Karlsruhe Institute of Technology

Alternative Interpolationsverfahren im Naor-Pinkas-Broadcast-Schema

Studienarbeit

Sébastien Jelsch

Institut für Kryptographie und Sicherheit
Europäisches Institut für Systemsicherheit
Fakultät für Informatik
Karlsruher Institut für Technologie

Betreuender Professor:	Jun.-Prof. Dr. D. Hofheinz
Betreuender Mitarbeiter:	Dipl.-Inf. C. Striecks

Bearbeitungszeit: 15. November 2012 – 13. Mai 2013

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, den 13. Mai 2013

Inhaltsverzeichnis

1. Einleitung

1.1 Motivation

In der heutigen Welt sind Multicast-Übertragungen durch die immer weiter steigende Nachfrage nach Pay-TV, Multimedia-Anwendungen und dem zur Verfügungstellen von geschützten Inhalten zu einem sehr wichtigen und zentralen Thema geworden; denn digitale Medien sind einfach zu kopieren und zu verteilen. Dies führt zu vielen nützlichen Programmen, gleichzeitig wird aber das illegale Kopieren von digitalen Inhalten wie Musik, Videos oder Software zu einem nicht mehr zu vernachlässigbarem Problem. Dies betrifft die unterschiedlichsten Formen der digitalen Verbreitung, wie beispielsweise DVDs, Satelliten- und Kabelfernsehen. Daher ist es erforderlich digitale Inhalte zu verschlüsseln, um unerlaubten Empfängern die Möglichkeit zu nehmen den Inhalt illegal zu verbreiten.

Im Gegensatz zu Punkt-zu-Punkt-Verbindungen sind die zu treffenden Sicherheitsmaßnahmen bei Multicast-Übertragungen deutlich komplexer und erfordern andere Denkansätze. Dabei handelt es sich um das folgende Problem: Wie kann ein Sender einer vordefinierten und sich ständig ändernden Menge an Empfängern verschlüsselte Nachrichten zusenden und gleichzeitig den Inhalt vor unerlaubten Empfängern schützen?

Naor und Pinkas [NP00] haben auf der Suche nach dieser Antwort mit Hilfe der aus der numerischen Mathematik bekannten Polynominterpolation ein Revocation-Schema entwickelt. Ein Revocation-Schema beschreibt die Möglichkeit, einer Teilmenge von Teilnehmern die Berechtigung zu entziehen, den geheimen Schlüssel und damit verbunden die entschlüsselte Nachricht ermitteln zu können.

1.2 Zielsetzung

In dieser Studienarbeit wird die oben genannte Fragestellung genauer untersucht, jedoch mit dem Zusatz, andere Ansätze zu finden die im Naor-Pinkas-Schema angewendet werden können. Dabei wurde im ersten Analyseschritt der Bereich genauer betrachtet, der die für die Entschlüsselung notwendigen Informationen enthält. An dieser Stelle wurde überprüft, ob andere Interpolationsverfahren in diesem Schema angewendet werden können und welche Auswirkungen diese gegenüber der von Naor und Pinkas [NP00] eingesetzten Polynominterpolation haben. Von Interesse sind dabei die ausgeführten Berechnungsschritte eines Empfängers um den geheimen Schlüssel bestimmen zu können.

1.3 Verwandte Arbeiten

Viele hier durchgeführten Analysen basieren auf der von Naor und Pinkas [NP00] veröffentlichten Arbeit über ein Revocation-Schema mit Hilfe der Lagrange-Interpolation. In der Veröffentlichung wurden mehrere Möglichkeiten eines solchen Schemas ausführlich beschrieben. Die vorliegende Studienarbeit bezieht sich jedoch ausschließlich auf das vorgestellte *Schema für viele Revocations*. Zusätzlich wurde in der vorgestellten Arbeit von Naor und Pinkas das Revocation-Schema mit einem sogenannten *Self Enforcement* und *Tracing* kombiniert. Letzteres wurde mit einem *Black-Box-Confirmation-Test* realisiert, bei dem überprüft wird, ob es sich bei einem Teilnehmer um jemanden handelt, der unerlaubt digitalen Inhalt verbreitet oder seinen persönlichen Schlüssel an andere weitergegeben hat. Diese Teilnehmer werden als *Traitors* bezeichnet.

Eine weitere verwandte Arbeit zu diesem Thema ist die Veröffentlichung mit dem Titel *Broadcast Encryption* von Fiat und Naor [FN93]. Dabei werden grundlegende Schemata vorgestellt, die dem Sender die Möglichkeit geben, die verschlüsselte Nachricht an eine beliebige und sich ständig ändernde Teilmenge von Empfängern zu versenden.

