left(\Pr\left[m_t \leq \frac{1}{2}\cdot p_{\text{cm}}n\right]+\Pr\left[m_t \geq \frac{3}{2}\cdot p_{\text{cm}}n\right]\right) \\
&\geq 1-\left(e^{-\frac{p_{\text{cm}}n}{8}}+e^{-\frac{p_{\text{cm}}n}{12}}\right) \\
&\geq 1-\left(e\cdot e^{-\frac{p_{\text{cm}}n}{12}}\right) \\
&= 1-\left(e^{-\frac{p_{\text{cm}}n}{12}+1}\right) \\
&= 1-\left(e^{-\frac{\ln(N)}{\ln(N)}\cdot\frac{p_{\text{cm}}n+12}{12}}\right) \\
&\geq 1-N^{-z}.
\end{aligned}$$

■

Wählen wir also $N \in e^{o(p_{\text{cm}}n)} \cap \Omega(p_{\text{cm}}n)$, dann folgt mit großer Sicherheit, dass

$$\frac{1}{2}\cdot p_{\text{cm}}n \leq m_t \leq \frac{3}{2}\cdot p_{\text{cm}}n < N.$$

In der folgenden Analyse wird daher angenommen, dass $m_{\max}$ um höchstens den Faktor 3 größer ist als $m_{\min}$ und dass $m_{\max} \leq N^{\varepsilon}$ für ein $0 < \varepsilon < 1$.

Lemma 5.8 *Sei $\delta > 1$. Für eine Nachricht $M \in \mathcal{M}_{[t]}$ mit $\text{time}(M) = t$ ist mit Wahrscheinlichkeit $1 - e^{-\frac{\delta}{9}\cdot\frac{\Delta-(d-1)}{N^{1-\varepsilon}}+\ln(N)}$ die Anzahl der Kandidaten in $C_{[t]}(M)$ beschränkt durch*

$$\left((1+\delta)\cdot\frac{\Delta-(d-1)}{N^{1-\varepsilon}}\right)^{d-1}.$$

BEWEIS Jeder Kandidat $\pi \in C_{[t]}(M)$ für die Teilhistorie von M ist ein Pfad in $G_{[t]}$, der die Länge d hat und nur Nachrichten aus Δ aufeinander folgenden Runden enthält. Deshalb beträgt der Abstand zwischen zwei Nachrichten in π höchstens $h = \Delta-(d-1)$. Für $t = \text{time}(M)$ sowie alle $i \in \{1,\ldots,N\}$ und $j \in \{0,\ldots,h-1\}$ beschreibe die Zufallsvariable X_i^j die Anzahl der Nachrichten $M' \in \mathcal{M}_{[t]}$ mit $\text{ID}(M') = i$ und $\text{time}(M') = t-j$. Da die ID einer Nachricht zufällig mit Wahrscheinlichkeit $\frac{1}{N}$ gewählt wird, ist X_i^j binomialverteilt mit Parametern m_t und $\frac{1}{N}$. Somit ist die Zufallsvariable $X_i = \sum_{j=0}^{h-1} X_i^j$ binomialverteilt mit den Parametern $\sum_{j=0}^{h-1} m_{t+j}$ und $\frac{1}{N}$. Sie beschreibt die Anzahl der Nachrichten mit derselben ID i, die die Sammelstelle in h aufeinander folgenden Runden empfangen hat.

Falls für jeden Wert i mit großer Sicherheit gilt, dass $X_i \leq c$, dann ist die Anzahl der Kandidaten für die Teilhistorie von M mit großer Sicherheit durch c^{d-1} beschränkt, weil für jeden der $d-1$ Vorgänger in der Teilhistorie von M mit großer Sicherheit höchstens c

Nachrichten entsprechender ID in Frage kommen. Aus $h \cdot m_{\min} \leq \sum_{j=0}^{h-1} m_{t+j} \leq h \cdot m_{\max}$ und $m_{\max} \leq 3 \cdot m_{\min}$, ergibt sich, dass

$$\sum_{j=0}^{h-1} m_{t+j} \geq \frac{h \cdot m_{\max}}{3}.$$

Durch Verwendung der Chernoff-Schranke erhalten wir für $\delta > 1$ und $c = (1+\delta)\frac{h}{N^{1-\varepsilon}}$

$$\begin{aligned}
\Pr[X_i \geq c] &= \Pr\left[X_i \geq (1+\delta)\frac{h}{N^{1-\varepsilon}}\right] \\
&= \Pr\left[X_i \geq (1+\delta)\frac{h \cdot m_{\max}}{N}\right] \\
&\leq \Pr\left[X_i \geq (1+\delta)\frac{\sum_j m_{t+j}}{N}\right] \\
&\leq e^{-\delta \frac{\sum_j m_{t+j}}{3N}} \\
&\leq e^{-\delta \frac{h \cdot m_{\max}}{9N}} \\
&= e^{-\delta \frac{h}{9N^{1-\varepsilon}}}.
\end{aligned}$$

Diese Abschätzung gilt für alle X_i. Daraus folgt:

$$\begin{aligned}
\Pr[\exists i\colon X_i \geq c] &\leq \sum_{i=1}^{N} \Pr[X_i \geq c] \\
&= N \cdot e^{-\delta \frac{h}{9N^{1-\varepsilon}}} \\
&= e^{-\delta \frac{h}{9N^{1-\varepsilon}} + \ln(N)}.
\end{aligned}$$

Folglich ist mit Wahrscheinlichkeit $\Pr[\forall i\colon X_i < c] = 1 - \Pr[\exists i\colon X_i \geq c]$ die Anzahl der Kandidaten für die Teilhistorie von M durch c^{d-1} beschränkt. ∎

Das exponentielle Wachstum der Anzahl der Kandidaten in d und damit auch das Wachstum der Laufzeit zur Prüfung auf Erreichen des Schwellwertes folgt aus der Struktur von $G_{[t]}$. Bei wachsendem Wert von N reduziert sich die Laufzeit, da sich durch die Erhöhung dieses Wertes die IDs auf eine größere Anzahl von Werten verteilen. Bleibt die Anzahl von Nachrichten je Runde und damit auch $m_{\max}$ etwa gleich, dann reduziert sich zudem der Wert ε entsprechend.

5.5.4 Analyse der Vermischung der Historien von Benutzern

In diesem Abschnitt betrachten wir die Menge $\widetilde{\mathcal{B}}_t$ von Benutzern, die bis zur Runde t in keinem Zeitabschnitt d kritische Aktionen durchgeführt haben und für die die Sammelstelle somit aus den entsprechenden Shares weder Identität noch Schlüssel rekonstruieren konnte. Wir untersuchen im Folgenden die Frage, ab welchem Zeitpunkt in der Vergangenheit für eine kritische Nachricht M eines Benutzers in $\widetilde{\mathcal{B}}_t$ alle von der Sammelstelle empfangenen Nachrichten der Benutzer aus $\widetilde{\mathcal{B}}_t$ mögliche Vorgänger von M sind. Mindestens vor diesem Zeitpunkt ist die Historie des Benutzers von M nicht mehr von den Historien der anderen „unerkannten“ Benutzer zu unterscheiden.

Mit $\widetilde{\mathcal{M}}_{[t]}$ bezeichnen wir die kritischen Nachrichten M von Benutzern aus $\widetilde{\mathcal{B}}_t$ mit $\text{time}(M) \leq t$. Sei $\widetilde{G}_{[t]}$ der von $\widetilde{\mathcal{M}}_{[t]}$ induzierte Subgraph des Nachrichtengraphen $G_{[t]}$. Sei

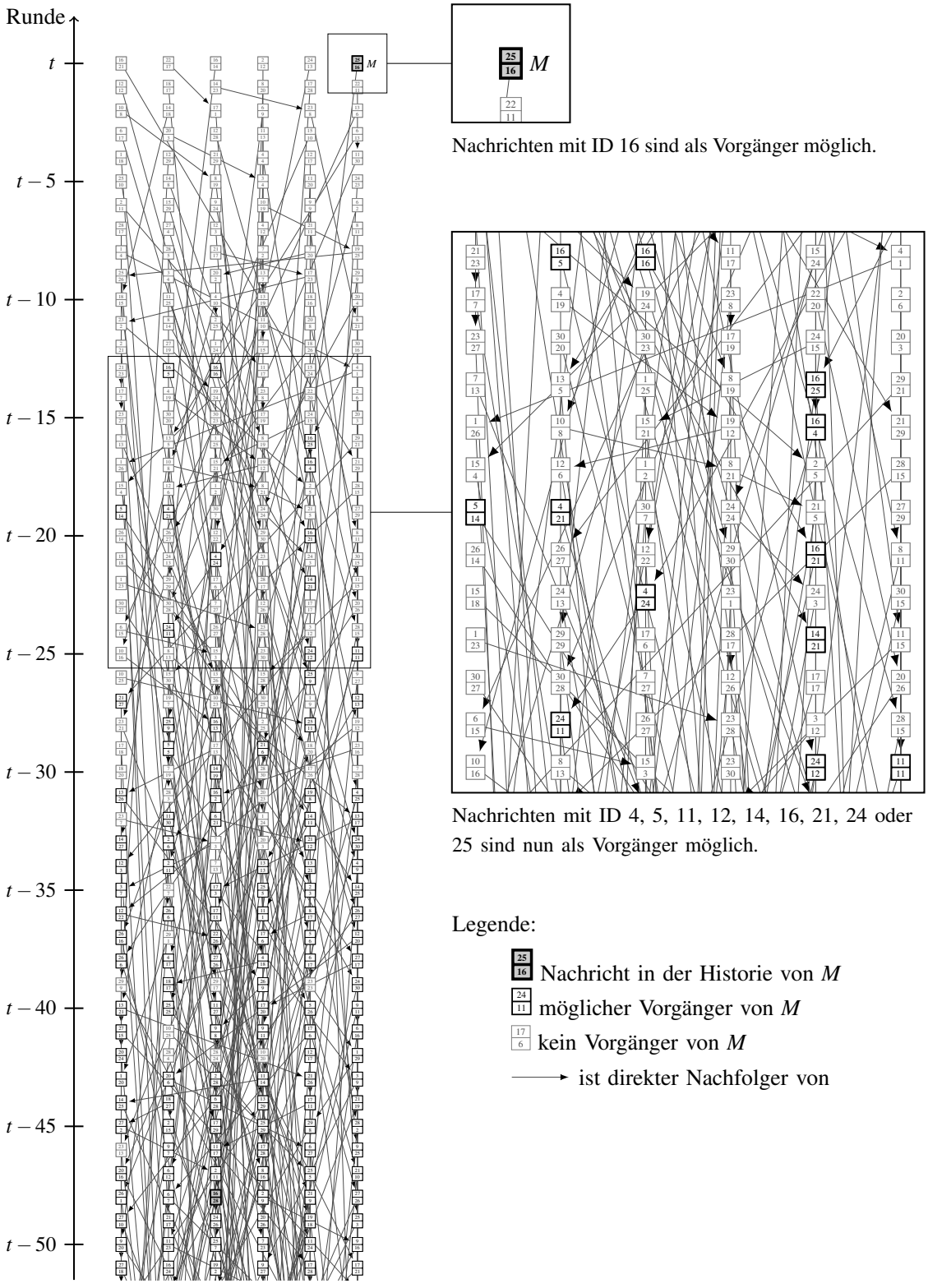


Abbildung 5.2: Darstellung der möglichen Vorgänger der Nachricht $M \in \mathcal{M}_{[t]}$ mit $t = \text{time}(M)$ in einem zufälligen Graphen $\widetilde{G}_t$ bei $N = 30$ IDs und $|\widetilde{\mathcal{B}}_t| = 150$. Alle Nachrichten vor Runde $t - 50$ sind mögliche Vorgänger von M. Zur besseren Übersicht wurde die Darstellung auf $\widetilde{m}_{\min} = 6$ Nachrichten je Runde eingeschränkt.

Runde	Anzahl IDs
t	1
$t-10$	1
$t-15$	2
$t-20$	6
$t-25$	9
$t-30$	14
$t-35$	21
$t-40$	25
$t-45$	28
$t-50$	30

Tabelle 5.1: Anzahl möglicher IDs von Vorgängern der Nachricht M aus Abbildung 5.2

$\widetilde{m}_t$ die Anzahl von kritischen Nachrichten M von Benutzern aus $\widetilde{\mathcal{B}}_t$ mit $t = \text{time}(M)$ und $\widetilde{m}_{\min} = \min_t \widetilde{m}_t$.

Gibt es für eine Nachricht $M \in \widetilde{\mathcal{M}}_{[t]}$ zwei oder mehr Nachrichten M_1 und M_2 in $\widetilde{\mathcal{M}}_{[t]}$ mit $\text{time}(M_1), \text{time}(M_2) < \text{time}(M)$ und $\text{ID}(M_1) = \text{ID}(M_2) = \text{IDpre}(M)$, dann ist für die Sammelstelle der Vorgänger von M in der Historie von M nicht eindeutig bestimmbar. Mit großer Wahrscheinlichkeit gibt es zudem mindestens zwei solcher potentiellen Vorgänger M_i und M_j, so dass $\text{IDpre}(M_i) \neq \text{IDpre}(M_j)$. Betrachten wir die Historien der Benutzer aus $\widetilde{\mathcal{B}}_t$ gemeinsam, vermischen sich diese Historien mit Blick in die Vergangenheit immer weiter. Dies deuten auch die Ergebnisse aus dem vorherigen Abschnitt an. Dort nimmt die Anzahl der Kandidaten ebenfalls mit der betrachteten Länge d der Historie zu.

In Abbildung 5.2 ist die Vermischung der Nachricht M eines Benutzers aus $\widetilde{\mathcal{B}}_t$ in einem zufällig generierten Graphen $\widetilde{G}_{[t]}$ zu sehen. Da die Sammelstelle für die Nachrichten die Verbindungen nicht kennt, nimmt die Anzahl der möglichen Vorgänger in der Historie nach und nach zu. Der Tabelle 5.1 können wir entnehmen, dass in Runde $t-50$ alle $N = 30$ IDs mögliche IDs von Vorgängern von M sind. Damit sind alle Nachrichten, die vor Runde $t-50$ empfangen wurden, in Bezug auf M mögliche Vorgänger. Die Sammelstelle kann somit kein Wissen über die Historie von M aus einer Runde $\leq t-50$ herleiten.

Im Weiteren untersuchen wir den Grad der Vermischung als eine Funktion über die Zeit. Dazu bezeichne $I(t', M)$ die Menge aller Vorgänger-IDs $\text{IDpre}(M')$, so dass es einen Pfad π in $\widetilde{G}_{[t]}$ gibt, auf dem $M \in \widetilde{\mathcal{M}}_{[t]}$ und $M' \in \widetilde{\mathcal{M}}_{[t]}$ mit $t = \text{time}(M) > \text{time}(M') \geq t'$ liegen, und dass der Zeitabstand zwischen zwei in π aufeinander folgenden Nachrichten mindestens $\frac{\Delta}{d-1}$ ist. Durch die Einführung des Mindestabstandes $\frac{\Delta}{d-1}$ bei der Definition der Mengen $I(\cdot, M)$ wird sichergestellt, dass kein Pfad zur Vergrößerung von $I(\cdot, M)$ im zeitlichen Verlauf beiträgt, der mehr als $d-1$ kritische Nachrichten in Δ aufeinander folgenden Runden enthält. Dadurch tragen Kandidaten aus $C_{[t]}(M)$ und damit Pfade, deren Shares eine Rekonstruktion zulassen, nicht zur Erhöhung von $I(\cdot, M)$ bei. Für eine Nachricht M eines Benutzers aus $\widetilde{\mathcal{B}}_t$ beschreibt die Menge $I(t', M)$ eine Teilmenge der IDs, so dass alle Nachrichten, die vor Runde t' generiert wurden und eine ID aus $I(t', M)$ tragen, potentielle Vorgänger in der Historie der Nachricht M sind. Wir werden nun einen Wert für t' bestimmen, so dass mit großer Sicherheit alle IDs in diese Menge fallen, das heißt, $|I(t', M)| = N$. Damit sind mit großer Sicherheit alle Nachrichten, die vor Runde t' generiert wurden, potentielle Vorgänger in der Historie von M. Somit gibt es kein Wissen über die Historie der Nachricht M in den Runden vor t'.

Fakt 5.9 *Da $\widetilde{G}_{[h]}$ gerichtet und azyklisch ist, gilt nach der Definition der Mengen $I(\cdot,M)$ mit $M \in \widetilde{\mathcal{M}}_{[t]}$, dass*

1. $I(\text{time}(M),M) = \{\text{IDpre}(M)\}$ *und*
2. *für alle $t' > \text{time}(M)$, dass $I(t',M) = \emptyset$.*
3. *Zudem gilt für alle $t' \leq \text{time}(M) - \left\lceil \frac{\Delta}{d-1} \right\rceil$, dass $i \in I(t',M) \setminus I(t'+1,M)$ genau dann, wenn es mindestens eine kritische Nachricht $M' \in \widetilde{\mathcal{M}}_{[\text{time}(M)]}$ gibt, so dass $\text{IDpre}(M') = i$ und $\text{ID}(M') \in I\left(t' + \left\lceil \frac{\Delta}{d-1} \right\rceil, M\right)$.*

Zum besseren Verständnis für den Rest dieses Abschnitts folgt eine Übersicht über die nächsten Schritte der Analyse.

1. In Lemma 5.10 untersuchen wir, wie viele Schritte ζ in die Vergangenheit nötig sind, um ausgehend von $|I(\text{time}(M),M)| = 1$ den Wert $|I(\text{time}(M) - \zeta, M)| \geq \widetilde{m}_{\min}$ zu erreichen.
2. Die Anzahl der Schritte (Wartezeit) ξ, bis sich $I(\cdot,M)$ um mindestens $\widetilde{m}_{\min} - 1$ IDs vergrößert hat, untersuchen wir im Lemma 5.11. Dabei wird davon ausgegangen, dass die Menge zu Beginn mindestens $\widetilde{m}_{\min}$ IDs enthält.
3. Warten wir für mindestens $\left(\left\lceil \frac{N}{\widetilde{m}_{\min}-1} \right\rceil\right)$-mal $\rho \geq \max(\zeta,\xi)$ Schritte, dann erhalten wir mit großer Sicherheit, dass $|I(\text{time}(M) - \left\lceil \frac{N}{\widetilde{m}_{\min}-1} \right\rceil \cdot \rho, M)| \geq N$ (Theoreme 5.12 und 5.13).

Im Folgenden beziehen wir uns bei Nachrichten, Benutzern, den entsprechenden Historien und dem Nachrichtengraphen immer auf die Mengen $\widetilde{\mathcal{M}}_{[\cdot]}$, $\widetilde{\mathcal{B}}_t$ sowie $\widetilde{G}_{[\cdot]}$.

Lemma 5.10 *Sei $N > \widetilde{m}_{\min}$ und*

$$\zeta(k) = k \cdot \frac{N^2 \cdot \ln N}{N - \widetilde{m}_{\min} + 1} + \frac{\Delta}{d-1}.$$

Für jede kritische Nachricht $M \in \mathcal{M}_{[t]}$, $k > 1$ und $t' \leq \text{time}(M) - \zeta(k)$ gilt mit einer Wahrscheinlichkeit von mindestens $1 - N^{-(k-1)}$, dass

$$|I(t',M)| \geq \tilde{m}_{\min}.$$

BEWEIS Seien B der Benutzer der Nachricht M und $c = \frac{\Delta}{d-1}$. Um den Mindestabstand c der Pfade in der Definition der Mengen $I(\cdot,M)$ einzuhalten, werden wir für $\ell \in \mathbb{N}$ die Runden $r \in \{\text{time}(M) - c, \ldots, \text{time}(M) - c - \ell\}$ untersuchen. Sei $M_{r,1}, \ldots, M_{r,\widetilde{m}_{\min}}$ eine Folge von $\widetilde{m}_{\min}$ Nachrichten mit $\text{time}(M_{r,i}) = r$ für $i \in \{1, \ldots, \widetilde{m}_{\min}\}$. Falls für den Benutzer B in der Runde r eine Nachricht M' vorkommt, so betrachten wir ohne Einschränkung der Allgemeinheit eine solche Folge mit $M' = M_{r,\widetilde{m}_{\min}}$. Man beachte, dass für jeden Benutzer höchstens eine Nachricht je Runde vorkommen kann. Zudem bezeichne

$$M_{\bullet,i} = M_{r,i}|_{r \in \{\text{time}(M)-c,\ldots,\text{time}(M)-c-\ell\}}$$

die Folge der jeweils i-ten Nachrichten $M_{r,1}, \ldots, M_{r,\widetilde{m}_{\min}}$. $M_{\bullet,i}$ nennen wir auch die i-te Spalte der Nachrichtenfolgen. In den Spalten 1 bis $\widetilde{m}_{\min} - 1$ befinden sich somit nur Nachrichten von anderen Benutzern als B.

Das Ziel ist nun, einen Wert ℓ zu bestimmen, so dass wir mit großer Sicherheit in jeder Spalte $M_{\bullet,i}$ mit $i \in \{1,\ldots,\widetilde{m}_{\min}-1\}$ eine Nachricht M_i finden, die ein potentieller Vorgänger von M ist, aber alle M_i unterschiedliche Vorgänger-IDs IDpre haben. Dann gilt für alle $i \neq j \in \{1,\ldots,\widetilde{m}_{\min}-1\}$:

1. $\mathrm{ID}(M_i) = \mathrm{IDpre}(M)$,
2. $\mathrm{IDpre}(M_i) \neq \mathrm{IDpre}(M)$,
3. $\mathrm{IDpre}(M_i) \neq \mathrm{IDpre}(M_j)$.

Für einen solchen Wert ℓ gilt somit auch $|I(\mathrm{time}(M)-c-\ell,M)| \geq 1+(\widetilde{m}_{\min}-1) = \widetilde{m}_{\min}$.

Da die IDs der Nachrichten der Benutzer unabhängig voneinander gewählt werden, gilt für eine Nachricht M' in $M_{\bullet,1}$ mit Wahrscheinlichkeit

$$\frac{1}{N}\cdot\frac{N-1}{N} = \frac{N-1}{N^2},$$

dass $\mathrm{ID}(M') = \mathrm{IDpre}(M)$ und $\mathrm{IDpre}(M') \neq \mathrm{IDpre}(M)$. Mit Wahrscheinlichkeit

$$\frac{N-1}{N^2}\cdot\left(1-\frac{N-1}{N^2}\right)^{j_1-1}$$

müssen wir die ersten j_1 Elemente von $M_{\bullet,1}$ durchsuchen, bevor wir eine passende Nachricht M_1 mit $\mathrm{ID}(M_1) = \mathrm{IDpre}(M)$ und $\mathrm{IDpre}(M_1) \neq \mathrm{IDpre}(M)$ finden.

Wir betrachten die Spalten $M_{\bullet,i}$ für $i = 1,\ldots,\widetilde{m}_{\min}-1$ nacheinander. Angenommen, die Nachrichten $M_1,\ldots,M_{i-1}$ wurden bereits bestimmt. Sei J_i die Zufallsvariable, die der Anzahl an Nachrichten in $M_{\bullet,i}$ entspricht, die wir durchsuchen müssen, bevor wir eine passende Nachricht M_i finden, das heißt,

1. $\mathrm{ID}(M_i) = \mathrm{IDpre}(M)$,
2. $\mathrm{IDpre}(M_i) \neq \mathrm{IDpre}(M)$,
3. $\mathrm{IDpre}(M_i) \notin \{\mathrm{IDpre}(M_1),\ldots,\mathrm{IDpre}(M_{i-1})\}$.

Dann gilt für jedes $j \in \mathbb{N}$:

$$\Pr[J_i = j] = \frac{N-i}{N^2}\cdot\left(1-\frac{N-i}{N^2}\right)^{j-1} \quad \text{und}$$
$$\Pr[J_i > j] = \left(1-\frac{N-i}{N^2}\right)^{j} \leq e^{-\frac{N-i}{N^2}\cdot j}.$$

Für jeden Wert $j \in \mathbb{N}$ ist die Wahrscheinlichkeit $\Pr[J_i > j]$ monoton steigend in i. Somit reicht es aus, einen Wert für j zu finden, so dass $\Pr[J_{\widetilde{m}_{\min}-1} > j]$ hinreichend klein ist.

Wenn wir $\ell = k\cdot\frac{N^2\cdot\ln N}{N-\widetilde{m}_{\min}+1}$ wählen, dann gilt für alle $i \in \{1,\ldots,\widetilde{m}_{\min}-1\}$, dass

$$\begin{aligned}\Pr[J_i > \ell] &\leq \Pr[J_{\widetilde{m}_{\min}-1} > \ell] \\ &\leq e^{-\frac{N-(\widetilde{m}_{\min}-1)}{N^2}\cdot\ell} \\ &= e^{-\frac{N-\widetilde{m}_{\min}+1}{N^2}\cdot k\cdot\frac{N^2\cdot\ln N}{N-\widetilde{m}_{\min}+1}} \\ &= N^{-k}.\end{aligned}$$

Daraus folgt, dass

$$\begin{aligned}\Pr[\exists i \in \{1,\ldots,\widetilde{m}_{\min}-1\}: J_i > \ell] &\leq \sum_{i=1}^{\widetilde{m}_{\min}-1} \Pr[J_i > \ell] \\ &\leq N \cdot N^{-k} = N^{-(k-1)}\end{aligned}$$

Somit fügen wir mit einer Wahrscheinlichkeit von $1-N^{-(k-1)}$ mindestens $\widetilde{m}_{\min}-1$ IDs zu $I(\text{time}(M),M)$ in $\ell + \frac{\Delta}{d-1} = \zeta(k)$ Runden hinzu. ∎

Wir erweitern das vorherige Lemma:

Lemma 5.11 *Sei $N > \widetilde{m}_{\min}$ und*

$$\xi(k) = k \cdot \frac{N^2 \cdot \ln N}{m(N-m-\widetilde{m}_{\min})} + \frac{\Delta}{d-1}.$$

Für jede kritische Nachricht $M \in \mathcal{M}_{[t]}$, für $k > 1$, $s,m \in \mathbb{N}$ und $t' \leq s - \xi(k)$ gilt: Aus $|I(s,M)| \geq m$ und $m + \widetilde{m}_{\min} - 1 \leq N$ folgt, dass mit einer Wahrscheinlichkeit von mindestens $1-N^{-(k-1)}$ gilt:

$$|I(t',M)| \geq m + \widetilde{m}_{\min} - 1.$$

BEWEIS Wie im vorherigen Beweis sei B der Benutzer der Nachricht M und $c = \frac{\Delta}{d-1}$. Zudem seien für $\ell \in \mathbb{N}$ und $r \in \{\text{time}(M)-c,\ldots,\text{time}(M)-c-\ell\}$ die Nachrichtenfolgen $M_{r,1},\ldots,M_{r,\widetilde{m}_{\min}}$ sowie $M_{\bullet,i}$ analog definiert. Falls für den Benutzer B in der Runde r eine Nachricht M' vorkommt, so betrachten wir ohne Einschränkung der Allgemeinheit eine solche Folge mit $M' = M_{r,\widetilde{m}_{\min}}$.

Das Ziel ist nun, einen Wert ℓ zu bestimmen, so dass wir mit großer Sicherheit in jeder Spalte $M_{\bullet,i}$ mit $i \in \{1,\ldots,\widetilde{m}_{\min}-1\}$ eine Nachricht M_i finden, die ein potentieller Vorgänger von M ist, aber alle M_i unterschiedliche Vorgänger-IDs IDpre haben. Für alle $i \neq j \in \{1,\ldots,\widetilde{m}_{\min}-1\}$ gilt somit:

1. $\text{ID}(M_i) \in I(s,M)$,
2. $\text{IDpre}(M_i) \notin I(s,M)$,
3. $\text{IDpre}(M_i) \neq \text{IDpre}(M_j)$.

Für einen solchen Wert ℓ gilt somit auch $|I(s-c-\ell,M)| \geq m + \widetilde{m}_{\min} - 1$.

Da die IDs der Nachrichten der Benutzer unabhängig voneinander gewählt werden, gilt für eine Nachricht M' in $M_{\bullet,1}$ mit Wahrscheinlichkeit:

$$\frac{m}{N} \cdot \frac{N-m}{N} = \frac{m(N-m)}{N^2},$$

dass $\text{ID}(M') \in I(s,M)$ und $\text{IDpre}(M') \notin I(s,M)$. Mit Wahrscheinlichkeit

$$\frac{m(N-m)}{N^2} \cdot \left(1 - \frac{m(N-m)}{N^2}\right)^{j_1-1}$$

müssen wir die ersten j_1 Elemente von $M_{\bullet,1}$ durchsuchen, bevor wir eine passende Nachricht M_1 mit $\text{ID}(M_1) \in I(s,M)$ und $\text{IDpre}(M_1) \notin I(s,M)$ finden.

Wir betrachten die Spalten $M_{\bullet,i}$ für $i = 1,\ldots,\widetilde{m}_{\min}-1$ nacheinander. Angenommen die Nachrichten $M_1,\ldots,M_{i-1}$ wurden bereits bestimmt. Sei J_i die Zufallsvariable, die der Anzahl an Nachrichten in $M_{\bullet,i}$ entspricht, die wir durchsuchen müssen, bevor wir eine passende Nachricht M_i finden, das heißt,

1. $\mathrm{ID}(M_i) \in I(s,M)$,
2. $\mathrm{IDpre}(M_i) \notin I(s,M)$,
3. $\mathrm{IDpre}(M_i) \notin \{\mathrm{IDpre}(M_1),\ldots,\mathrm{IDpre}(M_{i-1})\}$.

Dann gilt für jedes $j \in \mathbb{N}$:

$$\begin{aligned}\Pr[J_i = j] &= \frac{m(N-m-(i-1))}{N^2} \cdot \left(1 - \frac{m(N-m-(i-1))}{N^2}\right)^{j-1} \\ &= \frac{m(N-m-i+1)}{N^2} \cdot \left(1 - \frac{m(N-m-i+1)}{N^2}\right)^{j-1} \quad \text{und} \\ \Pr[J_i > j] &= \left(1 - \frac{m(N-m-i+1)}{N^2}\right)^{j} \leq e^{-\frac{m(N-m-i+1)}{N^2}\cdot j}.\end{aligned}$$

Für jeden Wert $j \in \mathbb{N}$ ist die Wahrscheinlichkeit $\Pr[J_i > j]$ monoton steigend in i. Somit ist es ausreichend, einen Wert für j zu finden, so dass $\Pr[J_{\widetilde{m}_{\min}-1} > j]$ hinreichend klein ist.