Das Revocation-Schema in der von Kumar et al. [KRS99] veröffentlichten Arbeit ist dem von Naor und Pinkas vorgestellten sehr ähnlich. Jedoch ist es mit ihrer Methode (*one-time revocation*) möglich, einer bestimmten Anzahl von unerlaubten Teilnehmern den Zugriff auf die geheime Information zu verwehren und gleichzeitig ist sie gegen eine vollständige Vereinigung aller Unberechtigten sicher.

1.4 Gliederung

Der erste Abschnitt dieser Studienarbeit enthält eine ausführliche Beschreibung der Grundlagen, wobei das Hauptaugenmerk hierbei auf dem 1979 entwickelten Secret-Sharing-Verfahren von Adi Shamir liegt. Zusätzlich wird das Single-Revocation-Schema aus Kapitel 2.1 der Veröffentlichung von Naor-Pinkas [NP00] detailliert beschrieben.

Der nächste und umfangreichste Abschnitt behandelt das von Naor und Pinkas entwickelte Schema für viele Revocations mit dem Versuch, dieses mit verschiedenen Polynominterpolationen erfolgreich umzusetzen. Hierbei wird die im Naor-Pinkas-Revocation-Schema verwendete Lagrange-Interpolation detailliert beschrieben. Ferner wird versucht, die auf die Newton-Basis aufbauende Newton-Interpolation und die Spline-Interpolation erfolgreich anzuwenden und ihre Vor- und Nachteile gegenüberzustellen. Zusätzlich wird der erforderliche Aufwand eines Teilnehmers bei den einzelnen Interpolationen ermittelt, der entsteht, um die geheime Information entschlüsseln zu können.

Im letzten Kapitel der Studienarbeit wird das Asmuth-Bloom-Secret-Sharing-Verfahren genauer analysiert. Dieses verwendet den aus der Zahlentheorie bekannten chinesischen Restsatz anstelle der Polynominterpolation aus der numerischen Mathematik. Zudem wird die Fragestellung herangezogen, ob das Asmuth-Bloom-Verfahren in ein Revocation-Schema umgewandelt werden kann, basierend auf dem Prinzip von Naor-Pinkas.

2. Grundlagen

In dieser Studienarbeit häufig verwendete Begriffe werden in diesem Abschnitt genauer erläutert. Des Weiteren werden die zugrundeliegenden Definitionen beschrieben, die in der gesamten Studienarbeit als Vorraussetzung gelten. Hierbei wird auf das von Adi Shamir entwickelte Secret-Sharing-Verfahren eingegangen, welches das Grundgerüst für alle vorgestellten Verfahren bildet. Zusätzlich wird das Single-Revocation-Schema von Naor und Pinkas [NP00] erklärt.

2.1 Notationen und Definitionen

Ein *Revocation-Schema* bezeichnet eine Methode um unerlaubten Teilnehmern aus einer bestimmten Teilnehmermenge die Möglichkeit zu verweigern, digitale Inhalte zu entschlüsseln. Dieser Prozess ist auch als *User Exclusion* oder *Blacklisting* bekannt.

In dieser Studienarbeit werden Revocation-Schemata vorgestellt, wobei von einem gleichbleibenden Szenario ausgegangen wird: Eine Gruppe von Teilnehmern erhält vom Gruppencontroller (auch als Sender oder Dealer bezeichnet) digitale Inhalte. Diese Inhalte, wie z. B. digitale Musik oder ein TV-Programm, werden über Kanäle wie das Internet, Satellitenübertragung, Kabel oder einer DVD übertragen. Die Informationen sind verschlüsselt und der für die Entschlüsselung notwendige Schlüssel ist anfänglich jedem Teilnehmer bekannt. Nach einer gewissen Zeit lernt der Gruppencontroller Teilnehmer kennen, die gegen die Bedingungen der Nutzungslizenz verstoßen. Beispielsweise bei Nichtzahlung des Abonnements oder bei Veröffentlichung oder Weitergabe des geheimen Schlüssels von einem zum anderen Teilnehmer. Dadurch ist der Gruppencontroller gezwungen diesen Traitors das Recht zu nehmen weiteren Inhalt entschlüsseln zu können.

Um die Effizienz eines Revocation-Schemas bestimmen zu können, müssen die wichtigen Faktoren der angewendeten Methoden bestimmt werden. Diese werden in drei Kategorien unterteilt:

- **Kommunikationsaufwand**
  Wie viel Information muss zwischen dem Gruppencontroller und einem einzelnen Teilnehmer versendet werden, um beispielsweise den geheimen Schlüssel zu erneuern? Hierbei wird besonders die Länge der Nachricht untersucht.

- **Speicheraufwand**
  Wie viele und welche Informationen muss der Teilnehmer bei sich speichern, um den neuen Schlüssel berechnen zu können? Im Allgemeinen wird hier die Anzahl an Schlüsseln betrachtet, die ein Teilnehmer für die Berechnung benötigt.

- **Berechnungsaufwand**
  Wie viele Berechnungen sind vor allem bei einem einzelnen Teilnehmer nötig, um einen neuen Schlüssel bestimmen zu können?

Das Verfahren von M. Naor und B. Pinkas ist deswegen interessant, da keine dieser Faktoren von der Gesamtzahl der Teilnehmer abhängt. Im späteren Verlauf wird deutlich, dass die Länge des Schlüssels konstant und die Kommunikation bzw. der Berechnungsaufwand linear zu der Anzahl der unerlaubten Teilnehmer ist.

Für die Verteilung der benötigten Schlüssel an die Teilnehmer wird ein Secret-Sharing-Verfahren benötigt. In der Kryptopgraphie bezeichnet das *Secret-Sharing* eine Technik der Geheimnisteilung, wobei das Geheimnis von einem Gruppencontroller auf eine Anzahl von Instanzen aufgeteilt wird, den sogenannten *Shares*. Jeder Instanz wird ein eindeutiger *Share* zugewiesen und es ist nicht möglich, ohne Zuhilfenahme zusätzlicher Shares der weiteren Instanzen das Geheimnis zu entschlüsseln. Dieser Fall wird als einfaches Secret-Sharing-Verfahren bezeichnet. Ist jedoch nur eine gewisse Untermenge der Instanzen erforderlich um das Geheimnis zu rekonstruieren, spricht man von einem erweiterten Secret-Sharing-Verfahren oder einem (k, n)-Schwellenwert-Schema.

Es existieren mehrere Möglichkeiten, das oben genannte Schema zu konstruieren. In dieser Studienarbeit wird sich größtenteils auf die aus der numerischen Mathematik bekannte Polyninterpolation konzentriert.

Definitionen

Sei $\mathcal{N}$ die Menge aller Teilnehmer mit $n = |\mathcal{N}|$. Ferner existiert eine Teilmenge $\mathcal{R} \subset \mathcal{N}$ mit $t = |\mathcal{R}|$ Teilnehmern, die nicht das Recht und die Möglichkeit besitzen sollen, die übertragene Nachricht zu entschlüsseln. Ferner sei $\mathcal{F}$ ein endlicher Körper und ein zufällig ausgewähltes Element $\mathcal{S} \in \mathcal{F}$ der geheime Schlüssel, welcher an die verschiedenen Teilnehmer verteilt und ausschließlich von den Teilnehmern $u \in \mathcal{N} \setminus \mathcal{R}$ bestimmt werden soll.

2.2 Shamirs-Secret-Sharing-Schema

Mit Hilfe des 1979 entwickelten Secret-Sharing-Verfahrens von Adi Shamir [S79] ist es möglich, ein Geheimnis auf mehrere Instanzen aufzuteilen, wobei nur eine gewisse Untermenge der Shares erforderlich ist, damit ein Teilnehmer das Geheimnis rekonstruieren kann. Dieses Schema wird auch als (k, n)-Schwellenwert-Schema bezeichnet, da nur k der insgesamt n Shares benötigt werden, um das Geheimnis eindeutig bestimmen zu können.
Hierzu erstellt der Gruppencontroller ein Polynom der Form

$$P(x) = \mathcal{S} + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}.$$

Im nächsten Schritt erstellt dieser zudem n Wertepaare $(x_i, P(x_i))$ und verteilt diese an die Teilnehmer. Hier entspricht der Wert x_i der öffentliche und $P(x_i)$ der private Teil des Shares. Letzteres müssen von den Teilnehmern geheim gehalten werden.
Laut dem Fundamentalsatz der Algebra [G15] kann ein Polynom P vom Grad k, welches durch

angegebenen k-Stützwerte $P(x_i)$ verläuft mit Hilfe von k verschiedenen Wertepaaren $(x_i, P(x_i))$ eindeutig bestimmt werden. Im Vergleich zum einfachen Secret-Sharing-Verfahren eine verbesserte und effizientere Situation, da die Teilnehmer nicht alle existierenden Shares für die Rekonstruktion benötigen. Wie im Aschnitt 3.2 und 3.3 später gezeigt wird, existieren verschiedene Möglichkeiten und Methoden, um das Polynom eindeutig zu rekonstruieren.