Wenn wir $\ell = k \cdot \frac{N^2 \cdot \ln N}{m(N-m-\widetilde{m}_{\min})}$ wählen, dann gilt für alle $i \in \{1,\ldots,\widetilde{m}_{\min}-1\}$:

$$\begin{aligned}\Pr[J_i > \ell] &\leq \Pr[J_{\widetilde{m}_{\min}-1} > \ell] \\ &\leq e^{-\frac{m(N-m-(\widetilde{m}_{\min}-1)+1)}{N^2}\cdot \ell} \\ &= e^{-\frac{N-m-\widetilde{m}_{\min}}{N^2}\cdot k \cdot \frac{N^2\cdot \ln N}{N-m-\widetilde{m}_{\min}}} \\ &= N^{-k}.\end{aligned}$$

Somit fügen wir mit einer Wahrscheinlichkeit von $1 - N^{-(k-1)}$ mindestens $\widetilde{m}_{\min} - 1$ IDs zu $I(s,M)$ in $\ell + \frac{\Delta}{d-1} = \xi(k)$ Runden hinzu. ∎

Mit großer Sicherheit nimmt also die Größe der Menge $I(\cdot,M)$ um den Wert $\widetilde{m}_{\min} - 1$ in jeweils $\xi(k)$ Runden zu. Folglich benötigen wir $\mathcal{O}\left(\frac{N}{\widetilde{m}_{\min}-1}\right)$ Iterationen dieser Strategie, bis $I(\cdot,M)$ einen konstanten Teil aller IDs $\{1,\ldots,N\}$ enthält. Der Wert m aus dem vorherigen Lemma hat einen Wert im Bereich von $\{1,\ldots,N-\widetilde{m}_{\min}+1\}$. Damit ist die Parabel $m(N-m-\widetilde{m}_{\min})$ in Abhängigkeit von m maximal für $m = \frac{N-\widetilde{m}_{\min}}{2}$ und hat in $m = 1$ und $m = N - \widetilde{m}_{\min} + 1$ ihre Randminima. Daraus können wir schließen, dass für $m \in \{1,\ldots,N-\widetilde{m}_{\min}+1\}$

$$m(N-m-\widetilde{m}_{\min}) \geq N - \widetilde{m}_{\min} - 1$$

gilt. Somit folgt für $k > 0$ und

$$\rho(k) = k \cdot \frac{N^2 \ln N}{N - \widetilde{m}_{\min} - 1} + \frac{\Delta}{d-1},$$

dass

$$\xi(k) \leq \rho(k) \quad \text{und} \quad \zeta(k) \leq \rho(k).$$

Wir erhalten

Theorem 5.12 *Für jede kritische Nachricht $M \in \mathcal{M}_{[t]}$, für $k > 2$ und*

$$t' \leq \mathrm{time}(M) - \left\lceil \frac{N}{\widetilde{m}_{\min} - 1} \right\rceil \cdot \rho(k)$$

gilt mit Wahrscheinlichkeit von mindestens $1 - N^{-(k-2)}$, dass

$$|I(t',M)| = N.$$

BEWEIS Aus den obigen Überlegungen und der Tatsache $\left\lceil \frac{N}{\widetilde{m}_{\min}-1} \right\rceil \leq N$ folgt für die Wahrscheinlichkeit, dass für ein $i \in \{1,\ldots,\left\lceil \frac{N}{\widetilde{m}_{\min}-1} \right\rceil - 1\}$ die Menge $I(\text{time}(M) - i \cdot \rho(k))$ nicht um $\widetilde{m}_{\min} - 1$ IDs innerhalb von $\xi(k)$ Runden wächst, kleiner oder gleich $N \cdot N^{-(k-1)} = N^{-(k-2)}$ ist. Somit gilt die Aussage. ∎

Da die Nachrichten höchstens eine von N verschiedenen IDs tragen können und die ID einer Nachricht uniform gewählt wird, gilt auch:

Theorem 5.13 *Für jeden genügend großen Wert T, für $k > 3$ und*

$$t' \leq T - \left\lceil \frac{N}{\widetilde{m}_{\min} - 1} \right\rceil \cdot \rho(k)$$

gilt gleichzeitig für alle Nachrichten M mit $T \leq \text{time}(M)$ mit einer Wahrscheinlichkeit von mindestens $1 - N^{-(k-3)}$, dass

$$|I(t', M)| = N.$$

Ist beispielsweise $\widetilde{m}_{\min} = N^{\tilde{\varepsilon}}$ mit $\tilde{\varepsilon} < 1$, dann ist mit großer Sicherheit jede vor $M \in \widetilde{\mathcal{M}}_{[t]}$ empfangene Nachricht mit einem zeitlichen Abstand von $\mathcal{O}\left(N^{2-\tilde{\varepsilon}} \cdot \ln(N)\right)$, ein möglicher Vorgänger in der Historie von M. Spätestens mit diesem Abstand kann die Sammelstelle mit großer Sicherheit keine Nachricht mehr der Historie eines Benutzers in $\widetilde{\mathcal{B}}_t$ zuordnen. Damit sind diese Nachrichten anonym.

Der Parameter N steuert den Trade-Off zwischen der Effizienz der Rekonstruktion und dem Grad der Anonymisierung der Nachrichtenhistorien. Ist N groß, dann ist die Wahrscheinlichkeit für die Wahl jeder ID klein und die Anzahl der möglichen Vorgänger im Verhältnis zur Zeit groß. Damit sinkt die Anzahl der zu überprüfenden Kandidatenpfade in dem Nachrichtengraphen. Gleichzeitig verlängert sich die Zeit, bis alle N Werte mögliche IDs eines Vorgängers jeder Nachricht sind. Bei kleinen Werten von N verhält es sich genau umgekehrt. Die Komplexität zur Rekonstruktion steigt, während die Anonymisierung schneller erreicht wird.

Sind nur wenige Benutzer identifiziert und somit $|\widetilde{\mathcal{B}}_t|$ und $\widetilde{m}_{\min}$ groß, dann vermischen sich die Historien der Benutzer schneller. Ein ähnliches Verhalten konnten wir bei der einfachen Erweiterung des Schwellwert-Schemas in Abschnitt 5.3 beobachten. Gibt es viele unidentifizierte Benutzer, dann erhöht sich auch die Anzahl der Shares, die die Sammelstelle je Runde erhält und damit die Komplexität des Tests auf Erreichen des Schwellwertes.

6

Entkopplung der Historie durch zwei separate Nachrichtenlinien

Der Schutz der Historie ist bei den Ansätzen des vorherigen Kapitels (mit Ausnahme des Abschnitts 5.4) stark abhängig von der Anzahl der unidentifizierten Benutzer. Die richtige Zuordnung der Nachrichten zu den Benutzern ist umso leichter, je weniger Benutzer existieren. Darüber hinaus wird der Schutz der Historie durch eine Lockerung der Schwellwertbedingung und bei der Vermischung der Historien durch eine erhöhte Laufzeit der Rekonstruktion erkauft (Abschnitte 5.4 und 5.5).

In diesem Kapitel widmen wir uns einem probabilistischen Ansatz, der sich durch eine effiziente Rekonstruktion bei Einhaltung des Schwellwertes und durch eine Anonymisierung der Nachrichten auszeichnet, die weitgehend unabhängig von der Benutzerzahl ist. Dieser Ansatz kann zudem in anderen Einsatzgebieten als der Vorratsdatenspeicherung verwendet werden, beispielsweise zur Stärkung der Privatheit bei der Erstellung von Kundenprofilen im Marketing.

Das Protokoll für diesen Ansatz baut auf einer Graphenstruktur mit zwei separaten Nachrichtenlinien auf. Anstatt einer kritischen Nachricht pro kritischer Aktion erzeugt der Provider nun zwei. Von diesen Nachrichten erhält die Sammelstelle nur eine zufällig ausgewählte. Eine Nachricht in der Graphenstruktur verweist jeweils auf k vorhergehende Nachrichten. Die *Entkopplung der Historie* entspricht einem Zerfall in mehrere Zusammenhangskomponenten der Graphenstruktur. Bei einer Entkopplung zum Zeitpunkt τ liegen alle Nachrichten vor dem Zeitpunkt τ in anderen Zusammenhangskomponenten als die Nachrichten zum und nach dem Zeitpunkt τ. In der Analyse des Graphen werden wir zeigen, unter welchen Umständen der Graph zerfällt und wann mit dem Zerfall zu rechnen ist. Wir können das Protokoll ebenfalls in anderen Szenarien einsetzen und unter Umständen in der Vorratsdatenspeicherung sogar auf den Provider verzichten.

Zunächst wird im folgenden Abschnitt die zugrundeliegende Graphenstruktur beschrieben und charakterisiert, wann eine Entkopplung der Historie stattfindet. In Abschnitt 6.2 wird die Struktur der Nachrichten sowie das Protokoll zwischen Benutzer und Sammelstelle vorgestellt. Anschließend untersuchen wir in den Abschnitten 6.3 und 6.4 die Zeit bis

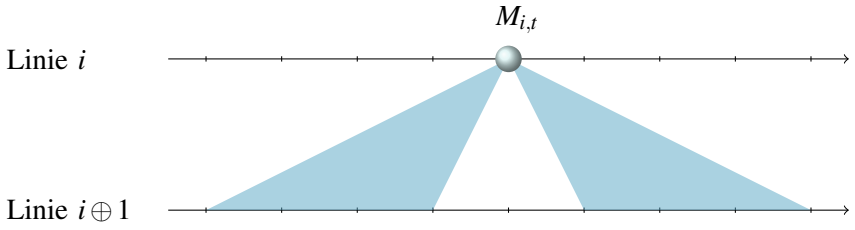


Abbildung 6.1: Nachbarschaft des Knotens $M_{i,t}$ im Verbindungsgraphen

zur Entkopplung der Historie. Der Abschnitt 6.6 beleuchtet die Effizienz zur Prüfung auf das Überschreiten des Schwellwertes. Die Sicherheit des Ansatzes wird in Abschnitt 6.7 betrachtet. Dort wird zudem darauf eingegangen, wie man bei der Anonymisierung und der Entkopplung der Historie auf den Provider als Trusted Party verzichten kann, selbst wenn der Benutzer das Protokoll aktiv angreift.

6.1 Definition und Analyse der zugrundeliegenden Graphenstruktur

Die zugrundeliegende Graphenstruktur bezeichnen wir als *Verbindungsgraph*. Der Verbindungsgraph $\mathcal{G} = (V, E)$ ist ein ungerichteter bipartiter Graph mit Knotenmenge $V = \mathcal{M}_0 \cup \mathcal{M}_1$. Die disjunkten Mengen $\mathcal{M}_0$ und $\mathcal{M}_1$ nennen wir *Linien*. Dabei sei für $i \in \{0,1\}$

$$\mathcal{M}_i = \{M_{i,0}, M_{i,1}, \ldots\}.$$

Der erste Index eines Knotens $M_{i,t}$ bezeichnet seine Linie, der zweite die Position auf seiner Linie. Wenn wir von *vorhergehenden* oder *nachfolgenden Knoten* sprechen, sind dies die Knoten mit kleinerem oder größerem Positionsindex. Die später im Protokoll verwendeten Nachrichten entsprechen den Knoten des Verbindungsgraphen. Daher sprechen wir auch von *Nachrichtenlinien* und von Nachrichten anstelle von Knoten.

Der Verbindungsgraph hat eine regelmäßige Struktur, die vom Protokollparameter $k \in \mathbb{N}$ abhängt. Ein Knoten ist jeweils mit den k vorhergehenden und k nachfolgenden Knoten der anderen Linie verbunden. Es gilt also:

$$\{M_{i,t}, M_{i',t'}\} \in E \quad \text{genau dann, wenn} \quad i' = i \oplus 1 \quad \text{und} \quad 1 \leq |t - t'| \leq k.$$

Sei $M'^0, M'^1, \ldots$ eine Folge von Knoten, so dass

$$M'^t \in \{M_{0,t}, M_{1,t}\},$$

dann wird diese Folge eine *Nachrichtenfolge* genannt.

Der *Sammlungsgraph* $\mathcal{S}$ ist der Subgraph $\mathcal{S}$ des Verbindungsgraphen $\mathcal{G}$, der durch die Knoten $M'^0, M'^1, \ldots$ induziert wird. Die Knotenfolge $M'^0, M'^1, \ldots$ entspricht später einer Folge von Nachrichten eines Benutzers, die die Sammelstelle empfängt und speichert. Daher wird eine solche Folge auch *Nachrichtenfolge* genannt. Die Sammelstelle kann zwei Nachrichten nur dann einander zuordnen, wenn sie diese Nachrichten empfangen hat und diese im Verbindungsgraphen adjazent sind. Damit beschreibt der Sammlungsgraph genau die Zuordenbarkeit der empfangenen Nachrichten und damit die aufgezeichnete Historie der Nachrichten eines Benutzers.

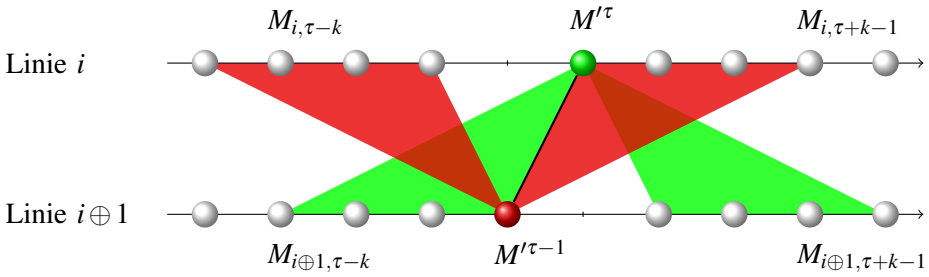


Abbildung 6.2: Ein Wechsel an Position τ im Sammlungsgraph verbindet die Nachrichten $M'^{\tau-k},\ldots,M'^{\tau+k-1}$ zu einer Zusammenhangskomponente.

Seien $M'^0, M'^1, \ldots$ die gespeicherten Nachrichten der Sammelstelle und sei $\mathcal{S}$ der entsprechende Sammlungsgraph. Wenn für ein $\tau \geq 0$ und ein $i \in \{0,1\}$ gilt, dass $M'^{\tau} \in \mathcal{M}_i$ und $M'^{\tau-1} \in \mathcal{M}_{i\oplus 1}$, dann sagen wir, es gibt einen *Wechsel an Position* τ *in* $\mathcal{S}$.

Lemma 6.1 *Sei $\mathcal{S} = (V,E)$ ein Sammlungsgraph von Nachrichten, der durch die Nachrichtenfolge $M'^0,\ldots,M'^{\omega}$ induziert wird. Falls es für ein $\tau > 0$ einen Wechsel an Position τ in $\mathcal{S}$ gibt, dann gehören alle M'^j mit $\max(0,\tau-k) \leq j \leq \min(\tau+k-1,\omega)$ zu derselben Zusammenhangskomponente von $\mathcal{S}$.*

BEWEIS Sei $M'^{\tau} \in \mathcal{M}_i$, dann ist $M'^{\tau-1} \in \mathcal{M}_{i\oplus 1}$. Für eine Nachricht $M \in V$ gilt, dass M sowohl mit den k vorherigen als auch den k nachfolgenden Nachrichten der anderen Linie adjazent ist. Somit sind M'^{τ} und $M'^{\tau-1}$ in $\mathcal{S}$ adjazent.
Für $j \in \{\max(0,\tau-k),\ldots,\min(\tau+k-1,\omega)\}$ gilt: Falls $M'^j \in \mathcal{M}_i$, dann sind M'^j und $M'^{\tau-1}$ adjazent. Falls $M'^j \in \mathcal{M}_{i\oplus 1}$, dann sind M'^j und M'^{τ} adjazent. Siehe dazu Abbildung 6.2. Somit liegen alle M'^j mit $j \in \{\max(0,\tau-k),\ldots,\min(\tau+k-1,\omega)\}$ in derselben Zusammenhangskomponente von $\mathcal{S}$. ∎

An dieser Stelle wollen wir auch die Randfälle betrachten. Wir untersuchen dazu, zu welcher Zusammenhangskomponente die ersten und die letzten Nachrichten der Nachrichtenfolge gehören. Dies ist unmittelbar abhängig davon, an welcher Position sich der erste beziehungsweise der letzte Wechsel befindet.

Ein Knoten heißt *isoliert*, wenn er der einzige Knoten in seiner Zusammenhangskomponente des Sammlungsgraphen ist.

Lemma 6.2 *Sei $\mathcal{S} = (V,E)$ ein Sammlungsgraph von Nachrichten, der durch die Nachrichtenfolge $M'^0, M'^2, \ldots, M'^{\omega}$ induziert wird.*

1. *Falls der erste Wechsel sich an Position $\tau \geq k+1$ befindet, so sind die Nachrichten $M'^0,\ldots,M'^{\tau-k-1}$ isolierte Knoten in $\mathcal{S}$.*

2. *Falls der letzte Wechsel sich an Position $\omega-\tau$ mit $\tau \geq k$ befindet, so sind die Nachrichten $M'^{\omega-\tau+k},\ldots,M'^{\omega}$ isolierte Knoten in $\mathcal{S}$.*

BEWEIS Da eine Nachricht auf einer Linie nur mit den k vorherigen und k nachfolgenden Nachrichten der anderen Linie benachbart sein kann, können die Nachrichten $M'^0,\ldots,M'^{\tau-k-1}$ ($M'^{\omega-\tau+k},\ldots,M'^{\omega}$) potentiell nur mit den Nachrichten $M'^0,\ldots,M'^{\tau-1}$ ($M'^{\omega-\tau},\ldots,M'^{\omega}$) adjazent sein. Da der erste (letzte) Wechsel sich erst an Position τ ($\omega-\tau$) befindet, gehören

die Nachrichten $M'^0,\ldots,M'^{\tau-1}$ ($M'^{\omega-\tau},\ldots,M'^{\omega}$) zu derselben Linie. Somit haben die Nachrichten $M'^0,\ldots,M'^{\tau-k-1}$ ($M'^{\omega-\tau+k},\ldots,M'^{\omega}$) keinen Nachbarn im Sammlungsgraphen und sind isolierte Knoten. ∎

Im Folgenden wollen wir weiter untersuchen, unter welchen Bedingungen eine Nachrichtenfolge der Sammelstelle, das heißt eine Momentaufnahme, eine zusammenhängende Historie der Nachrichten zulässt. Dazu bedienen wir uns einer Zerlegung der Nachrichtenfolge in gleich große Blöcke. Die *ℓ-Blockzerlegung der Knotenfolge* $M'^0,\ldots,M'^{\omega}$ ist eine Folge von Blöcken $B_0,B_1,\ldots,B_{\lceil\frac{\omega+1}{\ell}\rceil-1}$ der Größe ℓ mit $B_i = M'^{i\cdot\ell},\ldots,M'^{i\cdot\ell+\ell-1}$. Der Block $B_{\lceil\frac{\omega+1}{\ell}\rceil-1}$ enthält gegebenenfalls weniger als ℓ Knoten. Die nachfolgenden Definitionen und Analysen lassen sich jedoch auch direkt auf diesen Block übertragen. Daher betrachten wir den letzten Block immer als einen vollständigen Block. Gibt es einen Wechsel an Position $\tau \in \{i\cdot\ell,\ldots,i\cdot\ell+\ell-1\}$, so sagen wir, es gibt einen *Wechsel im ℓ-Block B_i*. Blockzerlegungen werden später für die Abschätzung einer oberen und unteren Schranke für den Zeitpunkt der Entkopplung der Historie verwendet.

Zwei Wechsel an den Positionen $\tau_1 < \tau_2$ heißen *aufeinander folgend*, falls es an keiner Position $\tau_1 < \tau' < \tau_2$ einen Wechsel gibt. Wir sagen dann, dass die beiden Wechsel $\tau_2 - \tau_1$ Positionen voneinander entfernt sind.

Lemma 6.3 *Sei $B_0,B_1,\ldots,B_{\lceil\frac{\omega+1}{k}\rceil-1}$ die k-Blockzerlegung der Nachrichtenfolge $M'^0,\ldots,M'^{\omega}$ sowie $\mathcal{S}$ der entsprechende Sammlungsgraph.*
Falls es in jedem Block B_j mit $j \geq 0$ einen Wechsel gibt, dann ist $\mathcal{S}$ zusammenhängend.

BEWEIS Seien B_i und B_{i+1} mit $i \geq 0$ zwei aufeinander folgende Blöcke. Der letzte Wechsel in Block B_i befinde sich an Position $\tau_i \in \{i\cdot k,\ldots i\cdot k+k-1\}$ und der erste Wechsel in Block B_{i+1} befinde sich an Position $\tau_{i+1} \in \{(i+1)\cdot k,\ldots (i+1)\cdot k+k-1\}$. Nach Lemma 6.1 gehören alle Nachrichten im Block B_i zu derselben Zusammenhangskomponente wie M'^{τ_i} und alle Nachrichten im Block B_{i+1} zu derselben Zusammenhangskomponente wie $M'^{\tau_{i+1}}$. Da $|(i\cdot k+k-1)-(\tau_{i+1})| \leq k$, liegt auch $M'^{i\cdot k+k-1}$ nach Lemma 6.1 in derselben Zusammenhangskomponente wie $M'^{\tau_{i+1}}$. Somit liegen alle Nachrichten aus B_i und B_{i+1} in derselben Zusammenhangskomponente. Falls es in jedem Block einen Wechsel gibt, können wir induktiv schließen, dass $\mathcal{S}$ aus einer einzigen Zusammenhangskomponente besteht und dadurch zusammenhängend ist. ∎

Solange also in jedem Block in der k-Blockzerlegung ein Wechsel vorkommt, ist die Historie der Nachrichten eines Benutzers nicht entkoppelt.

Lemma 6.4 *Sei $\mathcal{S}$ der Sammlungsgraph der Knotenfolge $M'^0,\ldots,M'^{\omega}$. Wir nehmen an, dass sich der erste Wechsel in $\mathcal{S}$ an der Position $\tau \leq k$ und sich der letzte Wechsel an der Position $\tau' \geq \omega-k+1$ befindet. Der Sammlungsgraph $\mathcal{S}$ zerfällt genau dann in mehrere Zusammenhangskomponenten, wenn es zwei aufeinander folgende Wechsel gibt, die mindestens $2k$ Positionen voneinander entfernt sind.*

BEWEIS Da der erste Wechsel nach Position $\tau \leq k$ auftritt, gehören die Nachrichten $M'^0,M'^1,\ldots,M'^{\tau+k-1}$ zu einer Zusammenhangskomponente in $\mathcal{S}$. Analog gilt, dass die Nachrichten nach dem letzten Wechsel $M'^{\tau'-k},\ldots,M'^{\omega}$ zu einer Zusammenhangskomponente gehören. Es gibt also keine isolierten Nachrichten am Anfang und am Ende der Knotenfolge.

Zunächst zeigen wir, dass der Sammlungsgraph zusammenhängend ist, falls alle aufeinander folgende Wechsel weniger als $2k$ voneinander entfernt sind. In diesem Fall argumentieren

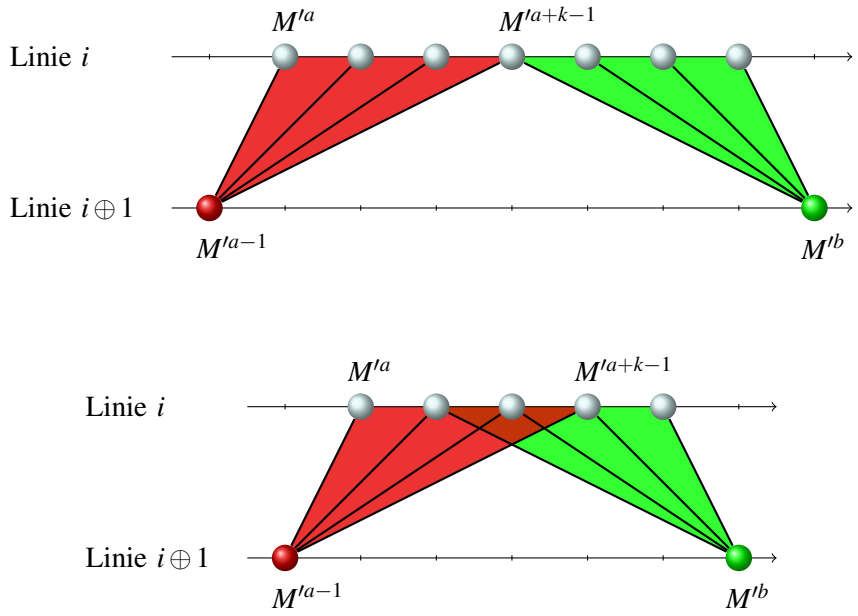


Abbildung 6.3: Zwei im Abstand von höchstens $2k-1$ aufeinander folgende Wechsel im Sammlungsgraphen verbinden die Nachrichten zwischen den Wechseln zu einer Zusammenhangskomponente ($k = 4$).

wir analog zum Beweis von Lemma 6.3. Seien a und b die Positionen zweier aufeinander folgender Wechsel. Die Nachricht M'^{a+k-1}, die sich k Positionen nach dem ersten Wechsel befindet, ist sowohl mit der Nachricht M'^{a-1} direkt vor dem ersten Wechsel als auch mit der Nachricht M'^{b} direkt nach dem zweiten Wechsel adjazent. Abbildung 6.3 veranschaulicht diese Situation. Daraus folgt, dass alle Nachrichten zwischen den Wechseln zu derselben Zusammenhangskomponente an den Positionen a und b gehören. Induktiv gilt dann, dass alle Nachrichten in $\mathcal{S}$ zu einer Zusammenhangskomponente gehören. Also ist der Sammlungsgraph $\mathcal{S}$ zusammenhängend.

Wir nehmen für den zweiten Teil des Beweises an, dass es zwei aufeinander folgende Wechsel gibt, die mindestens $2k$ Positionen voneinander entfernt sind. Daraus können wir schließen, dass es $2k$ Nachrichten $M'^{a},\dots,M'^{a+2k-1}$ in $\mathcal{S}$ zwischen den beiden Wechseln gibt, die zu derselben Linie gehören. Nach Definition des Verbindungsgraphen ist keine Nachricht M'^{b} für $b < a$ mit einer Nachricht M'^{c} für $c \geq a+k$ adjazent. Entsprechend gilt auch, dass keine Nachricht M'^{c} für $c \geq a+2k$ mit einer Nachricht M'^{b} für $b < a+k$ adjazent ist. Da die Nachrichten $M'^{a},\dots,M'^{a+2k-1}$ zu derselben Linie gehören, gibt es keine Kante zwischen diesen $2k$ Nachrichten – potentielle Nachbarn sind nur Nachrichten der anderen Linie. Siehe hierzu auch Abbildung 6.4. Somit können wir festhalten, dass es keine Kante zwischen den Nachrichtenmengen $\{M'^{b} \mid b < a+k\}$ und $\{M'^{c} \mid c \geq a+k\}$ gibt. Folglich zerfällt der Sammlungsgraph $\mathcal{S}$ in mindestens zwei Zusammenhangskomponenten. ∎

6.2 Protokoll für zwei Nachrichtenlinien

In diesem Abschnitt werden wir ein Protokoll vorstellen, durch das die Sammelstelle eine zufällige Nachrichtenfolge für einen Benutzer und damit auch einen zufälligen Sammlungsgraphen für diesen Benutzer erhält. Im Einsatz führen Provider und Sammelstelle das

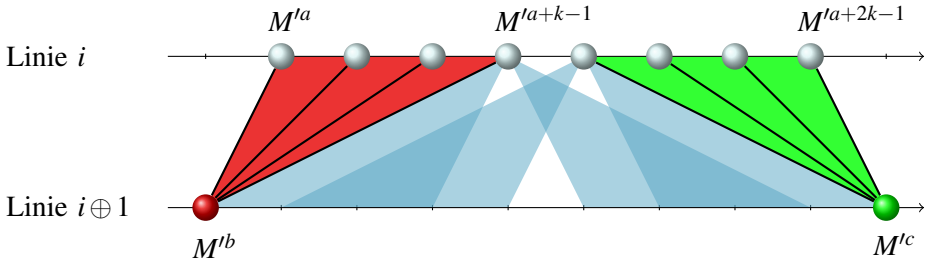


Abbildung 6.4: Der Sammlungsgraph zerfällt in mehrere Zusammenhangskomponenten, wenn zwei aufeinander folgende Wechsel im Sammlungsgraphen mindestens $2k$ Positionen voneinander entfernt sind ($k = 4$).

Protokoll für jeden Benutzer parallel und unabhängig voneinander aus.

6.2.1 Ziel des Protokolls

Auf der Grundlage des Verbindungsgraphen benutzen Sammelstelle und Provider ein probabilistisches Protokoll, um Nachrichten der zufälligen Folge $M'^0, M'^1, \ldots$ für die Sammelstelle zu generieren. M'^t soll dabei zufällig, das heißt mit Wahrscheinlichkeit $\frac{1}{2}$, entweder gleich $M_{i,t}$ oder $M_{i\oplus 1,t}$ sein. Durch wiederholte Anwendung des Protokolls erhält die Sammelstelle eine zufällige Nachrichtenfolge $M'^0, \ldots, M'^\omega$ und kann den induzierten Sammlungsgraphen $\mathcal{S}$ erstellen.

Die Anonymität der Benutzer basiert auf der folgenden wichtigen Eigenschaft des Protokolls: Nehmen wir an, dass die Sammelstelle den Verbindungsgraphen nicht kennt. Dann kann die Sammelstelle aus den empfangenen Nachrichten kein weiteres Wissen über den Verbindungsgraphen gewinnen, als sich aus dem Sammlungsgraphen $\mathcal{S}$ selbst berechnen lässt. Dies gilt ebenfalls, wenn das Protokoll für mehrere Benutzer parallel ausgeführt wird.

6.2.2 Aufbau der Nachrichten

Die Nachrichten des Protokolls entsprechen den Knoten des Verbindungsgraphen. Außerdem beinhalten die Nachrichten Information über die Kanten des Verbindungsgraphen in Form von anonymen Verweisen der Knoten auf ihre Nachbarn. Für die anonymen Verweise benutzen wir eindeutige IDs. Jeder Wert kommt nur in einer Nachricht als ID vor. Der Sammelstelle ist zudem nicht bekannt, welche IDs zu welchen Benutzern gehören und wann welche IDs von den Benutzern verwendet werden.

Jede Nachricht $M_{i,t}$ erhält k IDs: $\mathrm{ID}_0(M_{i,t}), \ldots, \mathrm{ID}_{k-1}(M_{i,t})$. Die Nachricht des j-ten Nachbarn $M_{i\oplus 1,t+j}$ enthält $\mathrm{ID}_{j-1}(M_{i,t})$ und verweist somit auf $M_{i,t}$. Um auf die ID des j-ten Vorgängers für $j \in \{1, \ldots, k\}$ zuzugreifen, verwenden wir den Bezeichner IDpre mit

$$\mathrm{IDpre}_{j-1}(M_{i,t}) = \mathrm{ID}_{j-1}(M_{i\oplus 1,t-j}).$$

Der Aufbau einer solchen Nachricht M besteht aus dem Tupel

$$M = (\mathrm{line}, \mathrm{ID}, \mathrm{IDpre}, \mathrm{load}),$$

wobei $\mathrm{line}(M)$ die Linie der Nachricht M bezeichnet.

i	$\mathrm{ID}_0(M_{i,t})$	…	$\mathrm{ID}_{k-1}(M_{i,t})$	$\mathrm{IDpre}_0(M_{i,t})$	…	$\mathrm{IDpre}_{k-1}(M_{i,t})$	load

Durch Einsetzen der Werte für IDpre erhalten wir:

i	$\mathrm{ID}_0(M_{i,t})$	…	$\mathrm{ID}_{k-1}(M_{i,t})$	$\mathrm{ID}_0(M_{i\oplus 1,t-1})$	…	$\mathrm{ID}_{k-1}(M_{i\oplus 1,t-(k-1)-1})$	load

Abbildung 6.5: Nachricht im Protokoll, die dem Knoten $M_{i,t}$ des Verbindungsgraphen eines Nutzers entspricht

Zur Vereinfachung nehmen wir an, dass load(M) alle weiteren Einträge enthält. Für unsere Vorratsdatenspeicherung wären dies also zusätzlich zum ursprünglichen load(M) noch die Einträge time(M) und share(M). Der Aufbau und der Zusammenhang der Nachrichten sind in den Abbildungen 6.5 und 6.6 genauer dargestellt.