2.3 Single-Revocation-Schema

In diesem Abschnitt wird das Single-Revocation-Schema aus dem Kapitel 2.1 der Veröffentlichung von Naor-Pinkas [NP00] beschrieben. Dieses Verfahren basiert auf das Shamirs-Secret-Sharing-Schema und kann für einzelne Revocations (*single revocations*) von bis zu t Teilnehmer genutzt werden. Das Schema wird hierbei in folgende zwei Phasen gegliedert:

1. **Vorbereitung**
   Gegeben sei ein Körper $\mathcal{F}$. Aus diesem wählt der Gruppencontroller ein zufälliges Element $\mathcal{S} \in \mathcal{F}$, sodass dieses als Schlüssel eines symmetrischen Verschlüsselungsschemas benutzt werden kann. Außerdem generiert der Gruppencontroller in dieser Phase ein zufälliges, reeles Polynom P vom Grad t über $\mathcal{F}$ und legt mit

   $$\mathcal{S} := P(0)$$

   das Geheimnis fest. Im nächsten Schritt werden n Wertepaare $I_{u_i} = (u_i, P(u_i))$ mit $i = 0, \ldots, n$ erstellt, wobei $\forall x_i \neq 0$ gelten muss. Anschließend werden diese an die jeweiligen Teilnehmer verteilt.

2. **Rekonstruktion**
   In dieser Phase sollen die Teilnehmer $u_i \in \mathcal{N} \setminus \mathcal{R}$ das Geheimnis $\mathcal{S}$ ermitteln können, wobei gleichzeitig die t Teilnehmer $u_{r_i} \in \mathcal{R}$ dies nicht können sollen.
   Hierzu lernt der Gruppencontroller mit der Zeit die Identitäten der t Teilnehmer $u_{r_i} \in \mathcal{R}$ kennen und übermittelt folgende Nachricht an alle Teilnehmer $u_i \in \mathcal{N}$:

   $$< u_{r_1}, P(u_{r_1}) >, < u_{r_2}, P(u_{r_2}) >, \ldots, < u_{r_t}, P(u_{r_t}) > .$$

   Da jeder Teilnehmer sein Wertepaar I_{u_i} bei sich gespeichert hat, können nach dem Fundamentalsatz der Algebra [G15] ausschließlich die Teilnehmer $u_i \in \mathcal{N} \setminus \mathcal{R}$ den Schlüssel $\mathcal{S} = P(0)$ eindeutig bestimmen, da nur diese Teilnehmer $t + 1$ verschiedene Stützstellen und Stützwerte besitzen. Ein Teilnehmer $u_i \in \mathcal{R}$ besitzt mit Hilfe der übertragenen Nachricht und der bei sich gespeicherten Information lediglich t voneinander verschiedenen Wertepaare. Das Geheimnis $\mathcal{S}$ kann von den unerlaubten Teilnehmern nicht eindeutig bestimmt werden kann.

Nachteil des Single-Revocation-Schemas

Aufgrund dessen, dass der geheime Schlüssel $\mathcal{S}$ nach der erfolgreichen Rekonstruktion des Polynoms P eindeutig bestimmbar ist, muss $\mathcal{S}$ bei Veränderung der Menge der unerlaubten Empfänger $\mathcal{R}$ neu gewählt werden. Daher ist es notwendig, dass der Gruppencontroller ein neues, beliebiges Polynom P wählt und jedem Teilnehmer vor der eigentlichen Übertragung des Chriffrats die persönlichen Shares über einen privaten Kanal übermittelt. Dies ist erforderlich, da beispielsweise ein im ersten Durchlauf noch legaler Empfänger im zweiten Durchlauf zu einem unerlaubten Empfänger werden kann und daher nicht mehr die Möglichkeit besitzen soll, das Geheimnis bestimmen zu können. Dadurch muss in dieser Situation das Geheimnis $\mathcal{S}_1$ sich im zweiten Durchlauf vom Geheimnis $\mathcal{S}_2$ unterscheiden.