Für die praktische Umsetzung gehen wir davon aus, dass viele Benutzer gleichzeitig Nachrichten an die Sammelstelle senden. Dann kann es schwierig sein sicherzustellen, dass die Benutzer global eindeutige IDs deterministisch und unabhängig voneinander generieren. Bei moderater Länge der IDs können wir allerdings die IDs zufällig erzeugen und die Wahrscheinlichkeit einer Kollision sehr gering ist, siehe dazu auch Abschnitt 6.6. Wir können somit annehmen, dass jede ID nur ein einziges Mal als Verweis benutzt wird. Da wir die IDs unabhängig voneinander generieren, kann die Sammelstelle den Nachrichten nur Verbindungsinformation entnehmen, wenn sie dieselbe ID zweimal sieht: sowohl in der Nachricht $M_{i,t}$ (das heißt $\mathrm{ID}_{j-1}(M_{i,t})$) als auch in $M_{i\oplus 1,t+j}$ (das heißt $\mathrm{IDpre}_{j-1}(M_{i\oplus 1,t+j}) = \mathrm{ID}_{j-1}(M_{i,t})$). Würden wir weniger als k IDs für die Verweise auf eine Nachricht $M_{i,t}$ benutzen, so gilt nach dem Taubenschlagprinzip, dass die Nachrichten für mindestens zwei Nachbarn $M_{i\oplus 1,t+j_1} \neq M_{i\oplus 1,t+j_2}$ von $M_{i,t}$ dieselbe ID von $M_{i,t}$ als Verweis enthalten. Falls in der Nachrichtenfolge $M'^{t+j_1} = M_{i\oplus 1,t+j_1}$ und $M'^{t+j_2} = M_{i\oplus 1,t+j_2}$, dann könnte die Sammelstelle anhand derselben Verweis-ID zu $M_{i,t}$ erkennen, dass M'^{t+j_1} und M'^{t+j_2} zum Verbindungsgraphen desselben Benutzers gehören.

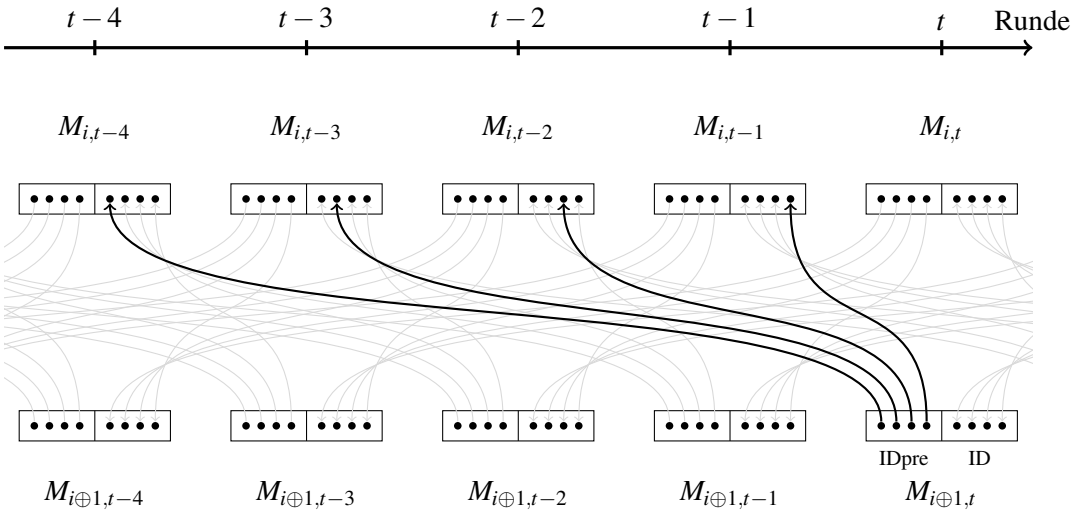


Abbildung 6.6: Skizze des Zusammenhangs der Nachrichten der beiden Nachrichtenlinien

6.2.3 Beschreibung des Protokolls

Nachdem wir gezeigt haben, wie sich die Nachrichten für die Knoten des Verbindungsgraphen zusammensetzen, wird im Weiteren dargelegt, wie der Provider die Nachricht M'^t für einen Benutzer an die Sammelstelle übermittelt. Wir gehen davon aus, dass der Provider zuvor die Nachrichten $M_{i,t'}$ für $t' < t$ generiert hat. In der Beschreibung des Protokolls beziehen wir uns auf die Nachrichten eines Benutzers.

Initialisierung (vor der ersten Benutzung des Protokolls)

1. Der Provider generiert zufällig und unabhängig für $i \in \{0,1\}$ und $\tau \in \{1,\ldots,k\}$ $\mathrm{ID}_{\xi-1}(M_{i,-\tau})$, wobei $\xi \in \{\tau,\ldots,k\}$.

Mithilfe dieser IDs können die Vorgänger-IDs IDpre_j der Nachrichten $M_{i,j}$ für alle $j \geq 0$ mit Werten besetzt werden. Da diese Werte unabhängig sind und die Sammelstelle keinesfalls Nachrichten mit diesen Werten als Nachrichten-ID bekommt, geben diese Werte kein Wissen über den Zusammenhalt der Nachrichten wieder.

Protokoll zur Auswahl und Übermittlung der Nachricht M'^t

1. Der Provider generiert zufällig und unabhängig für $i \in \{0,1\}$ die eindeutigen IDs $\mathrm{ID}_0(M_{i,t}),\ldots,\mathrm{ID}_{k-1}(M_{i,t})$.
2. Der Provider erstellt die Nachrichten $M_{0,t}$ und $M_{1,t}$.
3. Der Provider wählt ein zufälliges Bit $\pi \in \{0,1\}$.
4. Die Sammelstelle wählt ein zufälliges Bit $\sigma \in \{0,1\}$.
5. Die Sammelstelle erhält $M'^t = \mathrm{OT}_\sigma(M_{\pi,t}, M_{\pi\oplus 1,t})$ durch sicheres Oblivious-Transfer vom Provider.

Eine Nachrichtenfolge $M'^0, M'^1, \ldots, M'^\omega$ eines Benutzers ist nur eine Momentaufnahme der Nachrichten, die die Sammelstelle empfängt. Es ist durchaus zu erwarten, dass die Sammelstelle zu einem späteren Zeitpunkt eine weitere Nachricht des Benutzers empfängt. Dies wäre dann die Nachricht $M'^{\omega+1}$. Wenn diese Nachricht einen Wechsel herbeiführt, kann sie die vorherigen Nachrichten der anderen Linie aus $M'^{\omega-k+1},\ldots,M'^\omega$ verbinden. Die isolierten Nachrichten am Ende einer Nachrichtenfolge können also noch durch spätere Nachrichten mit einer Zusammenhangskomponente verbunden werden. Wir nennen daher die isolierten Nachrichten aus $M'^{\omega-k+1},\ldots,M'^\omega$ *die unbestimmte Historie* des entsprechenden Benutzers.

Dieses Protokoll beinhaltet bereits einige erweiterte Sicherheitsaspekte für den Fall, dass der Provider keine Trusted Party ist. Wir betrachten dazu das Beispiel, wenn der Benutzer die Rolle des Providers übernimmt. Ist der Benutzer ehrlich, dann reicht es, dass der Benutzer das Permutationsbit π zufällig wählt, um $M'^t = M_{\pi,t}$ an die Sammelstelle zu senden. Infolgedessen erhält die Sammelstelle die Nachricht einer Linie mit Wahrscheinlichkeit $\frac{1}{2}$. Für den Benutzer wäre es vorteilhaft, nur Nachrichten einer Linie zu senden. In diesem Fall wären alle Nachrichten isolierte Knoten im Sammlungsgraphen und die Historie wäre vollständig entkoppelt. Entsprechend wäre es für die Sammelstelle von Vorteil, Nachrichten von abwechselnden Linien zu bekommen. Wenn wir annehmen, dass der Benutzer und die Sammelstelle

lediglich die Linie von M'^t beeinflussen möchten, garantiert die Wahl von $M'^t = M_{\pi\oplus\sigma,t}$, dass die Linie von M'^t zufällig ist, solange das Permutationsbit π oder das Selektionsbit σ zufällig gewählt ist. Wenn wir zur Wahl der Nachricht wie oben beschrieben ein sicheres Oblivious-Transfer Protokoll einsetzen, dann ist es außerdem für eine Sammelstelle, die einen aktiven Angriff durchführt, nicht möglich M'^t zu beeinflussen. In Abschnitt 6.7 werden wir uns eingehender mit aktiven Angreifern befassen, insbesondere mit der Erweiterung zu einem robusten Protokoll bei aktiven Angriffen der Benutzer.

6.2.4 Verbindung zur privaten Vorratsdatenspeicherung

Wir können das vorherige Protokoll sehr leicht an das Szenario der privaten Vorratsdatenspeicherung anpassen. Der Provider berechnet die Einträge share, load und time wie in Kapitel 4. Mit diesen Einträgen füllen wir den load-Eintrag der Nachrichten dieses Protokolls. Der Provider sendet eine neue Nachricht M'^t an die Sammelstelle, falls ein Benutzer eine kritische Aktion ausführt. Da unkritische Nachrichten nur identifiziert werden können, wenn der entsprechende Schlüssel vorliegt, können sie wie in Kapitel 4 beschrieben übertragen werden.

6.2.5 Gleichzeitige Durchführung des Protokolls für mehrere Benutzer

Man beachte, dass die Nachrichten keine nutzerspezifischen Bezeichner enthalten. Die Sammelstelle speichert die kritischen Nachrichten aller Benutzer gleichzeitig und damit die Vereinigung über alle Sammlungsgraphen. Diesen Graphen bezeichnen wir als den *vereinigten Sammlungsgraphen*. Nur wenn die Nachrichten in derselben Zusammenhangskomponente des vereinigten Sammlungsgraphen enthalten sind, kann die Sammelstelle zwei Nachrichten M'^a und M'^b einander zuordnen. Die Sammelstelle erkennt dann direkt, dass diese Nachrichten von demselben Benutzer stammen.

Es ergeben sich im vereinigten Sammlungsgraphen keine Überschneidungen der Sammlungsgraphen einzelner Nutzer, da die IDs der Nachrichten eindeutig sind und die Nachrichten als Knoten nur Nachbarn im Sammlungsgraphen eines Nutzers haben. Als *mögliche Vorgänger einer Zusammenhangskomponente* Z in dem Sammlungsgraphen $\mathcal{S}$ bezeichnen wir dabei die Zusammenhangskomponenten Z' von $\mathcal{S}$, so dass alle Nachrichten in Z' vor allen Nachrichten in Z erzeugt wurden. Unter der *Vorhistorie* einer Zusammenhangskomponente Z verstehen wir alle möglichen Vorgänger von Z, die zu demselben Benutzer gehören wie Z. In dem vereinigten Sammlungsgraphen, eingeschränkt auf Nachrichten von unidentifizierten Benutzern, kann die Sammelstelle für eine Zusammenhangskomponente Z nicht zwischen den Komponenten der Vorhistorie und den Zusammenhangskomponenten mit Nachrichten unterscheiden, die zeitlich vorher erzeugt wurden und von anderen Nutzern stammen. Zerfällt folglich der Sammlungsgraph eines Nutzers in mehrere Zusammenhangskomponenten, so sind die entsprechenden Vorhistorien der anderen Nutzer als Vorhistorien des betrachteten Nutzers möglich. Die Sammelstelle kann dementsprechend die Nachrichten der unterschiedlichen Komponenten nicht mehr richtig zuordnen, die Historie eines Nutzers ist entkoppelt und nicht von entsprechenden Vorhistorien der anderer Nutzer zu unterscheiden.

6.3 Erwartungswert der Zeit bis zur Entkopplung der Historie

Bei der Analyse der strukturellen Eigenschaften des Sammlungsgraphen in Abschnitt 6.1 haben wir festgestellt, unter welchen Umständen dieser zerfällt. Eine Möglichkeit ist die Existenz von isolierten Knoten zu Beginn oder am Ende der Folge $M'^0,\ldots,M'^\omega$. Isolierte Knoten zu Beginn der Folge sind alt. Ihre Speicherung liegt bei längerem Lauf des Protokolls weit zurück, während isolierte Knoten am Ende der Folge gegebenenfalls zur unabhängigen Historie gehören und durch später eintreffende Nachrichten in Zusammenhang gebracht werden können. Daher ist für uns die Frage interessant, wann die Historie außerhalb der Randfälle entkoppelt wird. Wir gehen bei dieser Analyse davon aus, dass die Nachrichtenfolge $M'^0,\ldots,M'^\omega$ keine unabhängige Historie hat. Damit ist die Nachrichtenfolge lang und die Knoten am Ende der Folge sind nicht isoliert. Die Sammelstelle interessiert sich nun für die Rekonstruktion der Historie der zuletzt erhaltenen Nachricht M'^ω. Die Historie der Nachricht M'^ω ist zugleich ihre Zusammenhangskomponente im Sammlungsgraphen der Sammelstelle. Die Historie ist nach Lemma 6.4 genau dann entkoppelt, wenn nacheinander mindestens $2k$-mal die Nachrichten einer Linie gewählt werden. Bei der Übertragung einer Nachricht erhält die Sammelstelle mit gleicher Wahrscheinlichkeit die Nachricht der einen oder der anderen Linie. Im Folgenden zeigen wir, dass wir den stochastischen Prozess, der den Zerfall des Sammlungsgraphen in mehrere Zusammenhangskomponenten und somit die Entkopplung der Historie beschreibt, als Markov-Kette darstellen können.

Da die Sammelstelle die Nachricht M'^ω zuletzt empfangen hat und sie die Historie dieser Nachricht rekonstruieren möchte, untersuchen wir die Nachrichtenfolge der Sammelstelle rückwärts. Wir betrachten die Nachrichten $M'^t \in \{M_{0,t}, M_{1,t}\}$ als uniforme Zufallsvariable und analysieren nacheinander die Werte $M'^\omega, M'^{\omega-1},\ldots$. An dieser Stelle nehmen wir für die Analyse an, dass die Folge unendlich oder zumindest lang ist. Die Ausgänge der unabhängigen Zufallsexperimente für die Nachricht M'^i bestimmen die Zustandsübergänge einer Markov-Kette. Die Zustände der Markov-Kette beschreiben, wie viele Nachrichten derselben Linie die Sammelstelle hintereinander empfangen hat. Wir stellen den Zustandsraum $\mathcal{Z}$ wie folgt dar:

$$\mathcal{Z} = \{z_1,\ldots,z_{2k}\},$$

wobei die Markov-Kette sich im Zustand z_i befindet, falls die Sammelstelle genau i Nachrichten derselben Linie hintereinander empfangen hat. Wird der Zustand z_{2k} erreicht, so zerfällt der Sammlungsgraph in mehrere Komponenten und die Historie ist entkoppelt (siehe Lemma 6.4).

Beispiel 6.5 *Sei $k = 2$. Wir schauen uns rückwärts eine Folge von Nachrichten an und bestimmen die entsprechenden Zustände der Markov-Kette:*

τ	0	1	2	3	4	5	6	7
$M'^{\omega-\tau} \in \mathcal{M}_0$	✓	✓		✓				
$M'^{\omega-\tau} \in \mathcal{M}_1$			✓		✓	✓	✓	✓
Z_τ	z_1	z_2	z_1	z_1	z_1	z_2	z_3	z_4

In diesem Beispiel ist die Historie der Nachricht M'^ω entkoppelt, da die Nachrichten $M'^{\omega-4},\ldots,M'^{\omega-7}$ auf derselben Linie liegen und somit die Markov-Kette für $\tau = 7$ den entkoppelnden Zustand z_{2k} erreicht. Die Zusammenhangskomponente von M'^ω umfasst lediglich $M'^\omega,\ldots,M'^{\omega-5}$.

Da uns nur der Eintritt der ersten Entkopplung interessiert, modellieren wir den Zustand z_{2k} als absorbierenden Zustand, das heißt, es gilt:

$$\Pr[Z_{j+1} = z_{2k} \mid Z_j = z_{2k}] \quad = \quad 1.$$

Die Markov-Kette startet mit Nachricht M'^{ω} im Zustand z_1, da dies die erste betrachtete Nachricht ist und wir setzen

$$\Pr[Z_0 = z_1] \quad = \quad 1.$$

Für die übrigen Zustände gilt:

$$\begin{aligned} \Pr[Z_{j+1} = z_{i+1} \mid Z_j = z_i] &= \frac{1}{2} \quad \text{und} \\ \Pr[Z_{j+1} = z_1 \mid Z_j = z_i] &= \frac{1}{2} \end{aligned}$$

mit $i \in \{1, \ldots, 2k-1\}$, da die Nachricht M'^{j+1} mit gleicher Wahrscheinlichkeit derselben oder der gegenüberliegenden Linie von M'^{j} angehört. Im ersten Fall hat die Sammelstelle nunmehr $i+1$ Nachrichten derselben Linie hintereinander empfangen. Im zweiten Fall findet ein Wechsel statt. Somit hat die Sammelstelle am aktuellen Punkt $j+1$ nur eine Nachricht derselben Linie hintereinander empfangen.

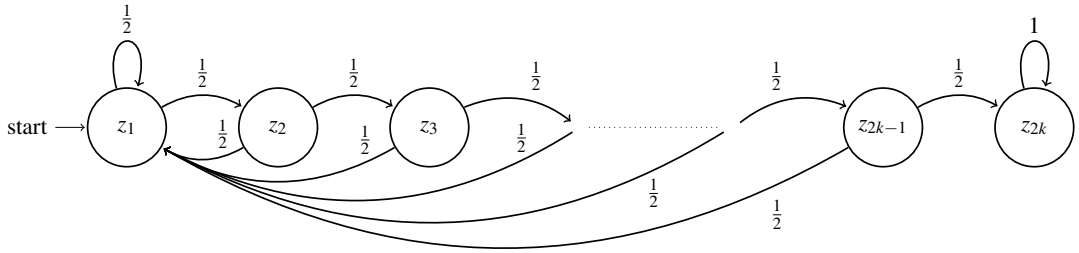


Abbildung 6.7: Darstellung der Markov-Kette für den stochastischen Prozess der Entkopplung

Nun können wir die Matrix $P = (P_{a,b})_{a,b \in \{1,\ldots,2k\}}$ der Übergangswahrscheinlichkeiten $P_{a,b} = \Pr[Z_{j+1} = z_b \mid Z_j = z_a]$ für alle j aufstellen.

$$P = \begin{array}{c} \\ z_a \\ \\ z_1 \\ \\ \\ z_{2k-1} \\ z_{2k} \end{array} \begin{array}{c} \begin{array}{cccccc} z_b & & z_1 & & & z_{2k-1} \; z_{2k} \end{array} \\ \left(\begin{array}{cccc|cc} \frac{1}{2} & \frac{1}{2} & & & 0 & \\ \frac{1}{2} & 0 & \frac{1}{2} & & & \\ \vdots & & & \ddots & & \\ & & & & 0 & \\ \hline 0 & \cdots & & 0 & 1 & \end{array} \right) \end{array}$$

Die Struktur von P beschreiben wir durch vier Submatrizen $Q, B, 0, 1$:

$$P = \left(\begin{array}{c|c} Q & B \\ \hline 0 & 1 \end{array}\right)$$

Die erwartete Zeit, in der die Markov-Kette in den nicht absorbierenden Zuständen verweilt, wenn sie im Zustand z_i startet, erhalten wir durch Berechnung von

$$((I-Q)^{-1})_i \cdot (1,\ldots,1)^{\mathsf{T}},$$

wobei I die Einheitsmatrix und A_i die i-te Zeile einer Matrix A ist.

Sei E die Zufallsvariable, die die Anzahl der Schritte beschreibt, bis die Markov-Kette ihren absorbierenden Zustand z_{2k} beginnend mit Zustand z_1 erreicht. Also ist die erwartete Anzahl von Schritten bis zur Absorption beginnend in Zustand z_1

$$\mathbb{E}(E) \quad = \quad ((I-Q)^{-1})_1 \cdot (1,\ldots,1)^{\mathsf{T}} + 1.$$

Lemma 6.6 *Die erwartete Anzahl von Schritten bis zur Absorption der Markov-Kette, die den stochastischen Prozess der Entkopplung des Sammelgraphen beschreibt, beträgt*

$$\mathbb{E}(E) = 2^{2k} - 1.$$

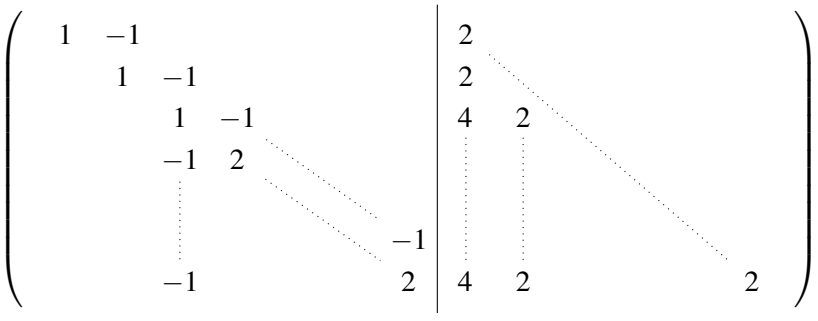
BEWEIS Zur Berechnung der Inversen von $(I-Q)$ überführen wir $(I-Q)$ schrittweise durch elementare Zeilenumformungen nach I. Die Zeilenumformungen angewandt auf die Einheitsmatrix ergeben dann die Inverse.

$$(I-Q \mid I) = \left(\begin{array}{cccc|ccc} \frac{1}{2} & -\frac{1}{2} & & & 1 & & \\ -\frac{1}{2} & 1 & -\frac{1}{2} & & & \ddots & \\ & \ddots & \ddots & \ddots & & & \\ & & & -\frac{1}{2} & & & 1 \\ -\frac{1}{2} & & & 1 & & & \end{array}\right).$$

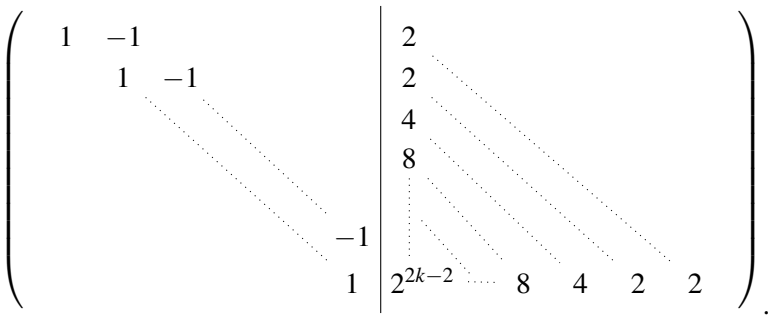
Alle nicht beschriebenen Einträge der Matrizen haben den Wert 0. Die Multiplikation der gesamten Matrix mit dem Faktor 2 und die anschließende Addition der ersten Zeile auf alle anderen Zeilen erzeugen die erste Spalte der Einheitsmatrix:

$$\left(\begin{array}{cccc|ccc} 1 & -1 & & & 2 & & \\ & 1 & -1 & & & \ddots & \\ & -1 & 2 & \ddots & & & \\ & & \ddots & -1 & & & 2 \\ & -1 & & 2 & 2 & & \end{array}\right).$$

Analog können wir nun die zweite Zeile auf alle folgenden Zeilen addieren:



Wenn wir nun Schritt für Schritt die dritte bis $(2k-2)$-te Zeile auf alle nachfolgenden Zeilen addieren, erhalten wir:



Auf der linken Seite erhalten wir die Einheitsmatrix und auf der rechten Seite die Inverse von $(I-Q)$, wenn wir sukzessive für $i = 2k-1,\ldots,2$ die i-te auf die $(i-1)$-te Zeile addieren. Da die Markov-Kette im Zustand z_1 startet, ist für die Berechnung des Erwartungswertes nur die erste Zeile der Inversen interessant. Diese setzt sich nach der letzten Umformungsphase als Zeilenvektor bestehend aus den Gesamtsummen der Spalten zusammen:

$$\begin{aligned}((I-Q)^{-1})_1 &= \left(2+\sum_{i=1}^{2k-2}2^i, 2+\sum_{i=1}^{2k-3}2^i, \ldots, 2+\sum_{i=1}^{1}2^i, 2\right)\\ &= \left(1+\sum_{i=0}^{2k-2}2^i, 1+\sum_{i=0}^{2k-3}2^i, \ldots, 1+\sum_{i=0}^{1}2^i, 1+2^0\right)\\ &= \left(2^{2k-1}, 2^{2k-2}, \ldots, 2^2, 2^1\right).\end{aligned}$$

Nun können wir schließen, dass

$$((I-Q)^{-1})_1 \cdot (1,\ldots,1)^{\mathsf{T}} + 1 = \left(\sum_{i=1}^{2k-1}2^i\right) + 2^0 = 2^{2k}-1.$$

■

Sei $E' = E+1$. Da wir M'^0 als ersten Schritt nicht mitzählen und gleich im Zustand z_1 beginnen, beschreibt die Zufallsvariable E' folglich die Anzahl der Nachrichten bis zur Entkopplung und hat den Erwartungswert

$$\mathbb{E}(E') = \mathbb{E}(E)+1 = 2^{2k}$$

Daraus können wir direkt folgenden Satz ableiten:

Theorem 6.7 *Sei $M'^0,\ldots,M'^\omega$ eine zufällige Nachrichtenfolge, das heißt, M'^i stammt mit Wahrscheinlichkeit $\frac{1}{2}$ aus $\mathcal{M}_0$ oder aus $\mathcal{M}_1$. Wir können erwarten, dass der entsprechende Sammlungsgraph $\mathcal{S}$ nach $\mathbb{E}(E') = 2^{2k}$ Nachrichten in mehrere Zusammenhangskomponenten zerfällt.*

6.4 Schranken für die Zeit bis zur Entkopplung der Historie

Um besser abschätzen zu können, wann mit großer Wahrscheinlichkeit mit einer Entkopplung der Historie zu rechnen ist, könnten wir nun mit Kenntnis von $\mathbb{E}(E')$ die Markov-Ungleichung anwenden. Allerdings können wir durch eine etwas relaxierte Analyse bessere Schranken erreichen. Dazu werden wir unter anderem die Ergebnisse der Lemmata 6.3 und 6.4 verwenden.

Zunächst wollen wir analysieren, nach wie vielen Schritten der Sammlungsgraph in mehrere Zusammenhangskomponenten zerfällt. Nach Lemma 6.4 gilt:

Korollar 6.8 *Seien $B_0,B_1,\ldots$ die $2k$-Blockzerlegung der Nachrichtenfolge $M'^0,M'^1,\ldots$ der Sammelstelle und $\mathcal{S}$ der entsprechende Sammlungsgraph. Falls es einen Block B_i ohne Wechsel gibt, so zerfällt $\mathcal{S}$ in mehrere Zusammenhangskomponenten.*

Der Fall nach Korollar 6.8 ist lediglich hinreichend, aber nicht notwendig für den Zerfall des Sammlungsgraphen. Folglich ist die Wahrscheinlichkeit, dass $\mathcal{S}$ spätestens nach der $((i+1)\cdot 2k)$-ten Nachricht eines Benutzers zerfällt, größer als die Wahrscheinlichkeit, dass die Nachrichten im Block B_i alle zu derselben Linie gehören und dass dies für die Blöcke $B_0,\ldots,B_{i-1}$ nicht gilt. Die Wahrscheinlichkeit für keinen Wechsel innerhalb eines $2k$-Blocks ist durch

$$\begin{aligned}\Pr[\text{kein Wechsel im } 2k\text{-Block } B_i] &= \Pr[\text{alle Nachrichten des Blocks gehören zu } \mathcal{M}_0] \\ &\quad + \Pr[\text{alle Nachrichten des Blocks gehören zu } \mathcal{M}_1] \\ &= 2\cdot 2^{-2k} \\ &= 2^{-2k+1}\end{aligned}$$

bestimmt. Sei KW die Zufallsvariable, die für $KW = i+1$ beschreibt, dass es im Block B_i keinen Wechsel und in allen Blöcken $B_0,\ldots,B_{i-1}$ mindestens einen Wechsel gibt. Das heißt, im $(i+1)$-ten Block gibt es zum ersten Mal keinen Wechsel. Offensichtlich ist KW geometrisch verteilt mit Parameter $p = 2^{-2k+1}$ und es gilt für den Erwartungswert

$$\mathbb{E}(KW) = \frac{1}{2^{-2k+1}} = 2^{2k-1}.$$

In einem $2k$-Block kommen $2k$ Nachrichten vor. Also erwarten wir, dass die Voraussetzung von Korollar 6.8 nach

$$\mu_{KW} = 2k\cdot\mathbb{E}(KW) = k\cdot 2^{2k} = k\cdot\mathbb{E}(E')$$

Nachrichten eintritt. Bezogen auf die Anzahl der Nachrichten ist μ_{KW} um den Faktor k größer als $\mathbb{E}(E')$. Allerdings werden wir sehen, dass wir durch die weitere strukturellen Annahmen der geometrischen Verteilung eine bessere Abschätzung für die Abweichung vom Erwartungswert $\mathbb{E}(E')$ mittels Analyse von KW erhalten als mit der Markov-Ungleichung für E'.

Die Verteilungsfunktion der geometrischen Verteilung lässt sich direkt berechnen:

$$\begin{aligned}\Pr[KW \geq a] &= 1 - \Pr[KW \leq a-1] \\ &= 1 - \sum_{b=0}^{a-2} (1-p)^b p \\ &= 1 - p\frac{((1-p)^{a-1}-1)}{(1-p)-1} \\ &= 1 - (1-(1-p)^{a-1}) \\ &= (1-p)^{a-1}.\end{aligned}$$

Damit können wir folgende Abschätzung herleiten:

$$\begin{aligned}\Pr[KW \geq c \cdot \mathbb{E}(KW)] &= (1-p)^{c\cdot\mathbb{E}(KW)-1} \\ &= \left(1 - \frac{1}{\mathbb{E}(KW)}\right)^{c\cdot\mathbb{E}(KW)-1} \\ &\leq e^{-c}. \end{aligned} \tag{6.1}$$

Da die Zufallsvariable KW Blöcke mit $2k$ Nachrichten zählt und

$$\Pr[KW < x] \leq \Pr[E' < x \cdot 2k]$$

für $x \geq 1$ ist, können wir die Abschätzungen für E' und KW in Beziehung setzen

$$\begin{aligned}\Pr[KW \geq c \cdot \mathbb{E}(KW)] &\geq \Pr[E' \geq c \cdot 2k \cdot \mathbb{E}(KW)] \\ &= \Pr[E' \geq c \cdot k \cdot 2^{2k}] \\ &= \Pr[E' \geq c \cdot k \cdot \mathbb{E}(E')] \end{aligned} \tag{6.2}$$

mit $c \geq 1$.

Zum Vergleich der Markov-Schranke für E' mit der Schranke für KW schätzen wir zunächst E' mithilfe der Markov-Ungleichung ab.

$$\Pr[E' \geq c \cdot k \cdot \mathbb{E}(E')] \leq \frac{\mathbb{E}(E')}{c \cdot k \cdot \mathbb{E}(E')} \tag{6.3}$$

$$= \frac{1}{c \cdot k} \tag{6.4}$$

für beliebige Werte $c \cdot k > 1$.

Beobachtung 6.9 *Für alle $c \geq 1$ mit*

$$ck \leq e^c \tag{6.5}$$

liefert die Schranke für KW (Ungleichung 6.1) für die Wahrscheinlichkeit $\Pr[E' \geq c \cdot k \cdot \mathbb{E}(E')]$ *eine genauso gute oder bessere obere Schranke als die Markov-Schranke (Ungleichung 6.4) für E'. Die Ungleichung 6.5 gilt insbesondere für $c \geq \frac{e}{e-1}\ln(k)$.*

BEWEIS Die erste Behauptung folgt unmittelbar aus den Ungleichungen 6.1, 6.4 und 6.5.

Wir werden nun eine Schranke für c in Abhängigkeit von k herleiten, so dass Ungleichung 6.5 erfüllt ist. Wir interessieren uns zunächst für möglichst kleine Werte r, so dass

$$c - \ln(c) \geq \frac{c}{r} \tag{6.6}$$

für alle $c > 0$. Dies gilt genau dann, wenn

$$\frac{r-1}{r}c \geq \ln(c). \tag{6.7}$$

Wir betrachten die Gerade $\frac{r-1}{r} \cdot c$ in Abhängigkeit von c und die Steigung dieser Geraden als Funktion in r. Da

$$\frac{d}{dr}\frac{r-1}{r} = \frac{1}{r^2} > 0,$$

ist die Steigung monoton wachsend in $r > 0$. Somit können wir das kleinste $r = r^*$, für das die Ungleichung 6.7 erfüllt ist, durch die Berechnung der Tangenten $\frac{r^*-1}{r^*}c$ an $\ln(c)$ bestimmen. Im Tangentialpunkt c^* muss folglich gelten, dass

$$\frac{d}{dc^*}\ln(c^*) = \frac{r^*-1}{r^*}.$$

Dies gilt genau dann, wenn

$$\frac{1}{c^*} = \frac{r^*-1}{r^*} \quad \text{beziehungsweise} \quad c^* = \frac{r^*}{r^*-1}.$$

Für den Tangentialpunkt gilt in Ungleichung 6.7 die Gleichheit. Siehe dazu auch Abbildung 6.8. Um den Tangentialpunkt zu erhalten, setzen wir den Wert für c^* in die resultierende Gleichung ein:

$$\frac{r^*-1}{r^*} \cdot \frac{r^*}{r^*-1} = \ln\left(\frac{r^*}{r^*-1}\right).$$

Somit erhalten wir:

$$\begin{aligned}
\ln\left(\frac{r^*}{r^*-1}\right) &= 1, \\
\frac{r^*}{r^*-1} &= e \quad \text{und} \\
r^* &= \frac{e}{e-1} \approx 1.5820 \quad \text{sowie} \\
c^* &= e.
\end{aligned}$$

Die Ungleichung 6.5 können wir umformen zu

$$\begin{aligned}
k &\leq e^{c-\ln(c)} \quad \text{und} \\
\ln(k) &\leq c - \ln(c).
\end{aligned}$$

Wählen wir

$$\begin{aligned}
\ln(k) &\leq \frac{c}{r^*} \quad \text{und damit nach Ungleichung 6.6} \\
c - \ln(c) &\geq \frac{c}{r^*} \geq \ln(k), \quad \text{dann gilt:} \\
c &\geq \frac{e}{e-1}\ln(k)
\end{aligned}$$

und die Ungleichung 6.5 ist erfüllt. ■

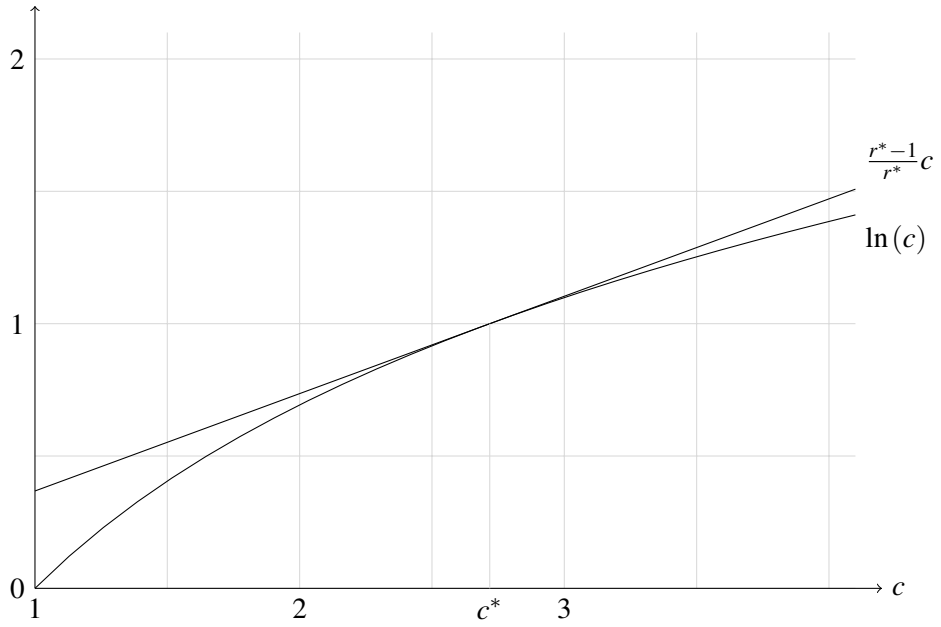


Abbildung 6.8: Vergleich der Lage von $\frac{r^*-1}{r^*} \cdot c$ und $\ln(c)$

6.5 Benötigter Zusammenhalt der Historie in Abhängigkeit des Schwellwertes d

In der folgenden Analyse werden wir bestimmen, wie groß der Wert von k in Bezug auf die Anzahl d der notwendigen kritischen Nachrichten bis zur Entschlüsselung sein sollte. Der Wert für k sollte möglichst klein gewählt werden. Kleinere Werte für k entkoppeln die Historie schneller, das heißt, die Zusammenhangskomponenten werden kleiner. Damit die Sammelstelle d aufeinander folgende kritische Nachrichten einander zuordnen kann, müssen diese im vereinigten Sammlungsgraphen in derselben Zusammenhangskomponente liegen. Für eine kleine Fehlerwahrscheinlichkeit ε und beliebiges $\tau \geq 0$ fordern wir daher:

$$\Pr[M'^{\tau}, \ldots, M'^{\tau+d-1} \text{ liegen in derselben Zusammenhangskomponente}] \stackrel{!}{\geq} 1-\varepsilon. \quad (6.8)$$

Lemma 6.10 *Sei* $1 \geq \varepsilon > 0$ *und* $d \geq k \geq 2$. *Falls* $k \geq \log(2d) + \log(\frac{1}{\varepsilon}) + 2 - \log\log(e)$, *so ist die Ungleichung 6.8 erfüllt.*

BEWEIS Nach Lemma 6.3 wissen wir, falls in aufeinander folgenden Blöcken der k-Blockzerlegung einer Nachrichtenfolge der Sammelstelle jeweils ein Wechsel vorkommt, dann gehören die Nachrichten dieser Blöcke zu derselben Zusammenhangskomponente. Wenn in $\lceil \frac{d}{k} \rceil$ k-Blöcken jeweils mindestens ein Wechsel vorkommt, dann enthält die Zusammenhangskomponente d oder mehr aufeinander folgende Nachrichten. Diese Bedingung ist hinreichend. Folglich ist die Wahrscheinlichkeit, dass für eine Nachrichtenfolge $M'^{\tau}, M'^{\tau+1}, \ldots$ die ersten d Nachrichten in derselben Zusammenhangskomponente liegen, größer als die Wahrscheinlichkeit, dass in den ersten $\lceil \frac{d}{k} \rceil$ k-Blöcken dieser Folge jeweils mindestens ein Wechsel

vorkommt. Wir zeigen daher:

$$\begin{aligned}
&\Pr[M'^{\tau},\ldots,M'^{\tau+d-1} \text{ liegen in derselben Zusammenhangskomponente}]\\
&\geq \Pr[\text{mindestens ein Wechsel in jedem der ersten } \lceil d/k\rceil \ k\text{-Blöcke}]\\
&\overset{!}{\geq} 1-\varepsilon.
\end{aligned}$$

Da die Linie jeder Nachricht der Sammelstelle unabhängig mit einer Wahrscheinlichkeit von $\frac{1}{2}$ gewählt wird, gilt:

$$\begin{aligned}
&\Pr[\text{mindestens ein Wechsel in jedem der ersten } \lceil d/k\rceil \ k\text{-Blöcke}]\\
&= (1-\Pr[\text{kein Wechsel in einem } k\text{-Block}])^{\lceil \frac{d}{k}\rceil}\\
&= (1-\Pr[\text{alle Nachrichten eines } k\text{-Blocks liegen auf derselben Linie}])^{\lceil \frac{d}{k}\rceil}\\
&= (1-2\cdot 2^{-k})^{\lceil \frac{d}{k}\rceil}\\
&= (1-2^{-k+1})^{\lceil \frac{d}{k}\rceil}.
\end{aligned}$$

Da $(1-\frac{1}{n})^n \geq 2^{-2}$ für $n \geq 2$ und $k \geq 2$, erhalten wir:

$$\begin{aligned}
(1-2^{-k+1})^{\lceil \frac{d}{k}\rceil} &\geq 2^{-2\cdot 2^{-k+1}\cdot \lceil \frac{d}{k}\rceil}\\
&= 2^{-\frac{\lceil \frac{d}{k}\rceil}{2^{k-2}}}.
\end{aligned}$$

Mit dieser Abschätzung ist Ungleichung 6.8 erfüllt, wenn

$$-\frac{\lceil \frac{d}{k}\rceil}{2^{k-2}} \geq \log(1-\varepsilon). \tag{6.9}$$

Da $k \geq \log(2d)+\log(\frac{1}{\varepsilon})+2-\log\log(e)$, gilt

$$2^{k-2} \geq 2d\cdot\frac{1}{\varepsilon\cdot\log(e)},$$

somit auch

$$k\cdot 2^{k-2}\cdot\varepsilon\cdot\log(e) \geq 2d$$

und

$$\varepsilon\log(e) \geq \frac{\frac{2d}{k}}{2^{k-2}}.$$

Aus der Annahme $d \geq k$ folgt $\frac{2d}{k} \geq \frac{d}{k}+1 \geq \lceil \frac{d}{k}\rceil$ und

$$-\frac{\lceil \frac{d}{k}\rceil}{2^{k-2}} \geq -\varepsilon\log(e) = \log\left(2^{-\varepsilon\cdot\log(e)}\right) = \log\left(e^{-\frac{1}{\varepsilon^{-1}}}\right).$$

Da $\varepsilon^{-1} \geq 1$ und somit $e^{-1} \geq (1-\frac{1}{\varepsilon^{-1}})^{\varepsilon^{-1}}$, gilt

$$-\frac{\lceil \frac{d}{k}\rceil}{2^{k-2}} \geq \log\left(1-\frac{1}{\varepsilon^{-1}}\right) = \log(1-\varepsilon).$$

Damit sind die Ungleichungen 6.9 und 6.8 erfüllt. ∎

Für $k = \log(2d) + \log(\frac{1}{\varepsilon}) + 2$ und $d \geq 2\log(\frac{1}{\varepsilon}) + 10$ mit $1 \geq \varepsilon > 0$ gilt

$$d - \log(2d) \geq \frac{d}{2} \geq \log\left(\frac{1}{\varepsilon}\right) + 2$$

und somit

$$d \geq k = \log(2d) + \log\left(\frac{1}{\varepsilon}\right) + 2 = \log\left(\frac{8d}{\varepsilon}\right).$$

Unter Verwendung von Theorem 6.7, des Lemmas 6.9 und der Ungleichung 6.2 erhalten wir durch Einsetzen:

Theorem 6.11 *Seien $\tau \geq 0$ und $M'^0, M'^1, \ldots$ eine Nachrichtenfolge der Sammelstelle eingeschränkt auf die Nachrichten eines Benutzers und $\mathcal{S}$ der entsprechende Sammlungsgraph. Sei $k = \log(2d) + \log(\frac{1}{\varepsilon}) + 2$ für $0 < \varepsilon \leq 1$, $d \geq 2\log(\frac{1}{\varepsilon}) + 10$ und $c \geq 1$, so gilt:*

$$\Pr\left[E' \geq c \cdot \left(\frac{8d}{\varepsilon}\right)^2 \cdot \log\left(\frac{8d}{\varepsilon}\right)\right] \quad \leq \quad e^{-c}$$

und

$$\Pr[M'^{\tau}, \ldots, M'^{\tau+d-1} \text{ liegen in derselben Komponente von } \mathcal{S}] \geq 1 - \varepsilon.$$

6.6 Effiziente Entschlüsselung durch eine Union-Find-Struktur

In den letzten Abschnitten haben wir uns damit beschäftigt herauszufinden, wann die Historie der Nachrichten entkoppelt wird und bis wann wir mit bestimmter Sicherheit garantieren können, dass sie nicht entkoppelt wird. Nun wollen wir uns das Öffnen der kritischen Nachrichten genauer ansehen. Ist der Schwellwert von d kritischen Nachrichten erreicht und gehören die d Nachrichten zu demselben Zeitintervall, so kann die Sammelstelle diese Nachrichten entschlüsseln.

Wenn die Sammelstelle eine gute Strategie hat, die Nachrichten schnell ihrer Zusammenhangskomponente zuzuordnen, ist sie auch in der Lage anhand des Zeitstempels der Nachrichten zu erkennen, ob der Schwellwert von d kritischen Nachrichten erreicht worden ist. Durch Kenntnis einer solchen Zuordnung kann die Sammelstelle anschließend die d Nachrichten effizient öffnen.

Die Zusammenhangskomponenten des vereinigten Sammlungsgraphen sind disjunkt. Jede enthält jeweils nur die Nachrichten eines Benutzers. Zur effizienten Lösung dieses Problems bietet sich eine *Union-Find-Struktur* an. Die Struktur verwaltet die Zusammenhangskomponenten als disjunkte Mengen. Jede Menge in der Struktur hat eine eindeutige Nachricht als Repräsentanten.

6.6.1 Methoden der Struktur

Die Struktur kennt die folgenden Methoden:

- `FIND`(id) sucht die Nachricht M', so dass für ein i gilt, dass $\mathrm{ID}_i(M') = id$, und gibt den Repräsentanten R der Menge von M' zurück.
- `UNION`(R_1, R_2) vereinigt die Mengen S_{R_1} und S_{R_2} mit den Repräsentanten R_1 und R_2 und legt den Repräsentanten der vereinigten Menge S fest. Die Menge S wird der Struktur hinzugefügt, S_{R_1} und S_{R_2} werden entfernt.

- `INSERT`(M') fügt die neue Menge $\{M'\}$ mit Repräsentanten M' zu der Struktur hinzu.
- `DELETE`(M') löscht die Nachrichten M' aus der Struktur.

6.6.2 Typische Benutzung der Struktur

Die Sammelstelle hält den vereinigten Sammlungsgraphen und damit die Gesamtheit der kritischen Nachrichten in der Struktur. Sie aktualisiert die Struktur, wenn eine neue kritische Nachricht eintrifft oder wenn sie zu alte Daten löscht.

1. Wenn eine neue Nachricht M' die Sammelstelle erreicht, erstellt die Sammelstelle eine neue Menge $S_{M'}$ für die Nachricht durch den Aufruf der Methode `INSERT`(M').
2. Anschließend vereinigt sie alle Zusammenhangskomponenten, die in dem Sammlungsgraphen durch M' verbunden werden. Dazu führt die Sammelstelle für alle $i \in \{1,\ldots,k\}$ folgende Schritte durch:
   (a) $id = \mathrm{ID}_i(M')$
   (b) $id_{\mathrm{pre}} = \mathrm{IDpre}_i(M')$
   (c) $R =$ `FIND`(id)
   (d) $R_{\mathrm{pre}} =$ `FIND`(id_{pre})
   (e) `UNION`(R, R_{pre}).

   R ist der aktuelle Repräsentant der Menge, die M' enthält, und R_{pre} ist der Repräsentant der Menge, zu der die Nachricht mit der i-ten Vorgänger-ID gehört. Iterativ werden die Zusammenhangskomponenten durch den `UNION`-Befehl aktualisiert. Die Struktur ist dann konsistent mit den Zusammenhangskomponenten des vereinigten Sammlungsgraphen, der M' enthält.
3. Falls die Sammelstelle eine Nachricht nicht mehr benötigt, beispielsweise weil diese zu alt ist und zu viel Speicherplatz einnimmt, kann die Sammelstelle sie mittels `DELETE`(M') löschen.

6.6.3 Implementierung und Effizienz der Struktur

UNION-FIND-Strukturen gehören zu den grundlegenden algorithmischen Problemen. Bereits Aho, Hopcroft und Ullman [2] beschreiben, wie man die Operationen `UNION` und `FIND` über disjunkten Mengen implementieren kann. Gehen wir zunächst davon aus, dass die IDs der Nachrichten nicht zu große Werte annehmen, dann benötigen m `UNION`-/`FIND`-Operationen auf n_e Elementen (Nachrichten) in der Struktur $\mathcal{O}(m \cdot \alpha(m, n_e))$ Schritte [106, 26]. Dabei ist die Funktion α die Inverse der Ackermann-Funktion und wächst sehr langsam. Für die meisten in der Praxis vorkommenden Werte für m und n_e ist $\alpha(m, n_e) \leq 5$. Diese Abschätzungen gelten auch, wenn wir zusätzlich die Methoden `INSERT` und `DELETE` betrachten und $n_e = n_{\mathrm{maxID}}$ die maximale Anzahl von Elementen in der Struktur ist. Die Sammelstelle kann dafür sorgen, dass die Zahl n_{maxID} nicht zu groß wird. Sie kann kritische Nachrichten aus der Struktur löschen, wenn sie zu alt geworden sind.

In unserem Fall können die IDs große Werte annehmen. Somit können wir in der Struktur keine Liste verwenden, die einer ID die interne Referenz auf ein Element (Nachricht) zuordnet. Diese Liste würde den Speicher überlaufen lassen. Als Lösung für dieses Zuordnungsproblem in der `FIND`-Methode bieten sich zwei Möglichkeiten an [2]. Erstens, wir benutzen einen

balancierten Baum, zum Beispiel einen AVL-Baum oder Rot-Schwarz-Baum. Dann benötigen die Operationen zum Finden, Einfügen und Löschen einer ID $\mathcal{O}(\log(n_{\text{maxID}}))$ Schritte. Also können wir die Operationen `UNION`, `FIND`, `INSERT` und `DELETE` im Durchschnitt jeweils in $\mathcal{O}(\log(n_{\text{maxID}}))$ Schritten durchführen. Zweitens, wir benutzen eine Hashtabelle zum Speichern der Zuordnung der ID zur entsprechenden internen Referenz in der Struktur. Die drei Operationen zum Finden, Einfügen und Löschen einer ID benötigen dann im Durchschnitt eine konstante Laufzeit und folglich auch die vier Operationen auf der Struktur. Allerdings liegt die Laufzeit für die Operationen auf der Hashtabelle und somit auch auf der Struktur im Worst-Case bei $\mathcal{O}(n_{\text{maxID}})$. Wenn man die Kapazität der Hashtabelle groß genug wählt, tritt in der Praxis bei guter Wahl einer Hashfunktion der Worst-Case so gut wie nie auf.

Ein Praktiker wird sich vielleicht die Wahl abnehmen lassen und er wird eine effiziente Datenbank benutzen. Diese sorgt für eine effiziente Verwaltung und Indizierung der Daten. So können wir leicht alle IDs aus der Datenbank extrahieren, die vor einem bestimmten Zeitpunkt eingefügt wurden und damit veraltete Nachrichten repräsentieren, die wir gegebenenfalls löschen können.

Nachdem die Sammelstelle bei Eintreffen einer neuen kritischen Nachricht die Struktur und damit die Zusammenhangskomponenten des Sammlungsgraphen aktualisiert hat, kann sie überprüfen, ob der Schwellwert d erreicht wurde. Durch einen weiteren Aufruf der `FIND`-Methode finden wir schnell die Zusammenhangskomponente einer Nachricht. Nun kann die Sammelstelle die Zusammenhangskomponente in der Struktur durchsuchen. Die oben angesprochenen Techniken verwenden Bäume zur internen Speicherung einer Menge und führen außerdem Zähler über die Größen der Mengen. Wenn die Sammelstelle vor der Überprüfung alle alten Daten entfernt, kann sie direkt am Zähler der Komponente ablesen, ob der Schwellwert d erreicht wurde. Sie kann in diesem Fall die d Nachrichten einfach aus der Struktur extrahieren, wenn wir die Kanten des Baumes doppelt verkettet implementieren.

Im Folgenden nehmen wir an, dass die Sammelstelle nach Eintreffen einer Nachricht M' die Zusammenhangskomponenten aktualisiert und alte Daten löscht. Dann benötigt sie nicht mehr als $\mathcal{O}(k\log(n_{\text{maxID}}))$ Schritte, um die Zusammenhangskomponenten zu aktualisieren. Die Sammelstelle muss nur die Zusammenhangskomponenten aktualisieren und überprüfen, in denen keine Nachricht entschlüsselt wurde. Andernfalls kann die Sammelstelle die IDs der letzten entschlüsselten Nachrichten eines Nutzers speichern und bei Eintreffen von M' direkt entschlüsseln. Wir können diese IDs wieder in einem balancierten Baum speichern und in $\mathcal{O}(\log(n_{\text{maxID}}))$ Schritten feststellen, ob M' direkt entschlüsselt werden kann. Wir betrachten in der Struktur nur Mengen von Nachrichten, die noch nicht entschlüsselt wurden. Nach dem Löschen von alten Daten enthält keine dieser Mengen mehr als d Elemente, da ansonsten der Schwellwert im aktuellen Zeitintervall erreicht wurde und die Entschlüsselung möglich ist. Da wir zur Aktualisierung der Struktur und damit zur Vereinigung der Zusammenhangskomponenten die `UNION`-Methode k-mal anwenden und wir vor Aufruf der `UNION`-Methode die alten Nachrichten einer zu vereinigenden Menge löschen, enthält eine aktualisierte Menge höchstens $\mathcal{O}(k \cdot d)$ Nachrichten. Aus diesem Grund beträgt die Laufzeit zur Aktualisierung der Struktur für die Nachricht M' einschließlich der Löschung alter Daten und der Überprüfung auf Erreichen des Schwellwertes $\mathcal{O}(k \cdot (k \cdot d \cdot \log(n_{\text{maxID}}) + \log(n_{\text{maxID}}) + 1))$ $= \mathcal{O}(k^2 \cdot d \cdot \log(n_{\text{maxID}}))$ Schritte.

Im schlechtesten Fall konnte keiner der n Benutzer identifiziert werden und es gibt für jeden Benutzer $d-1$ nicht gelöschte Nachrichten im Sammlungsgraphen. Somit können wir für eine Sammelstelle, die regelmäßig alte Daten löscht, annehmen, dass $n_{\text{maxID}} \leq n \cdot d$.

Theorem 6.12 *Eine Sammelstelle, die regelmäßig alte Nachrichten löscht, kann bei Eingang*

einer kritischen Nachricht M' in $\mathcal{O}(k^2 \cdot d \cdot \log(n \cdot d))$ Schritten den vereinigten Sammlungsgraphen aktualisieren und die d zur Rekonstruktion nötigen Nachrichten identifizieren, falls durch M' der Schwellwert von d kritischen Nachrichten in einer Zusammenhangskomponente des vereinigten Sammlungsgraphen überschritten wurde.

Da wir $k \approx \log(d)$ wählen können und sich für diesen Wert mit hoher Wahrscheinlichkeit die d letzten Nachrichten innerhalb einer Zusammenhangskomponente befinden, ist das Verfahren mit logarithmischer Laufzeit in der Anzahl der Benutzer und fast linearer Laufzeit in dem Schwellwertparameter d je zu aktualisierender Nachricht sehr effizient.

6.6.4 Laufzeit und Speicherplatzbedarf in einem Fallbeispiel

Als Beispiel wählen wir ein großes System, das die kritischen Nachrichten von einer Milliarde Benutzern verwalten soll. Für den Schwellwert d wählen wir den Wert 1.000. Damit wir sicher sein können, dass die d letzten Nachrichten in derselben Zusammenhangskomponente liegen, setzen wir $\varepsilon = 0,01$. Um den Voraussetzungen von Theorem 6.11 zu genügen, wählen wir $k = 20$. Somit ist $n_{\text{maxID}} \leq 10^9 \cdot 10^3 = 10^{12}$. Die Länge der IDs soll 1.000 Bit betragen. Dadurch ist es möglich, die IDs der kritischen Nachrichten randomisiert zu wählen, ohne dass eine ID mit einer Wahrscheinlichkeit von größer als 10^{-100} doppelt vergeben wird. Dabei gehen wir davon aus, dass insgesamt weniger als 2^{80} Nachrichten gesendet werden. Pro Nachricht sind höchstens $c \cdot 1,6 \cdot 10^7$ Schritte für eine kleine Konstante c notwendig, um den Sammlungsgraphen zu aktualisieren. Bei einem 2-GHz-Prozessor beträgt damit die Laufzeit im Worst-Case etwa $0,032 \cdot c$ Sekunden. Der Speicherplatz für den Sammlungsgraphen beträgt im Worst-Case $c' \cdot n_{\text{maxID}} \cdot 1.000 = c' \cdot 10^{12} \cdot 1.000 = c' \cdot 10^{15}$ Bits, das entspricht $c' \cdot 125$ Terabytes für eine kleine Konstante c'. Diese Größe ist in Bezug auf die Anzahl der Benutzer vertretbar und wächst proportional zu dieser Anzahl. Dementsprechend werden für mehrere Millionen Benutzer bei den gleichen Parametern weniger als ein Terabyte benötigt.

6.7 Sicherheit bei verschiedenen Angreifertypen

Das Protokoll aus Abschnitt 6.2 arbeitet mit dem Provider als Trusted Party. In diesem Abschnitt betrachten wir einen allgemeineren Fall für die Sicherheit. Dazu sei angenommen, dass der Benutzer die Rolle des Providers übernimmt. Der Benutzer kann gegebenenfalls aktive (bösartige) Angriffe durchführen. Ist das Protokoll sicher bei einem passiven Benutzer, so folgt daraus direkt die Sicherheit bei Verwendung eines Providers. Die Betrachtung des allgemeineren Falls ermöglicht weitere Einsatzmöglichkeiten des Protokolls, zum Beispiel bei der Entkopplung der Historie im privaten Marketing.

Das Protokoll für zwei Nachrichtenlinien enthält bereits einige Sicherheitsmechanismen für die Sicherheit bei bösartigen Angreifern. Im Folgenden werden wir klären, in welchen Fällen diese Sicherheitsmechanismen ausreichen. Zudem werden wir die Protokolle so erweitern, dass bösartige Angriffe erkannt werden können. Bei dieser Analyse wird davon ausgegangen, dass bei gleichzeitiger Durchführung der Protokolle mit mehreren Benutzern und einer Sammelstelle alle Parteien allein arbeiten. Sie bilden keine Koalitionen und tauschen keine Information über die Protokolldurchführung hinaus aus.

Wir sehen unsere Protokolle zur Entkopplung der Historie als typische und nützliche Hilfsprotokolle für andere Szenarien wie beispielsweise der privaten Vorratsdatenspeicherung (siehe Kapitel 4) oder dem privaten Marketing (siehe Abschnitte 1.2.5 und 1.4). Sollte ein

bösartiger Benutzer bei einem Betrug oder einer Manipulation ertappt werden, wird dafür gesorgt, dass er keinen Zugriff mehr auf den gewünschten Dienst erhält. Aus diesem Grund ist der Benutzer bemüht, dass seine Angriffe schwer zu detektieren sind.

Bei passiven Angreifern können wir davon ausgehen, dass jede Partei dem durchzuführenden Protokoll ehrlich folgt. Keine Partei weicht vom Protokoll ab und verändert übertragene Daten. Ein passiver Angreifer analysiert alle während der Protokolldurchführung gesendeten und empfangenen Strings. Kann der passive Angreifer durch die Analyse seiner Kommunikation mehr über die Eingaben der anderen Parteien erfahren als sich aus seinen Eingaben und seinem Ergebnis des Protokolls erschließen lässt, dann ist das Protokoll auch nicht sicher gegen einen beliebigen realistischen Angreifer. Ein passiver Benutzer kann also lediglich Informationen über die Sammelstelle gewinnen. Er kann diese aber nicht im Protokollverlauf ausnutzen und sein Verhalten ändern. Ein passiver Benutzer entspricht in der privaten Vorratsdatenspeicherung aus Kapitel 4 einem vertrauenswürdigen Provider, der die Korrektheit der Daten auch bei einem bösartigen Benutzer garantieren kann.

Durch die Analyse passiver Angreifer erhalten wir eine obere Schranke für die Information, die ein außenstehender und die Kommunikation abhörender Beobachter bekommen kann. Passive Angreifer sind oft einfacher zu untersuchen als aktive Angreifer. Sie geben außerdem ein Gefühl für die Sicherheit eines Protokolls. Wenn das Protokoll bei einer passiven Attacke unsicher ist, dann ist es dies auch bei einer aktiven Attacke.

Im Gegensatz zu einem passiven Angreifer, kann ein bösartiger Angreifer Nachrichten beliebig fälschen, löschen oder hinzufügen. Dadurch kann sich seine Kommunikation von einer durch ehrliche Ausführung entstandenen Kommunikation unterscheiden. Die aktiv geänderten Nachrichten können unter Umständen die Berechnungen einer anderen Partei beeinflussen. Sie können sowohl der anderen Partei Informationen über ihre private Eingaben entlocken als auch das Ergebnis des Protokolls verändern und damit von der idealen Funktionalität abweichen lassen. Führt der Benutzer bösartige Angriffe aus, kann er direkt die Nachrichten beeinflussen, die an die Sammelstelle fließen. Für den Fall, dass der Benutzer Nachrichten löscht, gehen wir davon aus, dass er keinen Zugriff auf den angeforderten Dienst bekommt. Er wird ausgeschlossen. Fügt der Benutzer eine Nachricht hinzu und unterscheidet diese sich von der als nächstes zu erwartenden Nachricht nach dem Protokoll, erkennt die Sammelstelle entweder eine zusätzliche Nachricht oder wir können diesen Fall wie eine Manipulation behandeln.