3. Naor-Pinkas-Revocation-Schema

Das von M. Naor und B. Pinkas vorgestellte Verfahren stellt eine abstrakte Definition eines Revocation-Schemas dar. Die Hauptaufgabe des Verfahrens bezieht sich darauf, auf geschickte Art und Weise eine Verteilung des geheimen Schlüssels sicherzustellen, sodass aus einer beliebig großen Menge von erlaubten Empfängern bis zu maximal t Empfänger ausgeschlossen werden können.

In der vorgestellten Arbeit werden drei Schemata vorgestellt, um unerlaubten Teilnehmern den Zugriff auf den Schlüssel und somit auf den Inhalt zu verweigern. Dabei konzentriert sich diese Studienarbeit ausschließlich auf das zweite vorgestellte Schema aus Kapitel 2.2 in [NP00].

3.1 Verfahren

Das nun vorgestellte Schema ist für eine Anzahl von maximal t zu sperrende Empfänger sicher und basiert auf der Decisional-Diffie-Hellman-Annahme.

3.1.1 Decisional-Diffie-Hellman-Annahme

Gegeben sei eine endliche und zyklische Gruppe $\mathcal{G}$ mit einem Erzeuger g. Das bedeutet, dass zu jedem Element f der Gruppe $\mathcal{G}$ eine Ganzzahl z mit $f = g^z$ existiert. Bei der DDH-Annahme geht es darum, dass kein effizienter Algorithmus existiert, der zwischen den Verteilungen $< g^a, g^b, g^{ab} >$ und $< g^a, g^b, g^c >$ mit zufälligen und beliebigen a, b, c aus $[1, \ldots, |\mathcal{G}|]$ unterscheiden kann. Anders gesagt, kann unter der DDH-Annahme kein Angreifer den Wert g^{ab} von einem zufälligen Gruppenelement unterscheiden.

3.1.2 Schema für viele Revocations

Das vorgestellte Schema aus Kapitel 2.2 der Veröffentlichung von Naor und Pinkas [NP00] ist über eine Untergruppe $\mathbb{Z}_q$ mit Ordnung q in $\mathbb{Z}_p^*$ definiert, wobei p eine Primzahl und $q|p-1$ definiert ist. Außerdem sei ein Erzeuger g aus $\mathbb{Z}_q$ gegeben, sodass die Decisional Diffie-Hellman Annahme für $\mathbb{Z}_q$ und g zutrifft.

Dieses Schema kann für viele Revocations genutzt werden, solange verhindert wird, dass mehr als t Teilnehmer ausgeschlossen werden sollen. Zudem wird später gezeigt, dass das Verfahren bis auf eine Vereinigung von $(t-1)$ unerlaubte Teilnehmer sicher ist.

Um das Schema umzusetzen, werden folgende zwei Phase benötigt:

- **Initialisierungsphase**
  In dieser Phase generiert der Gruppencontroller ein beliebiges Polynom P vom Grad t welches über die Gruppe $\mathbb{Z}_q$ definiert ist. Nach diesem Schritt veröffentlicht der Gruppencontroller über einen privaten Kanal p und q und sendet jedem Teilnehmer u seinen eindeutigen und persönlichen Schlüssel
  $$\mathcal{K}_u =< I_u, P(I_u) > .$$
  Dabei entspricht der Wert I_u der öffentliche Teil des Shares und beschreibt gleichzeitig das zum Teilnehmer u gehörende Identifizierungsmerkmal. Wichtig zu betonen ist, dass diese Phase ein einziges Mal für alle später folgenden Revocations durchgeführt werden muss.

- **Revocation-Phase**
  In dieser Phase wird den unerlaubten Teilnehmern die Möglichkeit genommen, die vom Gruppencontroller übermittelte Nachricht zu ermitteln.
  Dabei müssen die Identitäten der t Teilnehmer $\mathcal{R} = \{I_{u_1}, \ldots, I_{u_t}\}$ herausgefunden werden, die aus der Menge der erlaubten Empfänger ausgeschlossen werden sollen. Da dies nicht Teil des Revocation-Schemas ist, wird dies hier nicht näher erläutert.
  Ist die unerlaubte Teilnehmermenge $\mathcal{R}$ bekannt, wählt der Gruppencontroller ein zufälliges, beliebiges $r \in \mathbb{Z}_q$ und setzt mit
  $$\mathcal{S} = g^{rP(0)}$$
  als neuen Schlüssel fest. Zudem werden für alle Empfänger $I_{t_i} \in \mathcal{R}$ der Wert $g^{rP(I_{t_i})}$ berechnet. Im nächsten Schritt übermittelt der Gruppencontroller jedem Empfänger $u \in \mathcal{N}$ die berechneten Werte und den persönlichen Schlüssel der unerlaubten Empfängern:
  $$< g^r, g^{rP(I_{u_1})}, g^{rP(I_{u_2})}, \ldots, g^{rP(I_{u_t})}, I_{u_1}, I_{u_2}, \ldots, I_{u_t} >$$