6.8 Sicherheit bei passivem Benutzer

In diesem Abschnitt gehen wir davon aus, dass der Benutzer ein passiver Angreifer ist und dem Protokoll ehrlich folgt. Der Sammelstelle unterstellen wir, dass sie ein bösartiger Angreifer ist.

Zunächst betrachten wir folgendes vereinfachtes Protokoll für zwei Nachrichtenlinien.

Vereinfachtes Protokoll für zwei Nachrichtenlinien:

1. Der Benutzer generiert zufällig und unabhängig für $i \in \{0,1\}$ die eindeutigen IDs $\mathrm{ID}_0(M_{i,t}), \ldots, \mathrm{ID}_{k-1}(M_{i,t})$ und erstellt die Nachrichten $M_{0,t}$ und $M_{1,t}$.

2. Der Benutzer wählt ein zufälliges Bit π.

3. Der Benutzer sendet $M'^{t} = M_{\pi,t}$ an die Sammelstelle.

Wir betrachten hier noch einmal die ideale Funktionalität des Protokolls, siehe dazu auch Abschnitt 6.2.1. Die Sammelstelle erhält die Nachricht M'^t, die gleichverteilt aus den Nachrichten $M_{0,t}$ und $M_{1,t}$ gewählt wird.

Wenn wir die oben beschriebene Protokollvariante betrachten, sehen wir, dass die Information nur in eine Richtung fließt, vom Benutzer zur Sammelstelle. Die Kommunikation des Benutzers besteht demgemäß nur aus den Nachrichten an die Sammelstelle. Also kann der Benutzer nichts über die Daten der Sammelstelle erfahren. Die Sammelstelle erhält nur M'^t. An dieser Stelle lassen wir es zu, dass der Benutzer erfährt, von welcher Linie die Nachricht kommt.

Wir sehen leicht, dass es nicht darauf ankommt, ob die Sammelstelle bösartig ist oder nicht. Laut vereinfachtem Protokoll sendet sie keine Daten und kann daher weder die Berechnung des Benutzers beeinflussen noch das Ergebnis ändern. Die Sammelstelle kann folglich keine Nachrichten an den Benutzer löschen oder manipulieren. Sollte sie Nachrichten hinzufügen, erkennt der Benutzer den Angriff und kann die Nachricht ignorieren. Da der Benutzer keine Nachrichten von der Sammelstelle empfängt, kann der Benutzer nicht mehr Information gewinnen als im Fall einer passiven Sammelstelle. Wir können schließen:

Beobachtung 6.13 *Das vereinfachte Protokoll ist sicher bei passiven Benutzern.*

Betrachten wir nun das Protokoll für zwei Nachrichtenlinien nach Abschnitt 6.2. Da wir annehmen, dass das verwendete Oblivious-Transfer-Protokoll kryptographisch sicher ist, liefert der Aufruf des Oblivious-Transfer-Protokolls dem Benutzer kein Wissen über die Eingaben der Sammelstelle. Zudem folgt aus der Sicherheit des Oblivious-Transfer-Protokolls, dass die Sammelstelle entweder $M_{0,t}$ oder $M_{1,t}$, aber keine Information über die andere Nachricht erhält. Die Sammelstelle könnte allerdings die Wahrscheinlichkeit beeinflussen mit der die Nachricht einer Linie gewählt wird. Bei der idealen Funktionalität beträgt diese Wahrscheinlichkeit $\frac{1}{2}$. Da der Benutzer das Permutationsbit π gleichverteilt und unabhängig von σ wählt, wirkt π wie ein One-time Pad und $\pi \oplus \sigma$ ist ebenfalls gleichverteilt. Aus diesem Grund kann die Sammelstelle die Wahl der Linie von M'^t nicht beeinflussen.

Führen wir mehrere Iterationen des Protokolls zur Übertragung einer Nachrichtenfolge von Benutzer an die Sammelstelle aus, so kann eine Manipulation der Sammelstelle einer Iteration keine andere beeinflussen, da die Durchführungen des Protokolls unabhängig voneinander sind. Eine bösartige Sammelstelle kann die Protokolle nicht angreifen.

Theorem 6.14 *Die Protokolle zur zufälligen Auswahl von Nachrichten $M'^0, \ldots, M'^\omega$ aus zwei Linien und Übertragung dieser Nachrichtenfolge an die Sammelstelle ist sicher bei bösartiger Sammelstelle und passiven Benutzern.*

Korollar 6.15 *Die Protokolle zur zufälligen Auswahl von Nachrichten $M'^0, \ldots, M'^\omega$ aus zwei Linien und Übertragung dieser Nachrichtenfolge an die Sammelstelle ist privat, das heißt, sicher bei passiver Sammelstelle und passiven Benutzern.*

6.9 Sicherheit bei bösartiger Sammelstelle und bösartigem Benutzer

Das vereinfachte Protokoll ist nicht sicher gegen einen bösartigen Benutzer. Schon das Setzen des Permutationsbits π auf denselben konstanten Wert für alle Durchläufe sorgt dafür, dass der Sammlungsgraph nur aus isolierten Knoten besteht. Danach ist die gesamte

Historie des Benutzers entkoppelt. Denselben Effekt kann der Benutzer ebenfalls für das normale Protokoll für Nachrichtenlinien erreichen. Er kann als Eingabe für den Aufruf des Oblivious-Transfer-Protokolls anstatt verschiedener Nachrichten mehrfach die Nachricht derselben Linie verwenden. Die Sammelstelle erhält also nur Nachrichten einer Linie und kann diese Nachrichten nicht in Beziehung bringen.

Im Folgenden gehen wir darauf ein, welche Manipulation der Benutzer bei der Durchführung des ursprünglichen Protokolls vornehmen kann. Dies kann nur durch Veränderung der Werte für π und den Nachrichten $M_{i,t}$ geschehen. Ein mögliches Tauschen der Argumente für das Oblivious-Transfer-Protokoll können wir als Veränderung der Argumente betrachten. Das Ziel dieser Untersuchungen ist es, ein neues Protokoll zu entwickeln, welches gegen beliebige bösartige Angriffe robust ist.

6.9.1 Änderung des Permutationsbits π

Ist die Änderung des Permutationsbits π die einzige Attacke des Benutzers, dann kann er nicht die Verteilung der Linie der gesendeten Nachricht ändern, da die Sammelstelle das Selektionsbit σ wählen kann. Wählt die Sammelstelle σ gleichverteilt, dann ist auch $\pi \oplus \sigma$ gleichverteilt. Können wir annehmen, dass die Sammelstelle ehrlich ist, dann reicht es, wenn wir nur das Selektionsbit σ verwenden und π aus dem Protokoll streichen. Durch eine zufällige Wahl von π oder σ kann sowohl die Sammelstelle als auch der Benutzer sicherstellen, dass $\pi \oplus \sigma$ zufällig ist.

6.9.2 Kollision der IDs mehrerer Benutzer

Falls ein bösartiger Benutzer die IDs der Nachrichten ändert und gleichzeitig andere Benutzer über dasselbe Protokoll mit der Sammelstelle kommunizieren, dann ist es möglich, dass ein bösartiger Benutzer seine Historie mit denen von anderen verbindet. Wenn zum Beispiel eine Verweis-ID des bösartigen Benutzers identisch mit einer Nachrichten-ID eines anderen Benutzers ist, dann muss die Sammelstelle unter Umständen davon ausgehen, dass Nachrichten beider Benutzer aus einer Quelle stammen. Auf diese Weise kann ein bösartiger Benutzer einen Teil seiner Historie in der Historie eines anderen verstecken.

Wir haben bereits im Abschnitt 6.6 gesehen, dass bei randomisierter Wahl der IDs und die Wahrscheinlichkeit einer Kollision der IDs sehr gering ist. Wenn ein bösartiger Benutzer versucht, seine IDs zu fälschen, um dabei eine korrekte ID eines anderen Benutzers zu treffen, dann ist dieses nur äußerst selten erfolgreich. Da die Benutzer nicht zusammenarbeiten, kann ein bösartiger Benutzer auch nicht auf anderem Weg an die IDs anderer Benutzer gelangen. Bei der folgenden Analyse wird davon ausgegangen, dass ein bösartiger Benutzer es nicht schafft, IDs anderer Benutzer zu raten oder über einen anderen Weg zu bestimmen.

6.9.3 Mehrfache Benutzung desselben Wertes als Nachrichten-ID

In den nächsten Abschnitten wollen wir darauf eingehen, welchen Einfluss es hat, wenn der Benutzer die IDs der Nachrichten oder ihre Verweis-IDs (IDpre) ändert. Wir werden Erweiterungen unserer Protokolle vorstellen, durch die die Sammelstelle feststellen kann, ob der Benutzer vom richtigen Protokollablauf abweicht. Hier gehen wir zunächst zur Vereinfachung davon aus, dass es nur einen Benutzer gibt. Es findet zunächst keine Vermischung der Sammlungsgraphen verschiedener Benutzer statt. In der Bewertung der Angriffe muss allerdings die gemeinsame Durchführung des Protokolls bei mehreren Benutzern und einer

Sammelstelle beachtet werden. Wir betrachten in diesem Abschnitt die Manipulationsmöglichkeiten der Nachrichten-ID, wie eine solche Manipulation entdeckt werden kann und welche Maßnahme die Sammelstelle anschließend ergreifen kann.

In der Protokollbeschreibung gehen wir davon aus, dass alle Nachrichten-IDs eindeutig sind. Diese Bedingung werden wir an zwei Punkten lockern, ohne das gewünschte Verhalten des Protokolls zu beeinträchtigen. Da die Koordinierung von verschiedenen Nachrichten bezüglich der eindeutigen Wahl der Nachrichten-IDs schwierig sein kann, lassen wir es zu, dass die Nachrichten-IDs zufällig generiert werden. Da wir annehmen, dass ein bösartiger Benutzer die Werte anderer Benutzer nicht kennt, ist die Wahrscheinlichkeit, dass er eine Nachrichten-ID eines anderen Benutzers bestimmen kann, sehr klein. Ein bösartiger Benutzer ist somit nicht in der Lage (bei sehr kleiner Irrtumswahrscheinlichkeit), seine Nachrichten durch Manipulation der Nachrichten-ID als Nachricht eines anderen Benutzers auszugeben.

Als zweites relaxieren wir die Eindeutigkeitsbedingung der Nachrichten-ID wie folgt: Jeder Wert wird als Nachrichten-ID von höchstens einem Benutzer verwendet. Ein Benutzer verwendet einen Wert höchstens einmal pro Linie als Nachrichten-ID. Zuvor durfte der Benutzer einen Wert höchstens einmal insgesamt als Nachrichten-ID verwenden. Da es im Verbindungsgraphen nur Kanten zwischen Knoten unterschiedlicher Linien gibt, können auch bei dieser Relaxierung die Verweise in den Nachrichten den richtigen Knoten und Kanten zugeordnet werden. Die Struktur des Sammlungsgraphen wird somit nicht beeinflusst.

Außer einer zufälligen Kollision der Nachrichten-IDs von (verschiedenen) Benutzern, besteht die einzige nicht zulässige Wahl eines Wertes für eine Nachrichten-ID $\mathrm{ID}_b(M_{i,t})$ darin, dass es für einen Benutzer zwei unterschiedliche Nachrichten $M_{i,t}$ und $M_{i,t'}$ mit $t' < t$ und $(a,b) \in \{0,\ldots,k-1\}^2$ auf einer Linie i gibt, so dass $\mathrm{ID}_b(M_{i,t}) = \mathrm{ID}_a(M_{i,t'})$. In diesem Fall können Verweise von der Sammelstelle möglicherweise nicht richtig zugeordnet werden. Auf diese Weise kann ein Sammlungsgraph entstehen, der kein Subgraph des Verbindungsgraphen ist. Mit einer Wahrscheinlichkeit von mindestens $\left(\frac{1}{2}\right)^2$ empfängt die Sammelstelle sowohl $M'^t = M_{i,t}$ als auch $M'^{t'} = M_{i,t'}$. Daher kann die Sammelstelle diese Manipulation mit einer Wahrscheinlichkeit von $\left(\frac{1}{2}\right)^2$ erkennen.

Hat die Sammelstelle eine wiederholte Benutzung einer Nachrichten-ID auf einer Linie festgestellt, so kann sie mit großer Wahrscheinlichkeit davon ausgehen, dass es sich um keine zufällige Kollision zwischen den IDs verschiedener Benutzer handelt und dass eine Manipulation der Nachrichten-ID eines Benutzers vorliegt. Als nachfolgende Maßnahme kann die Sammelstelle nun die Nachrichtenannahme verweigern und damit beispielsweise dem Benutzer den Zugang zum angebotenen Dienst verweigern.

6.9.4 Mehrfache Benutzung desselben Wertes als Verweis-ID

Die mehrfache Benutzung desselben Wertes als Verweis-ID hat große Ähnlichkeit mit der mehrfachen Benutzung desselben Wertes als Nachrichten-ID. Da die Wahrscheinlichkeit einer Kollision von Verweis-IDs eines bösartigen Benutzers mit den Verweis-IDs anderer Benutzer sehr klein ist, können wir bei wiederholter Verwendung einer Verweis-ID auf einer Linie bei kleiner Fehlerwahrscheinlichkeit von einer Manipulation ausgehen. Damit erkennt die Sammelstelle anhand derselben Verweis-ID zweier Nachrichten, dass sie von demselben Benutzer stammen. Mit einer Wahrscheinlichkeit von $\left(\frac{1}{2}\right)^2$ empfängt die Sammelstelle die beiden Nachrichten mit derselben Verweis-ID und kann die Manipulation erkennen.

6.9.5 Benutzung einer unzulässigen Verweis-ID

Das Protokoll verlangt, dass für Linie jede Nachrichten-ID eindeutig ist. Eine unbenutzte ID bezeichnen wir in diesem Kontext als *neu*. Durch die Festlegung der Nachrichten-IDs sind alle Verweis-IDs bestimmt, da jede Verweis-ID Bezug auf eine Nachrichten-ID nimmt. Wenn ein bösartiger Benutzer eine Verweis-ID so ändert, dass ihr Wert keiner Nachrichten-ID entspricht, dann kann die Sammelstelle bei dem bisher vorgestellten Verfahren die entsprechende Verbindung im Sammlungsgraphen nicht herstellen. Ändert der Benutzer alle Verweis-IDs auf diese Weise, ist seine Historie vollkommen entkoppelt. Wir müssen also sicherstellen, dass

1. der Benutzer nur Verweis-IDs benutzt, die seinen Nachrichten-IDs entsprechen und
2. der Benutzer die Verweis-IDs an den richtigen Positionen verwendet.

Für diese beiden Punkte werden wir robuste Protokollerweiterungen vorstellen. Im Anschluss und im folgenden Abschnitt 6.9.6 kümmern wir uns um den ersten Punkt. Wir stellen sicher, dass der Benutzer nur Verweis-IDs benutzen kann, die zuvor als Nachrichten-ID verwendet wurden. Im Abschnitt 6.9.7 wird eine Erweiterung vorgestellt, die zudem den zweiten Punkt abdeckt.

Nach Abschnitt 6.9.3 wissen wir, dass ein bösartiger Benutzer mit großer Wahrscheinlichkeit dabei entdeckt wird, wenn er eine Nachrichten-ID doppelt verwendet. Wir gehen nun davon aus, dass er jede Nachrichten-ID nur einmal benutzt. Die Grundlage für die erste Protokollerweiterung bildet die blinde Signatur der IDs. Die Sammelstelle unterschreibt IDs und Linieninformation blind. Sie kann nach Erhalt einer Nachricht verifizieren, dass korrekte Nachrichten- und Verweis-IDs benutzt wurden. Zum Signieren benutzen wir ein sicheres Blinde-Signatur-Schema ohne Seitenkanal als Black-Box (siehe Abschnitt 3.6). Durch die Signaturen binden wir jede $\mathrm{ID}_j(M_{i,t})$ an die Linie i und Position j.

6.9.6 Protokollerweiterung 1: Signatur der IDs

Als ersten Schritt zu einem Protokoll, das unzulässige Werte für die Verweis-IDs erkennt, benutzen wir signierte Nachrichten. Mit $\mathrm{sig}M_{i,t}$ bezeichnen wir die Signatur einer Nachricht $M_{i,t}$. Die Signatur $\mathrm{sig}M_{i,t}$ besteht aus den signierten IDs von $M_{i,t}$:

$$\begin{aligned}\mathrm{sig}M_{i,t} \quad = \quad & M_{i,t} \\ & \circ (\mathrm{sig}(\langle i,0,\mathrm{ID}_0(M_{i,t})\rangle),\ldots,\mathrm{sig}(\langle i,k-1,\mathrm{ID}_{k-1}(M_{i,t})\rangle)) \\ & \circ (\mathrm{sig}(\langle i\oplus 1,0,\mathrm{IDpre}_0(M_{i,t})\rangle),\ldots,\mathrm{sig}(\langle i\oplus 1,k-1,\mathrm{IDpre}_{k-1}(M_{i,t})\rangle)).\end{aligned}$$

Dabei seien $\langle\cdot\rangle$ eine beliebige umkehrbare Paarungsfunktion, beispielsweise die Konkatenation $\circ$, und sig die Signaturfunktion der Sammelstelle, die mit einem sicheren Blinde-Signatur-Schema ohne Seitenkanal berechnet wird.

Protokollerweiterung 1

1. Der Benutzer generiert zufällig und unabhängig für $i \in \{0,1\}$ die eindeutigen IDs $\mathrm{ID}_0(M_{i,t+1}),\ldots,\mathrm{ID}_{k-1}(M_{i,t+1})$ für die nächste Iteration.
2. Der Benutzer erstellt die signierten Nachrichten $\mathrm{sig}M_{0,t}$ und $\mathrm{sig}M_{1,t}$.
3. Der Benutzer wählt ein zufälliges Bit π.

4. Die Sammelstelle wählt ein zufälliges Bit σ.

5. Die Sammelstelle erhält $\mathrm{sig}M'^{t} = \mathrm{OT}_{\sigma}(\mathrm{sig}M_{\pi,t}, \mathrm{sig}M_{\pi\oplus 1,t})$ durch sicheres Oblivious-Transfer von dem Benutzer.

6. Die Sammelstelle überprüft, ob die Signaturen aller IDs in $\mathrm{sig}M'^{t}$ korrekt sind. Falls nicht, bricht sie an dieser Stelle die Protokolldurchführung ab.

7. Für alle generierten $\mathrm{ID}_j(M_{i,t+1})$ mit $i \in \{0,1\}$ und $j \in \{0,\ldots,k-1\}$ signiert die Sammelstelle dem Benutzer blind den Wert $\langle i, j, \mathrm{ID}_j(M_{i,t+1})\rangle$. Der Benutzer erhält somit $\mathrm{sig}(\langle i, j, \mathrm{ID}_j(M_{i,t+1})\rangle)$.

Für die Initialisierung des Protokolls erhält der Benutzer für alle in der Initialisierung generierten IDs blinde Signaturen von der Sammelstelle. Anschließend verfügt der Benutzer demnach für $i \in \{0,1\}$, $\tau \in \{0,\ldots,k\}$ und $\xi \in \{\tau,\ldots,k\}$ über $\mathrm{sig}(\langle i, \xi-1, \mathrm{ID}_{\xi-1}(M_{i,-\tau})\rangle)$.

Die Sammelstelle kann bei jeder erhaltenen Nachricht überprüfen, ob die Signatur korrekt ist. Der Benutzer kann folglich nur signierte IDs als Nachrichten- und Verweis-IDs benutzen. In jeder Iteration des erweiterten Protokolls erhält der Benutzer signierte IDs für die folgende Nachricht. Dies geschieht jedoch erst, nachdem die Sammelstelle die Korrektheit der Signaturen überprüft hat. Stellt sich heraus, dass eine Signatur gefälscht ist, dann fehlen dem Benutzer die IDs für die nächste Nachricht. Zu Beginn der Übertragung der Nachrichten kann der Benutzer sich zwar aus der Menge der Initialisierungs-IDs bedienen, doch spätestens nach der k-ten Nachricht hat er bei der Fälschung einer Signatur keine unbenutzte signierte ID zur Verfügung. Es sei denn, er benutzt eine ID mehrfach. Diese Fälle haben wir bereits zuvor behandelt. Somit kann der Benutzer nur Verweis-IDs benutzen, die er auch als Nachrichten-ID benutzt.

Aus der signierten Nachricht $\mathrm{sig}M'^{t}$ kann die Sammelstelle nicht mehr Information über den Benutzer und seine Historie gewinnen als aus der entsprechenden Nachricht M'^{t}. Da das Signaturverfahren sicher, blind und frei von Seitenkanälen ist, kann die Sammelstelle aus den signierten IDs nur erkennen, dass der Benutzer über korrekte Verweis-IDs verfügt und diese auf der richtigen Linie eingesetzt werden. Aus den Eigenschaften des Signaturverfahrens folgt, dass eine bösartige Sammelstelle kein weiteres Wissen aus Schritt 7 der Protokollerweiterung lernen oder der Signatur weitere Information hinzufügen kann.

Der Benutzer kann durch die Protokollerweiterung ebenfalls kein weiteres Wissen gewinnen. Aus der Signatur der IDs kann er nur ableiten, ob die Sammelstelle den Wert des Benutzers korrekt unterschrieben hat. Da das Signaturverfahren sicher ist und wir es als Black-Box benutzen, erfährt der Benutzer keine privaten Daten der Sammelstelle aus der Anwendung dieses Verfahrens. Insgesamt gibt diese Protokollerweiterung einem bösartigen Benutzer und einer bösartigen Sammelstelle kein weiteres Wissen über die Historie des Benutzers oder des Sammlungsgraphen als das Protokoll ohne Erweiterung.

6.9.7 Protokollerweiterung 2: Überprüfung der Verweis-IDs

Auch wenn der Benutzer nicht beliebig viele neue IDs benutzen kann, so kann er allerdings die Verweis-IDs an nicht vorgesehenen Positionen verwenden. Im Folgenden werden wir eine zweite Erweiterung des Protokolls vorstellen, über die die Sammelstelle die Möglichkeit erhält, den richtigen Einsatz der Verweis-IDs zu überprüfen. Dazu kann die Sammelstelle anhand einer Stichprobe der Historie feststellen, ob die richtige Verweis-ID für den j-ten Vorgänger der Nachricht benutzt wurde. Damit diese Stichprobe der Sammelstelle kein weiteres

Wissen über die Historie des Benutzers vermittelt als im ursprünglichen Protokollverlauf, muss die Sammelstelle im Tausch für die Überprüfung auf die Historieninformation der aktuellen Nachricht verzichten. Somit kostet der Test auf die Korrektheit der Verweis-IDs eine Nachricht, die zum Zusammenhalt der Historie beiträgt.

Für diese Erweiterung müssen wir lediglich Schritt 5 der Protokollerweiterung 1 anpassen.

Protokollerweiterung 2

Sei $(\mathcal{K}, E, D)$ ein sicheres symmetrisches Kryptosystem mit Schlüsselgenerator $\mathcal{K}$, Verschlüsselungsfunktion E_K und Entschlüsselungsfunktion D_K bezüglich des Schlüssels K.

5. (a) Der Benutzer generiert zufällig die Schlüssel K und $K_{i,j}$ für $i \in \{0,1\}$ und $j \in \{0,\ldots,k-1\}$.

   (b) Der Benutzer erstellt für $i \in \{0,1\}$ die verschlüsselten Nachrichten $\widetilde{\text{sig}}M_{i,t}$ mit

   $$\widetilde{\text{sig}}M_{i,t} = E_K(\text{sig}M_{i,t}) \circ E_K(K_{i,0},\ldots,K_{i,k-1}).$$

   (c) Die Sammelstelle erhält

   $$\widetilde{\text{sig}}M'^t = \text{OT}_\sigma(\widetilde{\text{sig}}M_{\pi,t}, \widetilde{\text{sig}}M_{\pi\oplus 1,t})$$

   durch sicheres 1-2-Oblivious-Transfer von dem Benutzer.

   (d) Für $i \in \{0,1\}$ und $j \in \{0,\ldots,k-1\}$ sendet der Benutzer

   $$E_{K_{i,j}}(\text{IDpre}_j(M_{i,t}) \circ \text{sig}(\langle i \oplus 1, j, \text{IDpre}_j(M_{i,t})\rangle))$$

   an die Sammelstelle.

   (e) Die Sammelstelle wählt $\kappa \in \{0,\ldots,k\}$.

   (f) Sei $K_j = (K_{0,j}, K_{1,j})$. Die Sammelstelle erhält

   $$K' = \text{OT}_\kappa(K_0,\ldots,K_{k-1},K)$$

   durch sicheres 1-(k+1)-Oblivious-Transfer vom Benutzer.

   (g) Falls $\kappa \in \{0,\ldots,k-1\}$, dann entschlüsselt die Sammelstelle für $i \in \{0,1\}$ $\text{IDpre}_\kappa(M_{i,t})$ mit $K_{i,\kappa}$ und überprüft, ob die $\text{IDpre}_\kappa(M_{i,t})$ mit dem Sammlungsgraphen konsistent ist und ihre Signaturen gültig sind. Falls eine der Überprüfungen fehlschlägt, dann bricht die Sammelstelle das Protokoll ab. Andernfalls wird das Protokoll in Schritt 6 fortgesetzt.

   (h) Falls $\kappa = k$, dann kann die Sammelstelle $\text{sig}M'^t = \text{sig}M_{\pi\oplus\sigma,t}$ sowie $K_{\pi\oplus\sigma,0},\ldots,$ $K_{\pi\oplus\sigma,k-1}$ entschlüsseln und überprüfen, ob die in Schritt 5d empfangenen Werte für $\text{IDpre}_j(M_{i,t})$ und $\text{sig}(\langle i\oplus 1, j, \text{IDpre}_j(M_{i,t})\rangle)$ mit den Werten in $\text{sig}M'^t$ übereinstimmen. Falls eine der Überprüfungen fehlschlägt, dann bricht die Sammelstelle das Protokoll ab. Andernfalls wird das Protokoll in Schritt 7 fortgesetzt.

Unter der Annahme, dass das in Schritt 5f eingesetzte 1-(k+1)-Oblivious-Transfer sicher ist, erhält die Sammelstelle für $\kappa = k$ nur den Schlüssel K. Darauf kann sie $\widetilde{\text{sig}}M'^t = \widetilde{\text{sig}}M_{\pi\oplus\sigma,t}$ entschlüsseln und bekommt $\text{sig}M'^t$ sowie $K_{\pi\oplus\sigma,0},\ldots,K_{\pi\oplus\sigma,k-1}$. Folglich kann die Sammelstelle für $j \in \{0,\ldots,k-1\}$ die Verweis-IDs $\text{IDpre}_j(M_{\pi\oplus\sigma,t})$ und die entsprechende

Signatur $\mathrm{sig}(\langle \pi \oplus \sigma \oplus 1, j, \mathrm{IDpre}_j(M_{\pi\oplus\sigma,t})\rangle))$ sowohl aus $\mathrm{sig}M'^t$ als auch aus den in Schritt 5d empfangenen verschlüsselten Werten bestimmen und die Gültigkeit der jeweiligen Signaturen überprüfen. Zudem kann die Sammelstelle testen, ob der Benutzer in den Schritten 5c und 5d konsistente Werte übertragen hat. Wählt die Sammelstelle σ uniform und gilt $\kappa = k$, dann wird eine Inkonsistenz mit Wahrscheinlichkeit $\frac{1}{2}$ entdeckt.

Da wir annehmen, dass die Oblivious-Transfer-Protokolle sicher sind, können wir schlussfolgern, dass die Sammelstelle aus den Schritten 5c und 5f kein Wissen über $\mathrm{sig}M_{\pi\oplus\sigma\oplus1,t}$ und $K_{\pi\oplus\sigma\oplus1,0},\ldots,K_{\pi\oplus\sigma\oplus1,k-1}$ gewinnt. Unter der Annahme, dass das verwendete Kryptosystem sicher ist, gewinnt die Sammelstelle auch aus den anderen Schritten kein weiteres Wissen über $M_{\pi\oplus\sigma\oplus1,t}$. Damit folgt:

Lemma 6.16 *Wählt die Sammelstelle $\kappa = k$, dann lässt sich aus der Durchführung der zweiten Protokollerweiterung nicht mehr Wissen über den Verbindungsgraphen des Benutzers gewinnen als bei der Durchführung der ersten Protokollerweiterung.*

Wir repräsentieren eine Übertragung von Nachrichten eines Benutzers an die Sammelstelle mittels der Protokollerweiterung 2 durch eine *erweiterte Nachrichtenfolge* (M'^0,κ_0), $(M'^1,\kappa_1),\ldots,(M'^\omega,\kappa_\omega)$. Dabei entsprechen die Nachrichten M'^j einer Nachrichtenfolge bei Ausführung des Protokolls ohne Erweiterung. Der Wert κ_j gibt die Wahl des Wertes für κ bei der j-ten Iteration des Protokolls in der zweiten Erweiterung an. Analog zum ursprünglichen Protokoll beschreiben wir das Wissen über den Verbindungsgraphen eines Benutzers als Sammlungsgraph, der durch eine erweiterte Nachrichtenfolge induziert wird.

Lemma 6.17 *Sei $t \geq 0$ und $(M'^0,\kappa_0),\ldots,(M'^\omega,\kappa_\omega)$ eine erweiterte Nachrichtenfolge der Sammelstelle. Falls $\kappa_t \in \{0,\ldots,k-1\}$, dann ist M'^t mit höchstens einem Knoten $M'^{t'}$ mit $t' \neq t$ im Sammlungsgraphen, der durch $(M'^0,\kappa_0),\ldots,(M'^\omega,\kappa_\omega)$ induziert wird, direkt verbunden.*

BEWEIS Für $\kappa_t \in \{0,\ldots,k-1\}$ folgt analog aus der Sicherheit der Verschlüsselung und der Oblivious-Transfer-Protokolle, dass die Sammelstelle in Schritt 5 nur Wissen über $\mathrm{IDpre}_{\kappa_t}(M_{i,t})$ und die entsprechende Signatur $\mathrm{sig}(\langle i\oplus1, j, \mathrm{IDpre}_{\kappa_t}(M_{i,t})\rangle))$ mit $i \in \{0,1\}$ gewinnen kann. Wissen über die anderen Einträge von $M_{i,t}$ für $i \in \{0,1\}$ können aus der Kommunikation nicht gewonnen werden. Dies gilt insbesondere für die Nachrichten-IDs dieser Knoten. Somit kann die Sammelstelle nachfolgende Nachrichten nicht über Verweis-IDs mit den Nachrichten $M_{0,t}$ und $M_{1,t}$ in Verbindung bringen, da diese Werte der Sammelstelle nicht bekannt sind.