Jeder Empfänger $u \in \mathcal{N} \setminus \mathcal{R}$ besitzt mit der bei sich gespeicherten Information $\mathcal{K}_u =< I_u, P(I_u) >$ die notwendige $(t+1)$-te Information um das Polynom P vom Grad t eindeutig bestimmen zu können. Der Empfänger u berechnet $(g^r)^{P(I_u)}$ und besitzt aufgrund dessen $(t+1)$ Stützstellen $I_u, I_{u_1}, \ldots I_{u_t}$ und die dazugehörenden $(t+1)$ Stützwerte $(g^r)^{P(I_u)}, (g^r)^{P(I_{u_1})}, \ldots, (g^r)^{P(I_{u_t})}$. Mit Hilfe einer beliebigen Polynominterpolation wird das Interpolationspolynom eindeutig rekonsturiert. Der letzte Schritt beim Empfänger besteht aus der Berechnung von $(g^r)^{P(0)}$, um den Schlüssel $\mathcal{S}$ zu erhalten.

Ein Empfänger $u_{t_i} \in \mathcal{R}$ besitzt währenddessen mit Hilfe der übertragenen Nachricht und der gespeicherten Information $\mathcal{K}_{u_i} =< I_{u_i}, P(I_{u_i}) >$ nur insgesamt t verschiedene Stützwerte und Stützstellen. Zu wenig um das Interpolationspolynom vom Grad t durch die angegebenen Stützstellen eindeutig bestimmen zu können.

3.1.3 Unerlaubte Empfängermenge $\mathcal{R}$

Sollte zu einem beliebigen Zeitpunkt $t' < t$ unerlaubte Empfänger existieren, fehlen auch jedem erlaubten Teilnehmer $(t - t')$ Stützstellen um mit Hilfe einer Polynominterpolation das Interpolationspolynom vom Grad t zu bestimmen. Dieses Problem kann durch den Gruppencontroller behoben werden:

Existieren zu einem beliebigen Zeitpunkt $t' < t$ unerlaubte Empfänger, erstellt der Gruppencontroller $(t - t')$ Dummy-Benutzer aus P, wobei diese Dummy-Benutzer von allen existierenden Teilnehmern verschieden sein müssen. Diese werden mit den realen unerlaubten Teilnehmer an alle $u \in \mathcal{N}$ versendet, sodass zu jedem Zeitpunkt sichergestellt wird, dass t Stützstellen übermittelt werden.

3.1.4 Mehrfachausführung

Im Abschnitt 2.3 wurde beim Single-Revocation-Schema das Problem der Mehrfachausführung erklärt. Hierbei muss der Gruppencontroller bei Veränderung der Menge $\mathcal{R}$ ein neues Polynom P und somit einen neuen Schlüssel $\mathcal{S}$ bestimmen. Dadurch ist es erforderlich, dass jeder Teilnehmer u_i eine neue persönliche Information $\mathcal{K}_{u_i} =< I_{u_i}, P(I_{u_i}) >$ zugewiesen bekommt.

Diese Schritte sind beim Schema für viele Revocations nicht notwendig. Gegeben sei die Menge $\mathcal{R} = \{I_{u_1}, \ldots, I_{u_{t-1}}, I_{u_d}\}$ mit einem Dummy-Benutzer I_{u_d}. Sollte ein bei der ersten Übertragung erlaubter Teilnehmer u_j bei der zweiten Übertragung nicht mehr das Recht besitzen, den Schlüssel $\mathcal{S}$ zu bestimmen, wird der Dummy-Benutzer aus der Menge $\mathcal{R}$ durch den Teilnehmer I_{u_j} ersetzt. Die Menge sieht demnach wie folgt aus: $\mathcal{R} = \{I_{u_1}, \ldots, I_{u_{t-1}}, I_{u_j}\}$. In der Revocation-Phase wählt der Gruppencontroller ein beliebiges $r' \in \mathbb{Z}_q$ und erhält somit einen neuen Schlüssel $\mathcal{S}'$. Die übertragene Nachricht sieht demnach wie folgt aus:

$$< g^{r'}, g^{r'P(I_{u_1})}, g^{r'P(I_{u_2})}, \ldots, g^{r'P(I_{u_{t-1}})}, g^{r'P(I_{u_d})}, I_{u_1}, I_{u_2}, \ldots, I_{u_{t-1}}, I_{u_d} >$$

Jeder Teilnehmer $u_i \in \mathcal{N} \setminus \mathcal{R}$ besitzt mit seiner persönlichen Information $\mathcal{K}_{u_i} =< I_{u_i}, P(I_{u_i}) >$ und der Berechnung von $(g^r)^{P(I_{u_i})}$ weiterhin $(t+1)$ verschiedene Stützwerte und Stützstellen. Das Polynom kann eindeutig an der Stelle $\mathcal{S}' = g^{r'P(0)}$ bestimmt werden.