Aus dem Protokoll folgt, dass die Gleichheit $\mathrm{ID}_{\kappa_t}(M'^{t-\kappa_t-1}) = \mathrm{IDpre}_{\kappa_t}(M_{i,t})$ nur für $i = \mathrm{line}(M'^{t-\kappa_t-1})\oplus 1$ gilt. Falls die Sammelstelle $\kappa_{t-\kappa_t-1} = k$ gewählt hat, kennt sie $\mathrm{ID}_{\kappa_t}(M'^{t-\kappa_t-1})$. Somit ist M'^t in dem Sammlungsgraphen nur mit $M'^{t-\kappa_t-1}$ direkt verbunden.

Gilt $\kappa_{t-\kappa_t-1} \in \{0,\ldots,k-1\}$, dann hat die Sammelstelle keine Nachricht erhalten, die $\mathrm{IDpre}_{\kappa_t}(M_{i,t})$ für ein $i \in \{0,1\}$ als Nachrichten-ID enthält. Folglich ist M'^t in diesem Fall mit keinem anderen Knoten im Sammlungsgraphen direkt verbunden. ∎

Um zu untersuchen, wie der Sammlungsgraph durch den Ablauf der zweiten Protokollerweiterung verändert wird, betrachten wir eine erweiterte Nachrichtenfolge $(M'^0,\kappa_0),\ldots,$ $(M'^\omega,\kappa_\omega)$ und analysieren die unterschiedlichen Abläufe für ein $t \geq 0$:

- Ablauf 1 für $\kappa_t = k$ und
- Ablauf 2 für $\kappa_t \in \{0,\ldots,k-1\}$

Für eine gegebene erweiterte Nachrichtenfolge $(M'^0, \kappa_0), \ldots, (M'^\omega, \kappa_\omega)$ bezeichne S_1 den Sammlungsgraphen für Ablauf 1 und S_2 den Sammlungsgraphen für Ablauf 2. Zudem sei S_* der entsprechende Sammlungsgraph zu der erweiterten Nachrichtenfolge $(M'^0, \kappa_0), \ldots, (M'^{t-1}, \kappa_{t-1}), (M'^{t+1}, \ldots, \kappa_{t+1}), (M'^\omega, \kappa_\omega)$. Somit wird in S_* alles Wissen über M'^t ignoriert.

Für einen ungerichteten Graphen G und einen Knoten v aus G bezeichne $G[v]$ die Zusammenhangskomponente von v in G.

Beobachtung 6.18 *Sei $t \geq 0$ und $(M'^0, \kappa_0), \ldots, (M'^\omega, \kappa_\omega)$ die erweiterte Nachrichtenfolge der Sammelstelle. Dann gilt für jeden Knoten $M \in S_*$, dass*

$$S_*[M] \subseteq S_1[M].$$

BEWEIS Da für S_1 gilt, dass $\kappa_t = k$, empfängt die Sammelstelle nach Lemma 6.16 die Nachricht M'^t. Da für S_* die Nachricht M'^t ignoriert wird und die restliche erweiterte Nachrichtenfolge für S_1 und S_* übereinstimmt, folgt die Aussage direkt. ∎

Lemma 6.19 *Sei $t \geq 0$ und $(M'^0, \kappa_0), \ldots, (M'^\omega, \kappa_\omega)$ die erweiterte Nachrichtenfolge der Sammelstelle. Sei $\kappa_t \in \{0, \ldots, k-1\}$ für Ablauf 2. Dann gilt für alle Knoten $M' \in M'^0, \ldots, M'^\omega$,*

- *falls $M' \in S_*[M'^{t-\kappa_t-1}] \cup \{M'^t\}$, dass*

$$S_2[M'] \subseteq S_*[M'] \cup \{M'^t\} = S_*[M'^{t-\kappa_t-1}] \cup \{M'^t\},$$

- *falls $M' \notin S_*[M'^{t-\kappa_t-1}] \cup \{M'^t\}$, dass*

$$S_2[M'] \subseteq S_*[M'].$$

BEWEIS Da $\kappa_t \in \{0, \ldots, k-1\}$, gilt nach Lemma 6.17, dass der Knoten M'^t höchstens mit einem Knoten in S_2 direkt verbunden ist. Wenn M'^t einen Nachbarn in S_2 hat, ist $M'^{t-\kappa_t-1}$ der einzige Nachbar von M'^t in S_2. Folglich gilt für alle $M' \in S_*[M'^{t-\kappa_t-1}] \cup \{M'^t\}$, dass

$$S_2[M'] = S_2[M'^{t-\kappa_t-1}] \subseteq S_*[M'^{t-\kappa_t-1}] \cup \{M'^t\} = S_*[M'] \cup \{M'^t\}.$$

Zudem folgt, dass für Knoten $M' \notin S_*[M'^{t-\kappa_t-1}]$ die Zusammenhangskomponenten in S_2 und S_* identisch sind. ∎

Aus Beobachtung 6.18 und Lemma 6.19 können wir direkt folgern:

Theorem 6.20 *Sei $t \geq 0$ und $(M'^0, \kappa_0), \ldots, (M'^\omega, \kappa_\omega)$ die erweiterte Nachrichtenfolge der Sammelstelle. Sei $\kappa_t \in \{0, \ldots, k-1\}$ für Ablauf 2. Für alle Knoten $M' \in M'^0, \ldots, M'^\omega$ gilt dann,*

- *falls $M' \in S_1[M'^{t-\kappa_t-1}] \cup \{M'^t\}$, dass*

$$S_2[M'] \subseteq S_1[M'^{t-\kappa_t-1}] \cup \{M'^t\} = S_1[M'] \cup \{M'^t\},$$

- *falls $M' \notin S_1[M'^{t-\kappa_t-1}] \cup \{M'^t\}$, dass*

$$S_2[M'] \subseteq S_1[M'].$$

Wählt die Sammelstelle $\kappa_t \in \{0,\ldots,k-1\}$, so kann sie die Zusammenhangskomponente $S_1[M'^{t-\kappa_t-1}]$ aus dem vorherigen Lemma höchstens um einen Knoten vergrößern. Dieser Knoten hat nach Lemma 6.17 höchstens $M'^{t-\kappa_t-1}$ als direkten Nachbarn in dem Sammlungsgraphen. Dies gilt auch, wenn die Sammelstelle später weitere Nachrichten empfängt. Unter der Annahme, dass die Verschlüsselung und die Oblivious-Transfer-Protokolle sicher sind, haben wir bereits gesehen, dass die Sammelstelle für $\kappa_t \in \{0,\ldots,k-1\}$ in der zweiten Protokollerweiterung außer den Verweis-IDs zu den Nachrichten $M_{i,t-\kappa_t-1}$ mit $i \in \{0,1\}$ kein weiteres Wissen über $M_{i,t-\kappa_t-1}$ gewinnt. Dieses gilt insbesondere für $\text{load}(M_{i,t-\kappa_t-1})$. Insofern tauscht die Sammelstelle bei diesem Test der κ_t-ten Verweis-ID die Information über den Eintrag $\text{load}(M'^t)$ gegen die Zuordnung der Nachricht M'^t zu dem Vorgänger $M'^{t-\kappa_t-1}$. Da die getestete Nachricht höchstens einen Nachbarn im Sammlungsgraphen hat, verzichtet die Sammelstelle ferner darauf, über die getestete Nachricht spätere Nachrichten in Zusammenhang mit den vorherigen Nachrichten zu bringen. Für die Anwendung in der privaten Vorratsdatenspeicherung bedeutet dies, dass bei jedem Test der Sammelstelle auf ein Share zum Erreichen des Schwellwertes verzichtet werden muss.

Wir werden nun untersuchen, ob die zweite Protokollerweiterung den Zusammenhang der Nachrichten erhöht, die für die Identifikation eines Benutzers in der Vorratsdatenspeicherung nutzbar sind. Diese nutzbaren Nachrichten sind alle Nachrichten M'^t mit $\kappa_t = k$. Für eine erweiterte Nachrichtenfolge $F = (M'^0,\kappa_0),\ldots,(M'^\omega,\kappa_\omega)$ sei $F_{[\kappa=k]}$ die Menge aller Nachrichten M'^t der Folge F, so dass $\kappa_t = k$. Die Menge $F_{[\kappa=k]}$ beschreibt alle Nachrichten der Folge F, die bei der Identifikation und Entschlüsselung nutzbar sind. Ferner sei S_F der Sammlungsgraph, den wir aus der erweiterten Nachrichtenfolge F erhalten.

Theorem 6.21 *Seien $F = (M'^0,\kappa_0),\ldots,(M'^\omega,\kappa_\omega)$ und $\tilde{F} = (M'^0,k),\ldots,(M'^\omega,k)$ zwei erweiterte Nachrichtenfolgen der Nachrichtenfolge $M'^0,\ldots,M'^\omega$. Seien S_F und $S_{\tilde{F}}$ die entsprechenden Sammlungsgraphen. Dann gilt für alle Nachrichten $M' \in F_{[\kappa=k]}$, dass*

$$S_F[M'] \cap F_{[\kappa=k]} \subseteq S_{\tilde{F}}[M'].$$

BEWEIS Bei einem ehrlichen Benutzer entspricht der Sammlungsgraph $S_{\tilde{F}}$ nach Lemma 6.16 genau dem Sammlungsgraphen der Durchführung des Protokolls ohne Erweiterungen, bei dem die Nachrichtenfolge $M'^0,\ldots,M'^\omega$ übertragen wurde. Für $a \in \{1,\ldots,\omega\}$ sei $F_a = (M'^0,k),\ldots,(M'^{a-1},k),(M'^a,\kappa_a),\ldots,(M'^\omega,\kappa_\omega)$. Zudem seien $F_0 = F$ und $F_{\omega+1} = \tilde{F}$. Sukzessive betrachten wir nun für $a \in \{0,\ldots,\omega\}$ die Nachrichtenfolgen F_a und F_{a+1}. Falls $\kappa_a = k$, dann gilt $F_a = F_{a+1}$ und somit für alle $M' \in M'^0,\ldots,M'^\omega$, dass

$$S_{F_a}[M'] \cap F_{[\kappa=k]} \subseteq S_{F_a}[M'] = S_{F_{a+1}}[M'].$$

Falls $\kappa_a \in \{0,\ldots,k-1\}$, dann ist $M'^a \notin F_{[\kappa=k]}$. Somit gilt nach Theorem 6.20 für alle $M' \in M'^0,\ldots,M'^\omega$, dass

$$S_{F_a}[M'] \cap F_{[\kappa=k]} \subseteq S_{F_{a+1}}[M'].$$

■

Die Anwendung der zweiten Protokollerweiterung vergrößert somit nicht die Zusammenhangskomponenten der kritischen Daten, die für die Identifikation und Entschlüsselung nutzbar sind.

Nun werden wir untersuchen, ob durch die zweite Protokollerweiterung eine Fälschung der Verweis-IDs festgestellt werden kann. Die Sammelstelle kann anhand der Signatur der Verweis-IDs $\text{IDpre}_{\kappa_t}(M_{i,t})$ validieren, dass diese Verweis-IDs für die Position κ_t und Linie i zulässig sind. Zudem kann sie überprüfen, ob eine Verweis-ID konsistent mit der Nachrichten-ID von $M'^{t-\kappa_t-1}$ ist, das heißt, ob $\text{ID}_{\kappa_t}(M'^{t-\kappa_t-1}) = \text{IDpre}_{\kappa_t}(M_{i,t})$ für $i = \text{line}(M'^{t-\kappa_t-1}) \oplus 1$

gilt. Unter der Voraussetzung, dass die Sammelstelle die Nachricht $M'^{t-\kappa_t-1}$ vollständig empfangen hat, erkennt sie die Verwendung eines falschen Verweises in den Nachrichten $M_{0,t}$ und $M_{1,t}$ mit einer Wahrscheinlichkeit von mindestens $\frac{1}{2} \cdot \frac{1}{k}$. Angenommen, die Sammelstelle testet eine Nachricht mit einer Wahrscheinlichkeit von p_{test}. Dann beträgt die Wahrscheinlichkeit, dass die Sammelstelle die Nachricht M'^t testet und die Nachricht $M'^{t-\kappa_t-1}$ mit $\kappa \in \{0,\ldots,k-1\}$ nicht testet, $p_{\text{test}} \cdot (1-p_{\text{test}})$. Somit ist die Wahrscheinlichkeit, dass sie die Verwendung eines falschen Verweises bei der Übertragung einer Nachricht erkennt mindestens $p_{\text{test}} \cdot (1-p_{\text{test}}) \cdot \frac{1}{2} \cdot \frac{1}{k}$. Somit folgt aus diesem und den vorherigen Abschnitten direkt:

Theorem 6.22 *Sei $t \geq 0$ und $k \geq 2$. Sei p_{test} die Wahrscheinlichkeit, dass die Sammelstelle sich bei Durchführung der zweiten Protokollerweiterung für einen Test einer Verweis-ID entscheidet, das heißt, $\kappa_t \in \{0,\ldots,k-1\}$. Dann ist die Wahrscheinlichkeit, dass die Sammelstelle eine Manipulation von M'^t erkennt mindestens $p_{\text{test}} \cdot (1-p_{\text{test}}) \cdot \frac{1}{2} \cdot \frac{1}{k}$.*

7

Entkopplung der Historie durch mehrere Nachrichtenlinien

Im Kapitel 6 konnten wir sehen, dass wir durch ein effizientes Verfahren unter Verwendung von zwei Nachrichtenlinien sowohl die Entkopplung der Historie als auch die Möglichkeit zur Entschlüsselung während der Vorhaltezeit mit großer Sicherheit garantieren können. Jeder Knoten einer Linie verfügte dabei über k verschiedene IDs, auf die die nachfolgenden k Knoten der anderen Linie verwiesen. Im Folgenden werden wir die Eigenschaften eines auf mehrere Linien erweiterten Verbindungsgraphen untersuchen. Der Verbindungsgraph mit zwei Linien $\mathcal{M}_0$ und $\mathcal{M}_1$ wird dazu auf einen Graphen mit $\ell \geq 3$ Linien $\mathcal{M}_0, \ldots, \mathcal{M}_{\ell-1}$ verallgemeinert.

In dem Verbindungsgraphen mit zwei Nachrichtenlinien wurden k IDs pro Knoten benutzt. Würde nur eine ID je Knoten verwendet, dann würden alle Knoten einer Linie, die auf denselben Vorgänger verweisen, auch dieselbe Verweis-ID IDpre enthalten und könnten anhand dieser ID einander zugeordnet werden. Für das in Kapitel 6 vorgestellte Protokoll (2-Linien-Protokoll) würde die Verwendung einer ID pro Knoten zur Folge haben, dass der Sammlungsgraph immer zusammenhängend ist. Eine Entkopplung würde also nicht stattfinden. Besteht der Verbindungsgraph aus $\ell > 2$ Nachrichtenlinien und hat jeder Knoten nur eine ID, dann ist es möglich, dass der Sammlungsgraph in mehrere Komponenten zerfällt (siehe Abschnitt 7.2). Durch die Verwendung von einer ID anstelle von k IDs je Knoten reduzieren wir die Kommunikationskomplexität um etwa die Hälfte, da eine Nachricht in diesem Fall $k+1$ anstelle von $2k$ IDs enthält.

7.1 Verbindungsgraph und Sammlungsgraph für ℓ Nachrichtenlinien

Der *Verbindungsgraph* $\mathcal{G}^\ell = (V, E)$ für ℓ Nachrichtenlinien ist ein ungerichteter Graph. Die Knotenmenge $V = \mathcal{M}_0 \cup \ldots \cup \mathcal{M}_{\ell-1}$ besteht aus ℓ disjunkten Nachrichtenlinien $\mathcal{M}_i$ mit

$$\mathcal{M}_i = \{M_{i,0}, M_{i,1}, \ldots\}.$$

Eine Nachricht $M \in \mathcal{M}_i$ mit

$$M = (i, \mathrm{ID}, \mathrm{IDpre}_0, \ldots, \mathrm{IDpre}_{k-1}, \mathrm{load})$$

enthält die eindeutige Nachrichten-ID $\mathrm{ID}(M)$ sowie k Verweis-IDs (IDpre) auf die IDs der k vorhergehenden Nachrichten der Linie $(i-1) \bmod \ell$. Für alle $t \geq 0$, $i \in \{0,\ldots,\ell-1\}$ und $j \in \{1,\ldots,k\}$ gilt:

$$\mathrm{ID}(M_{i,t}) = \mathrm{IDpre}_{j-1}(M_{i+1 \bmod \ell, t+j}).$$

Für eine Nachricht $M_{i,t}$ bezeichnen wir eine Nachricht $M_{i',t'}$ als

- *Vorgänger von* $M_{i,t}$, falls $i' = i-1 \bmod \ell$ und $t' \in \{t-k,\ldots,t-1\}$. Die Menge aller Vorgänger von $M_{i,t}$ bezeichnen wir mit $M\mathrm{pre}(M_{i,t})$.
- *Nachfolger von* $M_{i,t}$, falls $i' = i+1 \bmod \ell$ und $t' \in \{t+1,\ldots,t+k\}$. Die Menge aller Nachfolger von $M_{i,t}$ bezeichnen wir mit $M\mathrm{suc}(M_{i,t})$.
- *Gefährte von* $M_{i,t}$, falls $i' = i$ und $t' \in \{t-k+1,\ldots,t-1,t+1,t+k-1\}$. Die Menge aller Gefährten von $M_{i,t}$ bezeichnen wir mit $M\mathrm{com}(M_{i,t})$.

Lemma 7.1 *In dem Verbindungsgraphen $\mathcal{G}^\ell$ enthalten zwei Nachrichten $M_{i,t} \neq M_{i',t'}$ genau dann einen gleichen Wert in ihren* ID- *und* IDpre-*Einträgen, falls $M_{i',t'}$ ein Vorgänger, Nachfolger oder Gefährte von $M_{i,t}$ ist.*

BEWEIS Ist $M_{i',t'}$ ein Vorgänger von $M_{i,t}$, so gilt nach obiger Definition

$$\mathrm{IDpre}_{t-t'-1}(M_{i,t}) = \mathrm{ID}(M_{i',t'}).$$

Analog gilt für einen Nachfolger $M_{i',t'}$ von $M_{i,t}$, dass

$$\mathrm{IDpre}_{t'-t-1}(M_{i',t'}) = \mathrm{ID}(M_{i,t}).$$

Da $i' = i$ und $|t-t'| \leq k-1$, gilt für einen Gefährten $M_{i',t'}$ nach Definition, dass $M_{i-1 \bmod \ell, t''}$ mit $t'' = \min(t,t') - 1$ ein Vorgänger sowohl von $M_{i,t}$ als auch von $M_{i',t'}$ ist. Somit haben Gefährten mindestens einen gemeinsamen Vorgänger und es gilt:

$$\mathrm{IDpre}_{t-t''-1}(M_{i,t}) = \mathrm{IDpre}_{t'-t''-1}(M_{i',t'}).$$

Alle anderen Nachrichten sind weder Vorgänger, Nachfolger noch Gefährte von $M_{i,t}$. Da zudem die IDs der Nachrichten eindeutig sind, folgt, dass andere Nachrichten in keinem Wert ihrer ID- oder IDpre-Einträge mit einer der IDs in $M_{i,t}$ übereinstimmen. ∎

Die Kanten des Verbindungsgraphen für ℓ Linien orientieren sich an der direkten Zuordenbarkeit der IDs:

$$\{M_{i,t}, M_{i',t'}\} \in E \quad \Longleftrightarrow \quad M_{i',t'} \in M\mathrm{pre}(M_{i,t}) \cup M\mathrm{suc}(M_{i,t}) \cup M\mathrm{com}(M_{i,t}).$$

Sei $M'^0, M'^1, \ldots$ eine Folge von Knoten, so dass $M'^\tau \in \{M_{0,\tau},\ldots,M_{\ell-1,\tau}\}$ für alle $\tau \geq 0$ gilt. Analog zum 2-Linien-Protokoll wird der Subgraph $\mathcal{S}^\ell$ des Verbindungsgraphen $\mathcal{G}^\ell$, der durch die Knotenfolge $M'^0, M'^1, \ldots$ induziert wird, *Sammlungsgraph* genannt.

In den folgenden Abschnitten werden wir untersuchen, unter welchen Umständen der Sammlungsgraph in mehrere Zusammenhangskomponenten zerfällt. Wie beim 2-Linien-Protokoll betrachten wir die Knotenfolge $M'^0, M'^1, \ldots, M'^\omega$ und somit auch den induzierten Sammlungsgraphen als Zufallsvariable. Dabei nehmen wir an, dass jede Nachricht M'^τ uniform aus der Menge $\{M_{0,\tau},\ldots,M_{\ell-1,\tau}\}$ gewählt wird. Analog zu Kapitel 6 werden im Weiteren die Blockzerlegungen der Folge $M'^0, M'^1, \ldots, M'^\omega$ betrachtet.

7.2 Entkopplung des Sammlungsgraphen für ℓ Nachrichtenlinien

Bei einer *Entkopplung* zum Zeitpunkt τ liegen alle Nachrichten vor dem Zeitpunkt τ in anderen Zusammenhangskomponenten als die Nachrichten, die zum Zeitpunkt τ oder danach gesendet wurden. Im Fall von zwei Nachrichtenlinien entspricht der Zerfall des Sammlungsgraphen in mehrere Zusammenhangskomponenten einer Entkopplung.

Beobachtung 7.2 *Für $\ell \geq 3$ Linien und $k \geq 2$ kann der Sammlungsgraph einer Nachrichtenfolge $M'^0, M'^1, \ldots, M'^\omega$ in mehrere Zusammenhangskomponenten zerfallen, ohne dass eine Entkopplung stattfindet.*

Um dieses Verhalten zu untersuchen, betrachten wir zunächst isolierte Knoten. Ein Knoten M des Sammlungsgraphen $\mathcal{S}^\ell$ heißt *isoliert*, wenn er keinen Nachbarn in $\mathcal{S}^\ell$ hat. Nach Lemma 7.1 ist ein Knoten ein Nachbar von M, entweder wenn dieser Vorgänger, Nachfolger oder Gefährte von M ist.

BEWEIS (BEOBACHTUNG 7.2) Seien $M'^t = M_{i,t}$,

$$M'^0, \ldots, M'^{t-1} = M_{i+1 \bmod \ell,0}, \ldots, M_{i+1 \bmod \ell,t-1} \quad \text{und}$$
$$M'^{t+1}, \ldots, M'^\omega = M_{i+2 \bmod \ell,t+1}, \ldots, M_{i+2 \bmod \ell,\omega}.$$

Da $\ell \geq 3$ ist, ist keiner der Knoten M'^τ mit $\tau \neq t$ Vorgänger, Nachfolger oder Gefährte von M'^t. Die Knoten $M'^0, \ldots, M'^{t-1}$ sowie $M'^{t+1}, \ldots, M'^\omega$ sind hingegen Gefährten von M'^t. Zudem ist M'^{t+1} ein Nachfolger von M'^{t-1}. Damit liegen alle Knoten außer M'^t in einer Zusammenhangskomponente. Folglich ist der Sammlungsgraph an keiner Stelle entkoppelt. Er zerfällt jedoch in mehrere Zusammenhangskomponenten, da M'^t ein isolierter Knoten ist. ∎

Für $\ell \geq 3$ Linien ist es demnach möglich, dass einige kritische Nachrichten eine eigene Zusammenhangskomponente bilden, ohne dass die Historie entkoppelt wird.

Entsprechend der Analyse in Abschnitt 6.1 leiten wir nun hinreichende Bedingungen für den Zusammenhalt und die Entkopplung des Sammlungsgraphen her. Wie im Fall von zwei Linien gibt $\text{line}(M)$ die Linie des Knotens M an.

Lemma 7.3 *Sei $B = M_1, \ldots, M_{2k}$ ein Block in der $2k$-Blockzerlegung der Nachrichtenfolge $M'^0, \ldots, M'^\omega$, so dass $\text{line}(M_1) = \ldots = \text{line}(M_k)$ und $\text{line}(M_{k+1}), \ldots, \text{line}(M_{2k}) \notin \{i, i+1 \bmod \ell\}$, wobei $i = \text{line}(M_1)$. Dann ist der durch die Nachrichtenfolge $M'^0, \ldots, M'^\omega$ induzierte Sammlungsgraph zum Zeitpunkt $\text{time}(M_{k+1})$ entkoppelt.*

BEWEIS Wir können feststellen, dass die Nachrichten $M_{k+1}, \ldots, M_{2k}$ weder Gefährten noch Nachfolger der ersten k Knoten in B sind. Somit gibt es keine Verbindung zwischen einem Knoten der ersten Hälfte und einem der zweiten Hälfte in B. Da die erste Hälfte von B k Knoten enthält, besitzt kein Knoten M'^a mit $\text{time}(M'^a) \leq \text{time}(M_k)$ eine Verbindung zu einem Knoten M'^b mit $\text{time}(M'^b) \geq \text{time}(M_{k+1})$. Folglich ist der Sammlungsgraph zum Zeitpunkt $\text{time}(M_{k+1})$ entkoppelt. ∎

Die Wahrscheinlichkeit, dass ein $2k$-Block eine Entkopplung bewirkt, beträgt folglich:

$$\ell \cdot \left(\frac{1}{\ell}\right)^k \cdot \left(1 - \frac{2}{\ell}\right)^k \geq \ell \cdot \left(\frac{1}{\ell}\right)^k \cdot \left(1 - \frac{2}{3}\right)^k = \frac{1}{3} \cdot \left(\frac{1}{3\ell}\right)^{k-1}.$$

Damit ist die Wahrscheinlichkeit, dass in einem $2k$-Block keine Entkopplung stattfindet, kleiner oder gleich $1 - \frac{1}{3} \cdot \left(\frac{1}{3\ell}\right)^{k-1}$. Sei EL die Zufallsvariable für die Wartezeit bis eine Entkopplung eintritt. Für $EL = i$ findet in den $2k$-Blöcken $B_0, \ldots, B_{i-2}$ keine Entkopplung statt. Der Sammlungsgraph ist jedoch in Block B_{i-1} entkoppelt. Damit gilt:

$$\Pr[EL = i] \geq \frac{1}{3} \cdot \left(\frac{1}{3\ell}\right)^{k-1} \cdot \left(1 - \frac{1}{3} \cdot \left(\frac{1}{3\ell}\right)^{k-1}\right)^{i-1}.$$

Aus den Eigenschaften der geometrischen Verteilung können wir somit ableiten, dass

$$\mathbb{E}(EL) \leq 3 \cdot (3\ell)^{k-1} \quad \text{und} \quad \Pr[EL \geq a] \leq \left(1 - \frac{1}{3} \cdot \left(\frac{1}{3\ell}\right)^{k-1}\right)^{a-1} \quad \text{für } a \geq 1.$$

Theorem 7.4 *Mit Wahrscheinlichkeit $1 - e^{-c}$ ist der durch eine Nachrichtenfolge M'^0, $\ldots, M'^\omega$ induzierte Sammlungsgraph mit ℓ Nachrichtenlinien spätestens nach $6kc \cdot (3\ell)^{k-1}$ Nachrichten entkoppelt.*

BEWEIS Nach den obigen Vorüberlegungen gilt, dass

$$\begin{aligned}
\Pr[EL \geq c \cdot 3(3\ell)^{k-1}] &\leq \left(1 - \frac{1}{3} \cdot \left(\frac{1}{3\ell}\right)^{k-1}\right)^{c \cdot 3(3\ell)^{k-1} - 1} \\
&\leq \left(1 - \frac{1}{3} \cdot \left(\frac{1}{3\ell}\right)^{k-1}\right)^{c \cdot 3(3\ell)^{k-1}} \\
&\leq e^{-c}.
\end{aligned}$$

Da sich die Definition von EL auf Blöcke der Größe $2k$ bezieht, folgt die Aussage des Lemmas. ∎

7.3 Zusammenhalt des Sammlungsgraphen für ℓ Nachrichtenlinien

In diesem Abschnitt leiten wir eine hinreichende Bedingung für den Zusammenhalt des Sammlungsgraphen her und bestimmen die Zuordenbarkeit von d aufeinander folgenden Nachrichten in Abhängigkeit der Parameter k und ℓ. Die hinreichende und notwendige Bedingung für den Zusammenhalt bei zwei Nachrichtenlinien in einem k-Block ist ein Wechsel zwischen den Nachrichtenlinien. Den Begriff des Wechsels können wir für ℓ Nachrichtenlinien zum Begriff der *aufsteigenden Folge* erweitern. Sei $B = M_1, \ldots, M_k$ ein k-Block der Nachrichtenfolge $M'^0, \ldots, M'^\omega$. Eine *aufsteigende Folge* existiert in B, falls es eine Folge von paarweise verschiedenen Indizes $i_1, \ldots, i_\ell$ gibt, so dass

1. für alle $j \in \{1, \ldots, \ell\}$ gilt, dass $1 \leq i_j \leq k$ und
2. für alle $j \in \{1, \ldots, \ell - 1\}$ gilt, dass $\text{line}(M_{i_{j+1}}) = \text{line}(M_{i_j}) + 1 \bmod \ell$.

In Abbildung 7.1 ist eine aufsteigende Folge eines k-Blocks für $k = 20$ dargestellt.