Der Gruppencontroller muss zu keinem Zeitpunkt ein neues Polynom bestimmen, da der Wert $P(0)$ durch die Zuhilfenahme des Erzeugers g aus $\mathbb{Z}_q$ keinem Teilnehmer bekannt ist.

3.1.5 Empfänger hinzufügen

Der Gruppencontroller kann jederzeit neue Empfänger hinzufügen, auch wenn diese nach der Initialisierungsphase hinzukommen. Sei u_{new} der neue Empfänger. Der Gruppencontroller weist ihm eine neue, eindeutige und noch nicht verwendete Identität $I_{u_{new}}$ zu und sendet das ihm zugeordnete Wertepaar $\mathcal{K}_{u_{new}} =< I_{u_{new}}, P(I_{u_{new}}) >$ über einen privaten Kanal mit. Bei der nächsten Übertragung kann der Benutzer bereits $\mathcal{S} = g^{rP(0)}$ mit Hilfe der übermittelten t Stützstellen und -werte berechnen.

3.1.6 Empfänger aus $\mathcal{R}$ entfernen

Gegeben sei ein Empfänger u_l der bei der letzten übertragenen Nachricht zum Zeitpunkt t_{-1} den Schlüssel $\mathcal{S}$ nicht bestimmen durfte, ihm jedoch beim darauffolgenden Zeitpunkt t_0 dieses Recht erneut gegeben wurde. Anders ausgedrückt, war der Empfänger zum Zeitpunkt t_{-1} $u_l \in \mathcal{R}$ und zum Zeitpunkt t_0 $u_l \in \mathcal{N} \setminus \mathcal{R}$. In diesem Szenario reicht es aus, dass der Gruppencontroller den Empfänger u_l aus der übertragenen Nachricht als unerlaubten Empfänger entfernt und durch einen Dummy-Benutzer oder einen anderen, unerlaubten Empfänger ersetzt. Der Empfänger benötigt keinen neuen persönlichen Schlüssel $\mathcal{K}_{u_l}$ und der Gruppencontroller braucht keinen neuen Schlüssel $\mathcal{S}$ oder ein Polynom P zu bestimmen. Denn die ihm zugewiesene persönliche Information $P(u_l)$ ist lediglich dem Gruppencontroller und dem Teilnehmer u_l bekannt. Durch das Verlagern der Information in den Exponenten mit Hilfe des Erzeugers g und dem vor jeder Übertragung zufällig ausgewählten r bleibt $P(u_l)$ aufgrund der Berechnung $g^{rP(u_l)}$ den anderen Empfängern unbekannt.

3.1.7 Speicheraufwand

Jeder Teilnehmer u muss seine persönliche Information $\mathcal{K}_u$ speichern. Das sind in diesem Fall ein Element $I_u \in \mathbb{Z}_q$ und der dazugehörige Wert des $P(I_u) \in \mathbb{R}$. Für den geheimen Schlüssel muss jeder Empfänger lediglich ein weiteres Element aus $\mathbb{Z}_q$ speichern.

Die Revocation-Nachricht hat eine Länge von $O(t)$. Genauer gesagt enthält die Nachricht $(t+1)$

Elemente aus Z_p^* und t Elemente aus Z_q.

Der Gruppencontroller muss, im Gegensatz zu einem Teilnehmer, etwas mehr Speicherkapazität aufweisen. Dieser speichert das generierte Polynom P und alle vergebenen Stützstellen, um eine doppelte Zuweisung zweier Teilnehmer zu vermeiden. Zudem müssen alle unerlaubten Empfänger markiert werden, um diese in der Nachricht übermitteln zu können.

3.1.8 Sicherheit

Im Abschnitt 3.1.4 wurde gezeigt, dass das Verfahren von Naor und Pinkas wiederholt für Revocations angewendet werden kann. Das Verfahren ist auch sicher wenn t Teilnehmer sich zusammenschließen. Solch ein Zusammenschluss von Benutzern kann unter der DDH-Annahme zu keinem Zeitpunkt zwischen dem Gruppenschlüssel und einem zufälligen Wert unterscheiden.