Lemma 7.5 *Sei $k \geq \ell$ und $M'^0, \ldots, M'^\omega$ die Nachrichtenfolge der Sammelstelle. Gibt es eine aufsteigende Folge in einem k-Block B der Nachrichtenfolge $M'^0, \ldots, M'^\omega$, dann gehören alle Knoten in B zu derselben Zusammenhangskomponente des entsprechenden Sammlungsgraphen $\mathcal{S}$.*

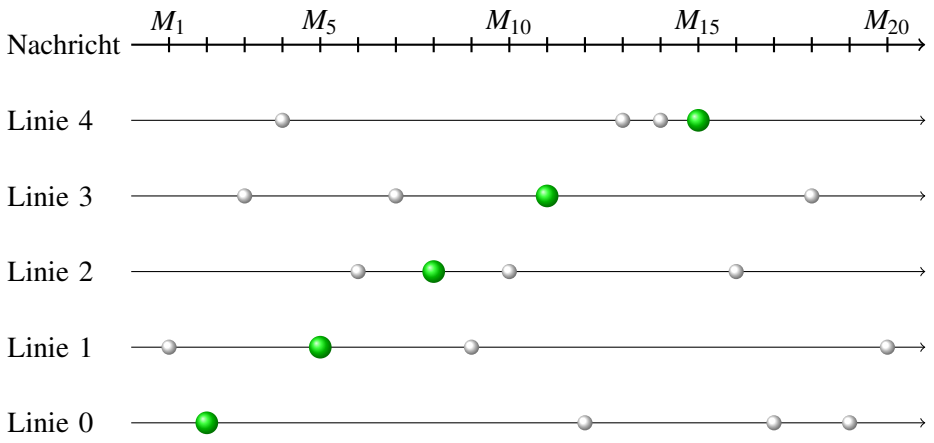


Abbildung 7.1: eine aufsteigende Folge des 20-Blocks $M_1,\ldots,M_{20}$

BEWEIS Sei $B = M_1,\ldots,M_k$ und $M_{i_1},\ldots,M_{i_\ell}$ eine aufsteigende Folge in B. Dann gilt für $j \in \{1,\ldots,\ell-1\}$, dass $\mathrm{ID}(M_{i_j}) = \mathrm{IDpre}_{i_{j+1}-i_j-1}(M_{i_{j+1}})$. Somit ist $M_{i_{j+1}}$ ein Nachfolger von M_{i_j}. Beide Nachrichten gehören demnach zu derselben Zusammenhangskomponente in $\mathcal{S}$. Da es innerhalb des k-Blocks B ℓ Knoten gibt, die sich auf ℓ unterschiedlichen Linien befinden und zusammenhängend sind, sind die verbleibenden Knoten in B Gefährten von einem dieser ℓ Knoten. Nach Lemma 7.1 sind alle Knoten in B zusammenhängend. ∎

Lemma 7.6 *Seien $k \geq \ell$, $M'^0,\ldots,M'^\omega$ die Nachrichtenfolge der Sammelstelle und $B_0,B_1,\ldots$ die k-Blockzerlegung der Nachrichtenfolge. Falls für eine Sequenz $B_j,B_{j+1},\ldots,B_{j+a}$ mit $a \geq 1$ von aufeinander folgenden k-Blöcken in jedem dieser Blöcke eine aufsteigende Folge existiert, dann gehören alle Knoten der Blöcke $B_j,B_{j+1},\ldots,B_{j+a}$ zu derselben Zusammenhangskomponente im entsprechenden Sammlungsgraphen.*

BEWEIS Wir zeigen zunächst, dass die Knoten der Blöcke B_j und B_{j+1} zu derselben Zusammenhangskomponente gehören. Sei M' die letzte Nachricht in dem Block B_j. Da B_{j+1} k Nachrichten und eine aufsteigende Folge enthält, gibt es in B_{j+1} einen Knoten M'' mit $\mathrm{line}(M'') = \mathrm{line}(M') + 1$. Damit ist M'' ein Nachfolger von M' und beide liegen in derselben Zusammenhangskomponente. Nach Lemma 7.5 gehören folglich alle Knoten aus B_j und B_{j+1} zu derselben Zusammenhangskomponente. Induktiv folgt für alle weiteren Blöcke der Sequenz, dass ihre Knoten ebenfalls zu derselben Zusammenhangskomponente gehören. ∎

Nun schätzen wir die Wahrscheinlichkeit für das Auftreten eines Blocks mit einer aufsteigenden Folge ab. Dadurch, dass ein Block mehrere aufsteigende Folgen enthalten kann, wird die kombinatorische Analyse erschwert. Daher betrachten wir das Auftreten von Blöcken mit einer *frühesten aufsteigenden Folge*. Eine aufsteigende Folge mit den Indizes $i_1,\ldots,i_\ell$ eines k-Blocks $B = M_1,\ldots,M_k$ nennen wir eine *früheste aufsteigende Folge* in B, falls

1. $i_1 = 1$ und
2. für alle $j \in \{1,\ldots,\ell-1\}$ und $i_j < \tau < i_{j+1}$ gilt, dass

$$\mathrm{line}(M_\tau) \neq \mathrm{line}(M_{i_{j+1}}) \bmod \ell.$$

Das bedeutet, $M_{i_{j+1}}$ ist der erste Nachfolger von M_{i_j}.

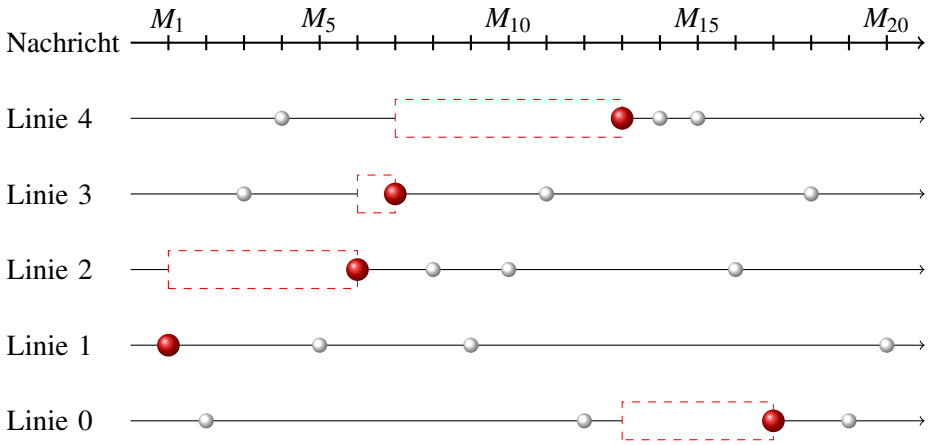


Abbildung 7.2: früheste aufsteigende Folge des 20-Blocks $M_1,\ldots,M_{20}$ aus Abbildung 7.1

Für eine früheste aufsteigende Folge mit den Indizes $i_1,\ldots,i_\ell$ sei

$$s = 1 + \sum_{j=2}^{\ell} \delta_j = i_\ell \leq k$$

mit $\delta_j = i_j - i_{j-1}$. Der Wert $s = i_\ell$ gibt dann die *Länge* dieser Folge an. Die früheste aufsteigende Folge eines Blocks ist eindeutig. In Abbildung 7.2 ist die früheste aufsteigende Folge des Blocks aus Abbildung 7.1 dargestellt. Aus der zweiten Bedingung, die eine früheste aufsteigende Folge erfüllen muss, folgt, dass keine Nachrichten in den markierten Bereichen vor den Nachrichten der Folge liegen.

Die Wahrscheinlichkeit für einen Block mit frühesten aufsteigende Folge, die durch die Parameter $i_1,\ldots,i_\ell$ repräsentiert wird, beträgt

$$\begin{aligned} &1 \cdot \left(1-\frac{1}{\ell}\right)^{\delta_2-1} \cdot \frac{1}{\ell} \cdot \left(1-\frac{1}{\ell}\right)^{\delta_3-1} \cdot \frac{1}{\ell} \cdot \ldots \cdot \left(1-\frac{1}{\ell}\right)^{\delta_\ell-1} \cdot \frac{1}{\ell} \cdot 1^{k-(1+\sum_j \delta_j)} \\ &= \left(\frac{1}{\ell}\right)^{\ell-1} \cdot \left(1-\frac{1}{\ell}\right)^{(1+\sum_j \delta_j)-\ell}. \end{aligned}$$

Die erste Nachricht $M_{i_1} = M_1$ kann auf jeder Linie liegen (Wahrscheinlichkeit 1). Für $j \in \{2,\ldots,\ell\}$ dürfen die Nachrichten zwischen der $(j-1)$-ten Nachricht $M_{i_{j-1}}$ und der j-ten Nachricht M_{i_j} der Folge nicht auf der Linie von M_{i_j} liegen (Wahrscheinlichkeit $(1-\frac{1}{\ell})^{\delta_j}$). Die Nachricht M_{i_j} muss sich auf der Linie $\text{line}(M_1)+(j-1) \bmod \ell$ befinden (Wahrscheinlichkeit $\frac{1}{\ell}$). Die verbleibenden Nachrichten des Blocks zu beliebigen Linien gehören. Sei $\ell \leq s \leq k$, dann gibt es $\binom{s-2}{\ell-2}$ verschiedene Folgen $i_1,\ldots,i_\ell$, die eine früheste aufsteigende Folge mit Länge s repräsentieren. Somit beträgt die Wahrscheinlichkeit für das Auftreten eines Blocks mit einer frühesten aufsteigenden Folge

$$\sum_{s=\ell}^{k} \binom{s-2}{\ell-2} \cdot \left(\frac{1}{\ell}\right)^{\ell-1} \cdot \left(1-\frac{1}{\ell}\right)^{s-\ell}.$$

Sei FF_i die binäre Zufallsvariable, die für $FF_i = 1$ beschreibt, dass der k-Block B_i eine früheste aufsteigende Folge enthält. Setzen wir $r = \ell - 1$ und verschieben wir den Index s um -1, dann gilt:

$$\begin{aligned}\Pr[FF_i = 1] &= \sum_{s=\ell}^{k} \binom{s-2}{\ell-2} \cdot \left(\frac{1}{\ell}\right)^{\ell-1} \cdot \left(1-\frac{1}{\ell}\right)^{s-\ell} \\ &= \sum_{s=r}^{k-1} \binom{s-1}{r-1} \cdot \left(\frac{1}{\ell}\right)^{r} \cdot \left(1-\frac{1}{\ell}\right)^{s-r}.\end{aligned}$$

Somit entspricht das Ereignis $FF_i = 1$ dem Ereignis $N_{r,p} \leq k-1$ einer Zufallsvariable $N_{r,p}$, die negativ binomialverteilt mit den Parametern r und $p = \frac{1}{\ell}$ ist. Folglich gilt:

$$\Pr[FF_i = 0] \;=\; \Pr[N_{r,p} > k-1] \;=\; \Pr[B_{k-1,p} < r] \;=\; \Pr[B_{k-1,1/\ell} < \ell - 1] \qquad (7.1)$$

für eine mit den Parametern $k-1$ und p binomialverteilte Zufallsvariable $B_{k-1,p}$.

Analog zur Analyse des Sammlungsgraphen für zwei Nachrichtenlinien untersuchen wir nun, wie wahrscheinlich es ist, dass d aufeinander folgende kritische Nachrichten in derselben Zusammenhangskomponente liegen. Somit können alle d Nachrichten einander zugeordnet werden und stehen zur Identifikation und Entschlüsselung bereit. Es geht keine Nachricht „verloren“.

Theorem 7.7 *Sei $M'^0, M'^1, \ldots$ eine Nachrichtenfolge, $\tau \geq 0$, $c > 0$, $\ell \geq 3$ und $d \geq k$. Ist $k > 2\ell^2 + 2\ell \cdot (c + \ln(d) - 1)$, dann gilt für den Sammlungsgraphen $\mathcal{S}^\ell$:*

$$\Pr[M'^\tau, \ldots, M'^{\tau+d-1} \textit{liegen in derselben Zusammenhangskomponente}] \geq 1 - e^{-c}.$$

BEWEIS Wir betrachten die k-Blockzerlegung $B_0, \ldots, B_{\lceil d/k \rceil - 1}$ der Nachrichtenfolge M'^τ, $\ldots, M'^{\tau + k \cdot \lceil d/k \rceil}$. Wenn in jedem der Blöcke eine aufsteigende Folge existiert, gehören nach Lemma 7.6 die Nachrichten $M'^\tau, \ldots, M'^{\tau+d-1}$ zu derselben Zusammenhangskomponente. Die Wahrscheinlichkeit, dass in jedem Block eine aufsteigende Folge vorkommt, beträgt

$$\Pr\left[\bigcap_{i=0}^{\lceil d/k \rceil - 1} FF_i = 1\right] = 1 - \Pr\left[\bigcup_{i=0}^{\lceil d/k \rceil - 1} FF_i = 0\right].$$

Da $k > 2\ell^2 + 2\ell \cdot (c + \ln(d) - 1)$, gilt auch $k > 2\ell^2 - 2\ell$ und für

$$\delta = 1 - \frac{\ell(\ell-1)}{k-1}$$

gilt $\frac{1}{2} \leq \delta < 1$. Da alle FF_i identisch verteilt sind, erhalten wir mit der Union-Bound, der Gleichung 7.1 und der Chernoff-Schranke, dass

$$\begin{aligned}\Pr\left[\bigcup_{i=0}^{\lceil d/k \rceil - 1} FF_i = 0\right] &\leq \sum_{i=0}^{\lceil d/k \rceil - 1} \Pr[FF_i = 0] \\ &= \lceil d/k \rceil \cdot \Pr[FF_0 = 0] \\ &= \lceil d/k \rceil \cdot \Pr[B_{k-1,1/\ell} < \ell - 1] \\ &= \lceil d/k \rceil \cdot \Pr[B_{k-1,1/\ell} < (1-\delta) \cdot \frac{k-1}{\ell}] \\ &= \lceil d/k \rceil \cdot \Pr[B_{k-1,1/\ell} < (1-\delta) \cdot \mathbb{E}(B_{k-1,1/\ell})] \\ &\leq \lceil d/k \rceil \cdot e^{-\delta^2 \cdot \frac{k-1}{2\ell}} \\ &\leq d \cdot e^{-\delta^2 \cdot \frac{k-1}{2\ell}} \\ &= e^{-\delta^2 \cdot \frac{k-1}{2\ell} + \ln(d)}.\end{aligned}$$

Wählen wir $k > 2\ell^2 + 2\ell \cdot (c + \ln(d) - 1)$, dann gilt ebenfalls

$$k \geq 2\ell^2 - 2\ell + 2\ell(c + \ln(d)) + 1$$

und
$$(k-1) - 2\ell \cdot (\ell - 1) \geq 2\ell \cdot (c + \ln(d)).$$

Addieren wir $\frac{\ell^2(\ell-1)^2}{(k-1)} > 0$ auf der linken Seite, so impliziert diese Ungleichung

$$\left(1 - \frac{2\ell(\ell-1)}{k-1} + \frac{\ell^2(\ell-1)^2}{(k-1)^2}\right) \cdot (k-1) \geq 2\ell \cdot (c + \ln(d))$$

sowie
$$\left(1 - \frac{\ell(\ell-1)}{k-1}\right)^2 \cdot \frac{k-1}{2\ell} \geq c + \ln(d).$$

Aus der letzten Ungleichung ergibt sich, dass

$$-\delta^2 \cdot \frac{k-1}{2\ell} + \ln(d) \leq -c$$

und somit

$$\Pr\left[\bigcup_{i=0}^{\lceil d/k \rceil - 1} FF_i = 0\right] \leq e^{-c}.$$

Die Aussage des Theorems folgt unmittelbar aus dieser Ungleichung. ∎

Fassen wir die Ergebnisse zur Entkopplung (Theorem 7.4) und zum Zusammenhalt des Sammlungsgraphen für ℓ Nachrichtenlinien (Theorem 7.7) zusammen, so erhalten wir:

Korollar 7.8 *Sei $\ell \geq 3$, $\tau \geq 0$, $d > 0$, $c > 0$, $k > 2\ell^2 + 2\ell \cdot (c + \ln(d) - 1)$ und $M'^0, M'^1, \ldots, M'^\omega$ eine Nachrichtenfolge der Sammelstelle eingeschränkt auf die Nachrichten eines Benutzers. Sei $\mathcal{S}^\ell$ der entsprechende Sammlungsgraph. Mit Wahrscheinlichkeit $1 - e^{-c}$ ist dann $\mathcal{S}^\ell$ spätestens nach $6kc \cdot (3\ell)^{k-1}$ Nachrichten entkoppelt und mit Wahrscheinlichkeit $1 - e^{-c}$ liegen die Nachrichten $M'^\tau, \ldots, M'^{\tau+d-1}$ in derselben Komponente von $\mathcal{S}^\ell$.*

7.4 Vergleich der Protokolle für zwei und für mehrere Nachrichtenlinien

Vergleichen wir die Protokolle für zwei und für mehrere Nachrichtenlinien, so können wir feststellen, dass jeweils ein Wert von $k \in \mathcal{O}(\log(d))$ ausreicht, um den Zusammenhalt von d aufeinander folgenden Nachrichten sicherzustellen. Im Gegensatz zum Fall mit zwei Nachrichtenlinien kann ein Knoten im Sammlungsgraphen für mehrere Nachrichtenlinien einen Gefährten haben. Dadurch erhöht sich der Zusammenhalt. Die Anzahl der Nachrichten bis zur Entkopplung ist für $k \in \mathcal{O}(\log(d))$ bei zwei Nachrichtenlinien quadratisch in d. Da $3\ell = 2^{\log(3\ell)} > 2^3$ ist diese Anzahl für mehrere Nachrichtenlinien mindestens kubisch in d. Für mehrere Nachrichtenlinien sind also mehr Nachrichten notwendig bis eine Entkopplung stattfindet. Dafür reduziert sich in diesem Fall die Anzahl der benötigten zufälligen IDs pro Nachricht von k auf 1.

Bezüglich der Sicherheit ist das Protokoll, das mehrere Nachrichtenlinien einsetzt, sicher gegen einen passiven Benutzer und eine bösartige Sammelstelle. Dies gilt analog zum 2-Linien-Protokoll. Da im Fall von mehreren Nachrichtenlinien mehr als eine Nachricht

auf dieselbe ID verweist, können wir hier die zweite Protokollerweiterung zum Schutz der Verweis-IDs nicht einsetzen. Zur Überprüfung der Korrektheit der Verweis-IDs können wir die Sammelstelle durch $\ell - 1$ vertrauenswürdige Datenbanken $DB_1, \ldots, DB_{\ell-1}$ erweitern. Empfängt die Sammelstelle eine Nachricht $M_{i,t}$, dann ist es möglich, dass die Datenbank DB_j die Nachricht $M_{i+j \bmod \ell,t}$ durch Techniken des sogenannten Private-Computation erhält. Da der Verbindungsgraph symmetrisch bezüglich der Linien ist, hat jede Datenbank einen ähnlich aufgebauten Sammlungsgraphen wie die Sammelstelle. Für alle Datenbanken sind insbesondere die Zusammenhangskomponenten ähnlich. Eine Entkopplung findet für alle gleichzeitig statt. Bei Empfang einer Nachricht und unter analoger Verwendung von signierten IDs können Sammelstelle und Datenbanken feststellen, ob nur erlaubte IDs verwendet wurden. Empfängt die Sammelstelle einen Nachfolger einer bereits erhaltenen Nachricht, so können alle Datenbanken testen, ob die Verweis-IDs der Nachricht korrekt sind. Somit ist eine Veränderung der Verweis-IDs nicht möglich. Ein Nachteil dieser Erweiterung ist jedoch, dass hierbei auf mehrere vertrauenswürdige Datenbanken zurückgegriffen werden muss.

8 Fazit

Eine Vorratsdatenspeicherung, die den berechtigten Interessen von Datensammlern an Nutzerdaten einerseits und den berechtigten Interessen der Benutzer am Schutz ihrer Daten und Ihrer Anonymität andererseits Rechnung trägt, ist möglich. Insbesondere dürfen die Daten bei den Datensammlern vorgehalten werden, ohne dass diese unberechtigt Zugriff erlangen können. Weiterhin kann die Kenntnis der Nutzungshistorie der Benutzer begrenzt werden, so dass alte Daten unbrauchbar werden.

Wir werden im Folgenden auf Erweiterungen und offene Probleme in der privaten Vorratsdatenspeicherung eingehen. Anschließend vergleichen wir die vorgestellten Protokolle zum Schutz der Historie und formulieren Ziele für die weitere Forschung. Den Abschluss bildet eine Diskussion über den Einfluss von dynamischen Effekten in kryptographischen Systemen und eine Einordnung der vorliegenden Arbeit in diesen Bereich.

8.1 Private Vorratsdatenspeicherung

In dieser Arbeit wurden die Anforderungen an die Sicherheit und die Anonymität von Vorratsdaten diskutiert und es wurde ein effizientes und sicheres Schema zur Vorratsdatenspeicherung unter Einsatz einer Schwellwertbedingung vorgestellt (Kapitel 4).

Für die weitere Forschung bietet es sich an, Protokolle für die private Vorratsdatenspeicherung zu entwickeln, die mit anderen Bedingungen zur Identifikation und Entschlüsselung als einem Schwellwert arbeiten. Es ist denkbar, dass andere Bedingungen mit dem Schwellwert-Schema kombiniert werden. Dazu könnte unter bestimmten Umständen eine Identifikation vorzeitig möglich gemacht oder die Vorhaltezeit von Daten unter einer vorgegebenen Bedingung verlängert werden. Letzteres kann zum Beispiel im Verkehrszentralregister geschehen, wenn neue Punkte eingetragen werden. Dabei muss sichergestellt werden, dass die Schlüssel kontrolliert und rechtmäßig der Sammelstelle zugänglich gemacht werden.

Eine andere Erweiterung des Schwellwert-Schemas betrifft den Wechsel des Providers. Im vorgestellten Schwellwert-Schema ist jeder Benutzer einem Provider zugeordnet. Möchte der Benutzer den Dienst eines anderen Providers in Anspruch nehmen, muss dieser Provider die Generierung der Vorratsdaten übernehmen. Dabei darf der Benutzer nicht in der Lage

sein zu betrügen. Bei einem Wechsel könnte sich der neue Provider beim alten Provider authentifizieren und die Daten des Benutzers erhalten. Das setzt voraus, dass die Provider immer direkt miteinander kommunizieren können. Alternativ kann der alte Provider die Daten des Benutzers unterschreiben und der Benutzer die unterschriebenen Daten dem neuen Provider übergeben. Hierbei müssen wir beachten, dass der Benutzer seine aktuellen Daten übergibt. Leitet der Benutzer Daten eines früheren Wechsels weiter, sind dem neuen Provider eventuell in der Zwischenzeit generierte Secret-Sharing-Instanzen unbekannt. Der neue Provider generiert dann neue Secret-Sharing-Instanzen. Folglich kann es für den Benutzer für einen Zeitabschnitt mehrere Secret-Sharing-Instanzen geben. Der Benutzer kann gegebenenfalls kritische Aktionen zwischen dem früheren und dem aktuellen Wechsel vertuschen, indem Shares von unterschiedlichen Secret-Sharing-Instanzen benutzt werden. Somit benötigen wir einen Mechanismus, der sicherstellt, dass die übergebenen Benutzerdaten aktuell sind. Dies lässt sich durch die Generierung eines eindeutigen Tokens realisieren. Beabsichtigt ein Benutzer seinen alten Provider zu verlassen, dann generiert der alte Provider ein eindeutiges Token und lässt dieses von der Sammelstelle kombiniert mit einem Fingerabdruck der aktuellen Daten des Benutzers blind unterschreiben. Der Benutzer überbringt seine Daten, das Token und die blinde Signatur dem neuen Provider. Dieser kann die Unterschrift überprüfen und validieren, ob die Daten des Benutzers und das Token zusammengehören. Anschließend übergibt der Provider das Token der Sammelstelle. Nur wenn die Sammelstelle das Token noch nicht empfangen hat, sind die Daten des Benutzers aktuell. Damit können wir sicherstellen, dass der neue Provider immer aktuelle Daten verwendet.

Eine weitere Herausforderung ist es, das Schema so zu erweitern, dass ein Benutzer mehrere Provider nutzen darf und dabei die Konsistenz der Vorratsdaten gewährleistet ist.

Der Provider stellt im Schwellwert-Schema sicher, dass der Benutzer nicht betrügt. Ein Ziel ist es, Protokolle zu finden, in denen eine Trusted Third Party nicht benötigt wird oder eine untergeordnete Rolle spielt. Für den Schutz der Historie können wir dies durch das Protokoll für zwei Nachrichtenlinien erreichen. Für die private Vorratsdatenspeicherung kann die Konsistenz der Shares untereinander durch den Einsatz eines Verifiable-Secret-Sharing-Schemes sichergestellt werden (siehe zum Beispiel [35]). Allerdings müssen die Geheimnisse (Identität, Schlüssel) der Secret-Sharing-Instanzen verifiziert werden. Ohne zusätzlichen Wissensgewinn über die Identität und die Schlüssel der Benutzer scheint es schwierig zu sein, dass die Sammelstelle sich in einem effizienten 2-Parteien-Protokoll mit einem bösartigen Benutzer von der Konsistenz eines Shares zu dem richtigen Geheimnis überzeugen lässt. Denkbar wäre, dass eine Trusted Third Party das Schema initialisiert. Anschließend kommunizieren die Benutzer und die Sammelstelle direkt miteinander und erstellen die Vorratsdaten gemeinsam, wobei Angriffe beider Parteien verhindert werden oder zumindest erkannt werden sollten. Weiterhin sollte untersucht werden, ob sichere Hardware und Krypto-Chips die Aufgaben des Providers ganz oder teilweise übernehmen können.

8.2 Schutz der Historie der Nachrichten

In Kapitel 5 haben wir festgestellt, dass aus der Kenntnis des zeitlichen Verlaufs der Datenspeicherung (Kenntnis der Historie) Wissen über Benutzer gewonnen werden kann. Wir haben mehrere Protokolle zum Schutz der Historie von kritischen Nachrichten untersucht:

- Abschnitt 5.3: Protokoll durch einfache Erweiterung der privaten Vorratsdatenspeicherung. Für jedes Share wird eine separate Nachricht an die Sammelstelle gesendet.

- Abschnitt 5.4: Protokoll mit Lockerung der Schwellwertbedingung. Eine Auswahl von Shares einer Nachricht wird an die Sammelstelle gesendet. Zwischen zwei Zeitabschnitten gibt es einen Versatz von θ Runden. Zu jedem Zeitpunkt gibt es λ überlappende Zeitabschnitte.
- Abschnitt 5.5: Protokoll zur Vermischung der Historien von Benutzern. IDs werden mehrfach verwendet. Jede Nachricht erhält eine zufällige von N IDs und einen Verweis auf die ID der vorhergehenden Nachricht des Benutzers.
- Kapitel 6: Protokoll unter Verwendung von zwei Nachrichtenlinien. In jeder Runde wird zufällig eine von zwei erstellten Nachrichten zur Speicherung ausgewählt. Eine Nachricht enthält Verweise auf die $k \in \Omega(\log(d))$ vorhergehenden Nachrichten der jeweils anderen Linie. Zur Erkennung von bösartigen Benutzern ohne Provider wird das Protokoll unter Verwendung von blinden Signaturen erweitert.
- Kapitel 7: Protokoll unter Verwendung von ℓ Nachrichtenlinien. In jeder Runde wird zufällig eine von ℓ erstellten Nachrichten zur Speicherung ausgewählt. Eine Nachricht enthält Verweise auf die $k \in \Omega(\ell^2 + \ell(c + \ln(d)))$ vorhergehenden Nachrichten der jeweils anderen Linien.

In Tabelle 8.1 sind die weiteren Eigenschaften dieser Protokolle zusammengefasst.

Ein wünschenswertes Ziel für die weitere Forschung wäre es, effiziente und sichere Protokolle zum Schutz der Historie zu finden, so dass

- alle Nachrichten eines Benutzers, die innerhalb von Δ Runden gespeichert werden, einander zugeordnet werden können.
- Nachrichten, die vor mehr als Δ Runden gespeichert wurden, nicht einer aktuellen Nachricht eines Benutzers zugeordnet werden können.

Wir vermuten, dass es kein Protokoll gibt, das diese Grenzen genau einhalten kann. Es scheint, dass immer ein gewisser Trade-Off zwischen Zuordnung der Nachrichten und Entkopplung der Historie vorhanden ist. Ein zentrales Ziel liegt in der Begrenzung dieses Trade-Offs.

8.3 Bezug der Arbeit zu dynamischen Effekten in der Kryptographie

Bei der Untersuchung von dynamischen Effekten in der Kryptographie wollen wir verstehen, wie der zeitliche Verlauf ein kryptographisches System beeinflusst. Bei der privaten Vorratsdatenspeicherung können wir sehen, dass es gelingen kann, durch ein Protokoll die gewünschten Daten zu schützen und zu anonymisieren. Jedoch könnte bei der Durchführung eines solchen Protokolls anderes Wissen über eine Partei gewonnen werden, zum Beispiel die Historie der Nachrichten. Zum Schutz der Privatheit sollte solches Wissen ebenfalls geschützt werden. Beim Verfahren von Jarecki und Shmatikov [62] zur Freigabe von Schlüsseln wird hingegen gefordert, dass der gesamte zeitliche Nachrichtenverlauf bei einer Freigabe eines Schlüssels preisgegeben wird. Es hängt folglich vom Szenario ab, ob eine bestimmte Information über eine Partei geschützt werden muss. Selten wird bei der Analyse eines sicheren Protokolls untersucht, welches weitere Wissen durch die Ausführung über teilnehmende Parteien gewonnen werden kann. Es sind viele Protokolle bekannt, die Peer-to-Peer-Kommunikation oder Broadcast-Kommunikation in Ad-Hoc-Netzwerken realisieren.

Abschnitt	Schwellwert	Entkopplung	Vorteile (+) / Nachteile (–)
5.3	d	falls keine kritische Aktion in Δ Runden	(+) einfache Erweiterung des Schwellwert-Schemas (–) bei kleiner Anzahl an Benutzern Gefahr der teilweisen Rekonstruktion der Historien
5.4	$\Theta(\lambda \cdot d)$	falls keine kritische Aktion in Δ Runden	(+) einfache Erweiterung des Schwellwert-Schemas (+) geringer Kommunikationsaufwand (–) bei kleiner Anzahl an Benutzern Gefahr der teilweisen Rekonstruktion von Historien von Benutzern, die häufig kritische Aktionen durchführen (–) keine kontinuierliche Verschiebung des Identifikationsfensters, sondern Versatz um θ Runden
5.5	d mit großer Sicherheit	nach $\mathcal{O}(N^{2-\tilde{\varepsilon}} \ln(N))$ Runden mit großer Sicherheit	(+) bei beliebiger Anzahl von Benutzern einsetzbar (+) unabhängig von der Vorratsdatenspeicherung (+) robuster als die anderen Ansätze bei Zusatzinformation über typisches Benutzerverhalten durch Vermischung der Historien verschiedener Benutzer (+) Trade-Off zwischen Laufzeit und Anonymisierung durch Parameter N steuerbar (–) nur für kleine Werte d praktikabel
6	d mit großer Sicherheit	nach $\mathcal{O}(\frac{d^2}{\varepsilon^2} \log(\frac{d}{\varepsilon}))$ Nachrichten mit großer Sicherheit	(+) bei beliebiger Anzahl von Benutzern einsetzbar (+) unabhängig von der Vorratsdatenspeicherung (+) ohne Provider einsetzbar (+) robust gegen bösartige Benutzer und Sammelstelle (–) Entkopplung abhängig von Anzahl der Nachrichten und unabhängig von der Zeit
7	d mit großer Sicherheit	nach $\mathcal{O}(kc(3\ell)^k)$ Nachrichten mit großer Sicherheit	(+) bei beliebiger Anzahl von Benutzern einsetzbar (+) unabhängig von der Vorratsdatenspeicherung (+) geringerer Bedarf an Zufallszahlen und kleinere Nachrichten als für zwei Nachrichtenlinien (+/-) Verlust von Nachrichten aus der Historie, ohne dass Entkopplung stattfindet (–) Entkopplung abhängig von Anzahl der Nachrichten und unabhängig von der Zeit (–) ohne Provider nur robust gegen bösartige Benutzer, falls Einsatz anderer Trusted Third Parties

Tabelle 8.1: Zusammenfassung der Eigenschaften der Protokolle zum Schutz der Historie

Welches Wissen die an der Kommunikation teilnehmenden Partei über das Netzwerk erhalten, wird bei der Vorstellung der Protokolle nicht untersucht. Dabei kann die Netzwerktopologie selbst ein Geheimnis darstellen. In [56] wird gezeigt, dass bei der Einbindung eines neuen Knotens in ein Netzwerk mehr Wissen über die Netzwerktopologie gewonnen werden kann als zur Durchführung der Kommunikation nötig ist.

In den hier vorgestellten Protokollen zur Vorratsdatenspeicherung können die gespeicherten Daten erst verzögert nach dem Eintritt einer bestimmten Bedingung geöffnet werden. Innerhalb der Vorhaltezeit muss dazu ein Schwellwert überschritten werden. Andere Ar-

beiten beschäftigen sich mit der verzögerten Öffnung von Geheimnissen unabhängig von einer solchen Bedingung. Rivest, Shamir und Wagner [96] schlagen vor, die Lösung eines zeitaufwendigen Problems als Schlüssel zur Öffnung eines Geheimnisses zu verwenden. Dieser Ansatz wird mehrfach aufgegriffen, unter anderem von Boneh und Naor [13] zur Umsetzung von sogenannten Timed Commitments. Damit lassen sich die Commitments von Alice nach einer bestimmten Zeit garantiert öffnen und verifizieren, auch wenn sich Alice vor der Öffnungsphase aus dem Protokoll zurückzieht.

Weitere dynamische Aspekte in der Kryptographie entstehen dann, wenn Daten von Parteien nach Beginn eines Protokolls verändert werden sollen. Durch den Einsatz von Proactive Secret Sharing [54, 17] lässt sich ein verteilter Schlüssel über die Zeit aktualisieren. Somit werden Informationen über den Schlüssel nutzlos, die ein Angreifer vor der Aktualisierung erhalten hat. Im Allgemeinen muss die Ausführung eines privaten Protokolls erneut gestartet werden, wenn eine teilnehmende Partei ihre Eingabe ändern will. Insbesondere bei privaten Auktionen ist es jedoch möglich, dass nur das sich ändernde Gebot neu abgegeben werden muss [59, 55]. Die anderen Bieter müssen nicht erneut aktiv werden. Dieses ist auch der Fall, wenn eine Auktion um neue Bieter erweitert werden muss.

Insgesamt wurden die wichtigsten Einsatzgebiete und Grundlagen der Kryptographie wie Verschlüsselung, Signatur oder die Sicherheit von Protokollen in der Literatur ausgiebig untersucht. Die Untersuchung der entsprechenden Protokolle beschränkt sich zumeist auf statische Anwendungen. Diese Arbeit legt mit den verschiedenen Verfahren zur privaten Vorratsdatenspeicherung und zum Schutz der Historie von Nachrichten besonderen Wert auf Daten im zeitlichen Verlauf und damit auf dynamische Aspekte. In der Zukunft erwarten wir ein stärkeres Interesse der Forschung an der Identifikation von weiterem zu schützenden Wissen wie der Historie und an der Untersuchung von dynamischen Aspekten in der Kryptographie.

Literaturverzeichnis

[1] Gesetz zur Neuregelung der Telekommunikationsüberwachung und anderer verdeckter Ermittlungsmaßnahmen sowie zur Umsetzung der Richtlinie 2006/24/EG. In: *Bundesgesetzblatt* Teil I (2007), Dezember, Nr. 70, S. 3198–3211

[2] AHO, A. V. ; HOPCROFT, J. E. ; ULLMAN, J. D.: *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974

[3] BARAK, B.: *Non-Black-Box Techniques in Cryptography*, Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel, Dissertation, 2004

[4] BEAVER, D.: Foundations of Secure Interactive Computing. In: *Proceedings of the 11th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '91)*, Springer, 1991, S. 377–391

[5] BEAVER, D.: Secure Multiparty Protocols and Zero-Knowledge Proof Systems Tolerating a Faulty Minority. In: *Journal of Cryptology* 4 (1991), Nr. 2, S. 75–122

[6] BELLARE, M. ; NAMPREMPRE, C. ; POINTCHEVAL, D. ; SEMANKO, M.: The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. In: *Journal of Cryptology* 16 (2003), Nr. 3, S. 185–215

[7] BLAKLEY, G.: Safeguarding Cryptographic Keys. In: *Proceedings of the National Computer Conference*, AFIPS Press, 1979, S. 313–317

[8] BLANCHETTE, J.-F. ; JOHNSON, D. G.: Cryptography, Data Retention, and the Panopticon Society (Abstract). In: *Proceedings of the ethics and social impact component on Shaping policy in the information age (ACM Policy)*, ACM, 1998, S. 1–2

[9] BLANCHETTE, J.-F. ; JOHNSON, D. G.: Data Retention and the Panoptic Society: The Social Benefits of Forgetfulness. In: *The Information Society* 18 (2002), S. 33–45

[10] BLUM, L. ; BLUM, M. ; SHUB, M.: Comparison of Two Pseudo-Random Number Generators. In: CHAUM, D. (Hrsg.) ; RIVEST, R. L. (Hrsg.) ; SHERMAN, A. T. (Hrsg.): *Advances in Cryptology: Proceedings of CRYPTO '82*, Plenum, 1982, S. 61–78

[11] BLUM, M. ; MICALI, S.: How to Generate Cryptographically Strong Sequences of Pseudo Random Bits. In: *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (FOCS '82)*, IEEE, 1982, S. 112–117

[12] BLUNDO, C. ; DE SANTIS, A. ; PERSIANO, G. ; VACCARO, U.: Randomness Complexity of Private Computation. In: *Computational Complexity* 8 (1999), Nr. 2, S. 145–168

[13] BONEH, D. ; NAOR, M.: Timed Commitments. In: *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO 2000)*, Springer, 2000, S. 236–254

[14] BRASSARD, G. ; CHAUM, D. ; CRÉPEAU, C.: Minimum Disclosure Proofs of Knowledge. In: *Journal of Computer and System* 37 (1988), Nr. 2, S. 156–189

[15] BRASSARD, G. ; CRÉPEAU, C. ; ROBERT, J.-M.: All-or-Nothing Disclosure of Secrets. In: ODLYZKO, A. M. (Hrsg.): *Proceedings of the 6th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '86)*, Springer, 1986, S. 234–238

[16] BUNDESVERFASSUNGSGERICHT: *1 BvR 256/08 vom 2.3.2010, Absatz-Nr. (1–345)*

[17] CACHIN, C. ; KURSAWE, K. ; LYSYANSKAYA, A. ; STROBL, R.: Asynchronous Verifiable Secret Sharing and Proactive Cryptosystems. In: *Proceedings of the 9th ACM Conference on Computer and Communications Security*, ACM, 2002, S. 88–97

[18] CANETTI, R.: Security and Composition of Multi-party Cryptographic Protocols. In: *Journal of Cryptology* 13 (2000), Nr. 1, S. 143–202

[19] CHAUM, D.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. In: *Communications of the ACM* 24 (1981), Nr. 2, S. 84–88

[20] CHAUM, D.: Blind Signatures for Untraceable Payments. In: CHAUM, D. (Hrsg.) ; RIVEST, R. L. (Hrsg.) ; SHERMAN, A. T. (Hrsg.): *Advances in Cryptology: Proceedings of CRYPTO '82*, Plenum, 1982, S. 199–203

[21] CHAUM, D.: Security Without Identification: Transaction Systems to Make Big Brother Obsolete. In: *Communications of the ACM* 28 (1985), Nr. 10, S. 1030–1044

[22] CHAUM, D. ; DAMGÅRD, I. ; VAN DE GRAAF, J.: Multiparty Computations Ensuring Privacy of Each Party's Input and Correctness of the Result. In: POMERANCE, C. (Hrsg.): *Proceedings of the 7th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '87)*, Springer, 1987, S. 87–119

[23] CHAUM, D. ; FIAT, A. ; NAOR, M.: Untraceable Electronic Cash. In: GOLDWASSER, S. (Hrsg.): *Proceedings of the 8th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '88)*, Springer, 1988, S. 319–327

[24] CHERNOFF, H.: A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations. In: *The Annals of Mathematical Statistics* (1952), S. 493–507

[25] CHOR, B. ; GOLDWASSER, S. ; MICALI, S. ; AWERBUCH, B.: Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (Extended Abstract). In: *Proceedings of the 26th Annual Symposium on Foundations of Computer Science (FOCS '85)*, IEEE, 1985, S. 383–395

[26] CORMEN, T. H. ; LEISERSON, C. E. ; RIVEST, R. L. ; STEIN, C.: *Introduction to Algorithms*. 2nd Revised edition. The MIT Press, 2001

[27] CRÉPEAU, C.: Equivalence Between Two Flavours of Oblivious Transfers. In: POMERANCE, C. (Hrsg.): *Proceedings of the 7th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '87)*, Springer, 1987, S. 350–354

[28] DAEMEN, J. ; RIJMEN, V.: *The Design of Rijndael*. Springer, 2002

[29] DAMGÅRD, I.: Collision Free Hash Functions and Public Key Signature Schemes. In: CHAUM, D. (Hrsg.) ; PRICE, W. L. (Hrsg.): *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT '87)*, Springer, 1987, S. 203–216

[30] DAVIES, D. W. ; PRICE, W. L.: The Application of Digital Signatures Based on Public Key Cryptosystems. In: *Proceedings of the Fifth International Computer Communications Conference*, 1980, S. 525–530

[31] DIFFIE, W. ; HELLMAN, M.: New Directions in Cryptography. In: *IEEE Transactions on Information Theory* 22 (1976), Nr. 6, S. 644–654

[32] DÜMBGEN, L.: *Stochastik für Informatiker*. Springer, 2003

[33] EUROPÄISCHES PARLAMENT UND RAT: *Richtlinie 2006/24/EG des Europäischen Parlaments und des Rates vom 15. März 2006 über die Vorratsspeicherung von Daten, die bei der Bereitstellung öffentlich zugänglicher elektronischer Kommunikationsdienste oder öffentlicher Kommunikationsnetze erzeugt oder verarbeitet werden, und zur Änderung der Richtlinie 2002/58/EG*. 2006

[34] EVEN, S. ; GOLDREICH, O. ; LEMPEL, A.: A Randomized Protocol for Signing Contracts. In: *Communications of the ACM* 28 (1985), Nr. 6, S. 637–647

[35] FELDMAN, P.: A Practical Scheme for Non-interactive Verifiable Secret Sharing. In: *Proceedings of the 28th Annual Symposium on Foundations of Computer Science (FOCS '87)*, IEEE, 1987, S. 427–437

[36] FERGUSON, N. ; SCHNEIER, B.: *Practical Cryptography*. John Wiley & Sons, Inc., 2003

[37] FISCHLIN, M. ; SCHRÖDER, D.: Security of Blind Signatures under Aborts. In: JARECKI, S. (Hrsg.) ; TSUDIK, G. (Hrsg.): *Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography (PKC 2009)*, Springer, 2009, S. 297–316

[38] GENNARO, R.: *Theory and Practice of Verifiable Secret Sharing*, Massachusetts Institute of Technology, Dissertation, 1996

[39] GOLDREICH, O.: *Modern Cryptography, Probabilistic Proofs, and Pseudorandomness*. Springer, 1999

[40] GOLDREICH, O. ; GOLDWASSER, S. ; MICALI, S.: How to Construct Random Functions. In: *Journal of the ACM* 33 (1986), Nr. 4, S. 792–807

[41] GOLDREICH, O.: *Foundations of Cryptography: Volume I Basic Tools*. Cambridge University Press, 2001

[42] GOLDREICH, O.: *Foundations of Cryptography: Volume II Basic Applications*. Cambridge University Press, 2004

[43] GOLDREICH, O.: *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008

[44] GOLDREICH, O.: *Pseudorandom Generators: A Primer*. `http://www.wisdom.weizmann.ac.il/~oded/prg-primer.html`. Version: 2008

[45] GOLDREICH, O. ; GOLDWASSER, S. ; MICALI, S.: How to Construct Random Functions. In: *Journal of the ACM* 33 (1986), Nr. 4, S. 792–807

[46] GOLDREICH, O. ; MICALI, S. ; WIGDERSON, A.: How to Prove all NP-Statements in Zero-Knowledge, and a Methodology of Cryptographic Protocol Design. In: ODLYZKO, A. M. (Hrsg.): *Proceedings of the 6th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '86)*, Springer, 1986, S. 171–185

[47] GOLDREICH, O. ; MICALI, S. ; WIGDERSON, A.: How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In: *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC '87)*, ACM, 1987, S. 218–229

[48] GOLDWASSER, S. ; MICALI, S.: Probabilistic Encryption. In: *Journal of Computer and System Science* 28 (1984), Nr. 2, S. 270–299

[49] GOLDWASSER, S. ; MICALI, S. ; YAO, A.: Strong Signature Schemes. In: *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC '83)*, ACM, 1983, S. 431–439

[50] GOLDWASSER, S. ; MICALI, S. ; RIVEST, R. L.: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. In: *SIAM Journal on Computing* 17 (1988), Nr. 2, S. 281–308

[51] GRINSTEAD, C. ; SNELL, J.: *Introduction to Probability*. American Mathematical Society, 1997

[52] HAGERUP, T. ; RÜB, C.: A Guided Tour of Chernoff Bounds. In: *Information Processing Letters* 33 (1990), Nr. 6, S. 305–308

[53] HÅSTAD, J. ; IMPAGLIAZZO, R. ; LEVIN, L. A. ; LUBY, M.: A Pseudorandom Generator from any One-way Function. In: *SIAM Journal on Computing* 28 (1999), Nr. 4, S. 1364–1396

[54] HERZBERG, A. ; JARECKI, S. ; KRAWCZYK, H. ; YUNG, M.: Proactive secret sharing or: How to cope with perpetual leakage. In: COPPERSMITH, D. (Hrsg.): *Proceedings of the 15th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '95)*, Springer, 1995, S. 339–352

[55] HINKELMANN, M. ; JAKOBY, A. ; MOEBIUS, N. ; ROMPF, T. ; STECHERT, P.: A Cryptographically t-Private Auction System. In: *Proceedings of the Third International Conference on Network and System Security (NSS 2009)*, IEEE, 2009, S. 44–51

[56] HINKELMANN, M. ; JAKOBY, A.: Communications in unknown networks: Preserving the secret of topology. In: *Theoretical Computer Science* 384 (2007), Nr. 2-3, S. 184–200

[57] HINKELMANN, M. ; JAKOBY, A.: Preserving Privacy versus Data Retention. Schriftenreihe der Institute für Informatik und Mathematik, Universität zu Lübeck, 2008 (SIIM-TR-A-08-04). – Forschungsbericht

[58] HINKELMANN, M. ; JAKOBY, A.: Preserving Privacy versus Data Retention. In: CHEN, J. (Hrsg.) ; COOPER, S. B. (Hrsg.): *Proceedings of the 6th Annual Conference on Theory and Applications of Models of Computation (TAMC 2009)*, Springer, 2009, S. 251–260

[59] HINKELMANN, M. ; JAKOBY, A. ; STECHERT, P.: *t*-Private and *t*-Secure Auctions. In: *Journal of Computer Science and Technology* 23 (2008), Nr. 5, S. 694–710

[60] IMPAGLIAZZO, R. ; RUDICH, S.: Limits on the Provable Consequences of One-way Permutations. In: GOLDWASSER, S. (Hrsg.): *Proceedings of the 8th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '88)*, Springer, 1988, S. 8–26

[61] ISHAI, Y. ; KUSHILEVITZ, E. ; LINDELL, Y. ; PETRANK, E.: Black-Box Constructions for Secure Computation. In: KLEINBERG, J. M. (Hrsg.): *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC 2006)*, ACM, 2006, S. 99–108

[62] JARECKI, S. ; SHMATIKOV, V.: Handcuffing Big Brother: an Abuse-Resilient Transaction Escrow Scheme. In: CACHIN, C. (Hrsg.) ; CAMENISCH, J. (Hrsg.): *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT 2004)* , Springer, 2004, S. 590–608

[63] JOUX, A.: *Algorithmic Cryptanalysis*. Chapman & Hall / CRC Press, 2009

[64] JUELS, A. ; LUBY, M. ; OSTROVSKY, R.: Security of Blind Digital Signatures (Extended Abstract). In: KALISKI, B. S. J. (Hrsg.): *Proceedings of the 17th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '97)*, Springer, 1997, S. 150–164

[65] KALFUS, O. ; RONEN, B. ; SPIEGLER, I.: A selective data retention approach in massive databases. In: *Omega* 32 (2004), Nr. 2, S. 87–95

[66] KILIAN, J.: Founding Cryptography on Oblivious Transfer. In: *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC '88)*, ACM, 1988, S. 20–31

[67] KOBLITZ, N.: Elliptic Curve Cryptosystems. In: *Mathematics of Computation* 48 (1987), Nr. 177, S. 203–209

[68] KOLESNIKOV, V.: Truly Efficient String Oblivious Transfer Using Resettable Tamper-Proof Tokens. In: MICCIANCIO, D. (Hrsg.): *Proceedings of the 7th Theory of Cryptography Conference, Theory of Cryptography (TCC 2010)*, Springer, 2010, S. 327–342

[69] LANDAU, S.: Security, Liberty, and Electronic Communications. In: FRANKLIN, M. K. (Hrsg.): *Proceedings of the 24th Annual Cryptology Conference, Advances in Cryptology (CRYPTO 2004)*, Springer, 2004, S. 355–372

[70] MAHLMANN, P. ; SCHINDELHAUER, C.: *Peer-to-Peer-Netzwerke: Algorithmen und Methoden*. Springer, 2007

[71] MARX, G. T.: Ethics for the New Surveillance. In: *The Information Society* 14 (1998), Nr. 3, S. 171–185

[72] MAURER, U.: Information-Theoretic Cryptography. In: WIENER, M. J. (Hrsg.): *Proceedings of the 19th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '99)*, Springer, 1999, S. 47–64

[73] MENEZES, A. ; VAN OORSCHOT, P. C. ; VANSTONE, S. A.: *Handbook of Applied Cryptography*. CRC Press, 1996

[74] MERKLE, R. ; HELLMAN, M.: Hiding Information and Signatures in Trapdoor Knapsacks. In: *IEEE Transactions on Information Theory* 24 (1978), Nr. 5, S. 525–530

[75] MERKLE, R. C.: A Certified Digital Signature. In: BRASSARD, G. (Hrsg.): *Proceedings of the 9th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '89)*, Springer, 1989, S. 218–238

[76] MERKLE, R.: A Fast Software One-way Hash Function. In: *Journal of Cryptology* 3 (1990), Nr. 1, S. 43–58

[77] MICALI, S. ; ROGAWAY, P.: Secure Computation (Abstract). In: FEIGENBAUM, J. (Hrsg.): *Proceedings of the 11th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '91)*, Springer, 1991, S. 392–404

[78] MILLER, V.: Use of Elliptic Curves in Cryptography. In: *Proceedings of the 5th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '85)* Springer, 1986, S. 417–426

[79] MITZENMACHER, M. ; UPFAL, E.: *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005

[80] MOTWANI, R. ; RAGHAVAN, P.: *Randomized Algorithms*. Cambridge University Press, 1995

[81] NAOR, M. ; YUNG, M.: Universal One-Way Hash Functions and their Cryptographic Applications. In: *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC '89)*, ACM, 1989, S. 33–43

[82] NAOR, M.: Bit Commitment Using Pseudorandomness. In: *Journal of Cryptology* 4 (1991), Nr. 2, S. 151–158

[83] NATIONAL BUREAU OF STANDARDS: *Data Encryption Standard*. FIPS-Pub. 46, 1977

[84] NATIONAL BUREAU OF STANDARDS: *Advanced Encryption Standard*. FIPS-Pub. 197, 2001

[85] NG, K. ; LIU, H.: Customer Retention via Data Mining. In: *Artificial Intelligence Review* 14 (2000), Nr. 6, S. 569–590

[86] PEDERSEN, T. P.: Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: FEIGENBAUM, J. (Hrsg.): *Proceedings of the 11th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '91)*, Springer, 1991, S. 129–140

[87] PFITZMANN, A. ; KÖPSELL, S.: Risiken der Vorratsspeicherung. In: *Datenschutz und Datensicherheit-DuD* 33 (2009), Nr. 9, S. 542–546

[88] PIRONIO, S. ; ACÍN, A. ; MASSAR, S. ; DE LA GIRODAY, A. ; MATSUKEVICH, D. ; MAUNZ, P. ; OLMSCHENK, S. ; HAYES, D. ; LUO, L. ; MANNING, T. u. a.: Random numbers certified by Bell's theorem. In: *Nature* 464 (2010), Nr. 7291, S. 1021–1024

[89] POINTCHEVAL, D. ; STERN, J.: Security Arguments for Digital Signatures and Blind Signatures. In: *Journal of Cryptology* 13 (2000), Nr. 3, S. 361–396

[90] PRENEEL, B.: *Analysis and Design of Cryptographic Hash Functions*, KU Leuven, Dissertation, 1993

[91] RABIN, M. O.: Digitalized Signatures and Public Key Functions as Intractable as Factorization. Massachusetts Institute of Technology, 1979 (TR-212). – Forschungsbericht

[92] RABIN, M. O.: How to Exchange Secrets by Oblivious Transfer. Harvard Aiken Computation Laboratory, 1981 (TR-81). – Forschungsbericht

[93] REINGOLD, O. ; TREVISAN, L. ; VADHAN, S. P.: Notions of Reducibility between Cryptographic Primitives. In: NAOR, M. (Hrsg.): *Proceedings of the First Theory of Cryptography Conference, Theory of Cryptography (TCC 2004)*, Springer, 2004, S. 1–20

[94] REISCHUK, R. ; HINKELMANN, M.: Einweg-Funktionen: Vorsicht Falle - Rückweg nur für Eingeweihte! In: *Taschenbuch der Algorithmen*. Springer, 2008, S. 139–148

[95] RIVEST, R. L. ; SHAMIR, A. ; ADLEMAN, L.: A Method for Obtaining Digital Signatures and Public Key Cryptosystems. In: *Communications of the ACM* 21 (1978), Nr. 2, S. 120–126

[96] RIVEST, R. L. ; SHAMIR, A. ; WAGNER, D. A.: Time-lock Puzzles and Timed-release Crypto. Massachusetts Institute of Technology, 1996. – Forschungsbericht

[97] ROSS, S.: *A first course in probability*. Macmillan Publishing Company, 1994

[98] ROSSNAGEL, A. ; BEDNER, M. ; KNOPP, M.: Rechtliche Anforderungen an die Aufbewahrung von Vorratsdaten. In: *Datenschutz und Datensicherheit-DuD* 33 (2009), Nr. 9, S. 536–541

[99] SCHNEIER, B.: *Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition*. John Wiley & Sons, Inc., 1996

[100] SCHOENMAKERS, B.: A Simple Publicly Verifiable Secret Sharing Scheme and Its Application to Electronic Voting. In: WIENER, M. J. (Hrsg.): *Proceedings of the 19th Annual Cryptology Conference, Advances in Cryptology (CRYPTO '99)*, Springer, 1999, S. 148–164

[101] SHAMIR, A.: How to Share a Secret. In: *Communications of the ACM* 22 (1979), Nr. 11, S. 612–613

[102] SHANNON, C. E.: A Mathematical Theory of Communication. In: *Bell System Technical Journal* 27 (1948), Nr. 3/4, S. 379–423, 623–656

[103] SHANNON, C. E.: Communication Theory of Secrecy Systems. In: *Bell System Technical Journal* 28 (1949), S. 656–715

[104] STADLER, M.: Publicly Verifiable Secret Sharing. In: MAURER, U. M. (Hrsg.): *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT '96)*, Springer, 1996, S. 190–199

[105] STOCKHUSEN, C.: *Implementierung und Evaluierung von Protokollen zur privaten Vorratsdatenspeicherung*. Studienarbeit, Institut für Theoretische Informatik, Universität zu Lübeck, 2009

[106] TARJAN, R. E.: Efficiency of a Good But Not Linear Set Union Algorithm. In: *Journal of the ACM* 22 (1975), Nr. 2, S. 215–225

[107] UNABHÄNGIGES LANDESZENTRUM FÜR DATENSCHUTZ SCHLESWIG-HOLSTEIN: *Tätigkeitsbericht 2010*. Landtagsdrucksache 17/210 des Landtags Schleswig-Holstein, 2010

[108] VAN WANROOIJ, W. ; PRAS, A.: Data on Retention. In: SCHÖNWÄLDER, J. (Hrsg.) ; SERRAT, J. (Hrsg.): *Proceedings of the 16th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2005)*, Springer, 2005, S. 60–71

[109] WERNER, A.: *Elliptische Kurven in der Kryptographie*. Springer, 2002

[110] WORSCH, T.: *Skript zur Vorlesung Randomisierte Algorithmen*. IAKS Vollmar, Universität Karlsruhe, 2010

[111] YAO, A. C.-C.: Theory and Applications of Trapdoor Functions. In: *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (FOCS '82)*, IEEE, 1982, S. 80–91

[112] ZHANG, Z. ; LIAN, Q. ; LIN, S. ; CHEN, W. ; CHEN, Y. ; JIN, C.: BitVault: a Highly Reliable Distributed Data Retention Platform. In: *SIGOPS Operating Systems Review* 41 (2007), Nr. 2, S. 27–36

Symbolverzeichnis

$\circ$, Konkatenation von Strings, Verkettung von Funktionen, 9
$\oplus$, XOR-Operation, 9
$\overset{c}{\equiv}$, nicht effizient unterscheidbar, 19
$\overset{s}{\equiv}$, statistisch ununterscheidbar, 19
$\equiv$, ununterscheidbar, 19

$\mathcal{AM}_t$, Menge aller Nachrichten, die die Sammelstelle bis zur Runde t gespeichert hat, 62

$\mathcal{BPP}$, Komplexitätsklasse Bounded-Probability Polynomial Time, 16
$\mathcal{B}_\tau$, Menge der Benutzer, die in jedem der Zeitabschnitte $\mathcal{T}_0, \ldots, \mathcal{T}_\tau$ weniger als d kritische Aktionen ausgeführt haben, 60
$\widetilde{\mathcal{B}}_t$, Menge der Benutzer, die bis zur Runde t in keinem Zeitabschnitt d kritische Aktionen durchgeführt haben, 88

$C_{[t]}(M)$, Menge aller Kandidaten für die Teilhistorie der Nachricht M, 85

d, Schwellwert, 49
Δ, Länge der Vorhaltezeit/Zeitabschnitte, 49

$\mathbb{E}$, Erwartungswert, 11
$\mathrm{enc}_K(x)$, Verschlüsselung von x mit dem Schlüssel K, 55
ε, kleine Fehlerwahrscheinlichkeit, 88
E', Zufallsvariable, die die Anzahl der Nachrichten bis zur Entkopplung der Historie beschreibt, 109

$\mathrm{fp}_{R,L}(\mathcal{I}_i)$, verschlüsselter Fingerabdruck der Identität $\mathcal{I}_i$ bezüglich Indikatorstring R und Schlüssel L, 55

$\widehat{G}$, Pseudozufallszahlengenerator für $\mathrm{seed}^{(\cdot)}(\cdot)$ und $\mathrm{rand}^{(\cdot)}(\cdot)$, 51
$\mathcal{G}^\ell$, Verbindungsgraph für ℓ Nachrichtenlinien, 131
$G_{[t]}$, Nachrichtengraph, 85
$\widetilde{G}_{[t]}$, von $\widetilde{\mathcal{M}}_{[t]}$ induzierter Nachrichtengraph, 88
$G[v]$, Zusammenhangskomponente des Knotens v im Graphen G, 127

$H(X)$, Entropie von X, 24

$\mathrm{ID}_j(M)$, j-te ID der Nachricht M, 102
$\mathrm{ID}(M)$, ID der Nachricht M, 49
$\mathrm{IDpre}_j(M)$, j-te Vorgänger-ID der Nachricht M, 102
$\mathrm{IDpre}(M)$, Vorgänger-ID der Nachricht M, 85

$\mathcal{I}_i$, Identität des i-ten Benutzers, 55
$\mathcal{I}(M)$, Identität des Benutzers der Nachricht M, 60
$I(X;Y)$, Mutual Information von X und Y, 24

k, Anzahl der Nachrichten, auf die eine Nachricht in den Linienprotokollen verweist, 98
K, Schlüssel zur Verschlüsselung der Nutzlast, 55
$\text{keys}^{[\tau]}(S)$, Folge der Schlüssel K und L der Zeitabschnitte $\mathcal{T}_0, \ldots \mathcal{T}_\tau$ bezüglich des Seeds S, 59
KW, Zufallsvariable, die beschreibt, in welchem Block zum ersten Mal ein Wechsel auftritt, 110

ℓ, Anzahl der Nachrichtenlinien, 131
ℓ, Seedlänge von $\widehat{G}$, 51
L, Schlüssel für den Fingerabdruck, 55
λ, Anzahl der überlappenden Zeitintervalle mit Versatz θ, 77
$\text{line}(M)$, Linie der Nachricht M, 102
$\text{load}(M)$, Nutzlast der Nachricht M, 49

m, Sicherheitsparameter des Fingerabdrucks, 56
M, Nachricht, 49
$\mathbb{M}$, momentgenerierende Funktion, 11
$M_{i,t}$, t-te Nachricht auf Linie i des Verbindungsgraphen, 98
$M\text{pre}(M)$, kritische Vorgängernachricht der Nachricht M, 85
M''^{t}, t-te Nachricht eines Benutzers, empfangen von der Sammelstelle, 98
$\widetilde{\mathcal{M}}_{[t]}$, Menge der kritischen Nachrichten M von Benutzern aus $\widetilde{\mathcal{B}}_t$ mit $\text{time}(M) \leq t$, 88
$\mathcal{M}_\tau$, Menge aller Nachrichten der Benutzer aus $\mathcal{B}_\tau$, 60

n, Anzahl aller Benutzer, 55
N, Anzahl der IDs (Vermischung der Historien), 85
$\mathbb{N}$, natürliche Zahlen, 10
$\mathcal{NP}$, Komplexitätsklasse Non-Deterministic Polynomial Time, 16

OT_b, 1-aus-n-Oblivious-Transfer bezüglich des Auswahlbits b, 38
$\text{OUTPUT}_i^{\mathcal{P}}$, Ausgabe der Partei i bei der Ausführung des Protokolls $\mathcal{P}$, 23

$\mathcal{P}$, Komplexitätsklasse Polynomial Time, 16
Π_t, Menge aller Zeitabschnitte $\mathcal{T}_\tau$ mit $t \in \mathcal{T}_\tau$, 49
$\mathcal{P}/\text{poly}$, Komplexitätsklasse, 17
Pr, Wahrscheinlichkeit, 10
p_τ, Polynom der Secret-Sharing-Instanz des Zeitabschnitts $\mathcal{T}_\tau$, 58

R, Indikatorstring des Identifikationsmechanismus, 55
$\mathbb{R}$, reelle Zahlen, 13

S, Seed eines Benutzers, 49
$\mathcal{S}$, Sammlungsgraph, 98
$\text{rand}^{(\cdot)}(\cdot)$, pseudozufällige Stringfunktion, 51
$\text{seed}^{(\cdot)}(\cdot)$, Seedgenerierungsfunktion, 51
$\text{share}(M)$, Shares der Nachricht M, 49
$\text{sig}M_{i,t}$, Signatur der Nachricht $M_{i,t}$, 123

Index