Die ausführliche Beweisidee der letzten Aussage kann in der Veröffentlichung von Naor und Pinkas [NP00] nachgelesen werden.

3.2 Naor-Pinkas-Polynom-Interpolation mit Lagrange

Im vorherigen Abschnitt 3.1 wurde das von M. Naor und B. Pinkas entwickelte Verfahren vorgestellt, jedoch ohne Angabe eines konkreten Interpolationsverfahrens für das Polynom P. Die im Paper angewendete Lagrange-Interpolation wird in diesem Abschnitt das Hauptthema sein.

3.2.1 Lagrange-Interpolationsaufgabe für Polynome

Mit Hilfe der Lagrange-Interpolation wird in der numerischen Mathematik nach einem eindeutigen Polynom gesucht, welches exakt durch angegebene, paarweise verschiedene Punkte verläuft.

Dabei seien $(t + 1)$ paarweise verschiedene Stützstellen $x_i \in \mathbb{R}$ mit $i = 0, \ldots, t$ und die zugehörigen Werte y_i, die sogenannten Stützwerte, bekannt. Das Problem bei der Interpolation lautet wie folgt: Gesucht wird nach einem reelen Polynom P von Höchstgrad t, welches alle Gleichungen $P(x_i) = y_i$ erfüllt. Dabei bezeichnet man die Polynome

$$\lambda_i(x) \coloneqq \prod_{j=0,\, j\neq i}^{t} \frac{x - x_j}{x_i - x_j}$$

als die zur Stützstelle x_i gehörenden Lagrange-Grundpolynome. Das Interpolationsproblem lässt sich in der Lagrange-Form mit Hilfe der λ_i direkt darstellen:

$$P(x) \coloneqq \sum_{i=0}^{t} \lambda_i(x) \cdot y_i$$

Nach dem Fundamentalsatz der Algebra [G15] existiert ein solches Polynom stets und ist eindeutig bestimmbar. Das Polynom wird als Lagrange'sches Interpolationspolynom in der Langrange'schen Darstellung bezeichnet. Die eben aufgeführte Formel unterstreicht zudem, dass eine lineare Abhängigkeit zwischen P und den Stützstellen y_i besteht. Wie später gezeigt wird, kann diese Eigenschaft in gewissen Situationen von Vorteil sein.

3.2.2 Lagrange-Interpolation im Naor-Pinkas-Verfahren

Der Gruppencontroller generiert ein zufälliges Polynom P vom Grad t über $\mathbb{Z}_q$ und erstellt für jeden Empfänger u ein persönliches Wertepaar $K_{u_0} =< I_{u_0}, P(I_{u_0}) >$. Dieses wird über einen privaten Kanal an den Empfänger u_0 gesendet, wobei I_{u_0} ein nichtgeheimes Indentifizierungsmerkmal beschreibt. Zudem wählt der Gruppencontroller ein zufälliges $r \in \mathbb{Z}_q$ und setzt $g^{rP(0)}$ als neuen Schlüssel $\mathcal{S}$. Nach diesen Schritten übermittelt der Gruppencontroller folgende Nachricht in Klartext an alle Empfänger:

$$< g^r, g^{rP(I_{u_1})}, g^{rP(I_{u_2})}, \ldots, g^{rP(I_{u_t})}, I_{u_1}, I_{u_2}, \ldots, I_{u_t} >$$

Mit Hilfe der Lagrange-Interpolation soll jeder Empfänger $u \in \mathcal{N} \setminus \mathcal{R}$ das vom Gruppencontroller generierte Polynom eindeutig bestimmen können. Zu Beginn berechnet u den Wert $(g^r)^{P(I_{u_0})}$ und wertet das erste Lagrange-Grundpolynom

$$\lambda_0 \coloneqq \lambda_0(0) = \prod_{j=1}^{t} \frac{-I_{u_j}}{I_{u_0} - I_{u_j}} = \prod_{j=1}^{t} \frac{I_{u_j}}{I_{u_j} - I_{u_0}}$$

an der gewünschten Stelle $x = 0$ aus.

Die beiden Werte kann lediglich der Empfänger u_0 berechnen, da das Wertepaar $K_{u_0} =< I_{u_0}, P(I_{u_0}) >$ nur ihm bekannt ist. Im nächsten Schritt werden mit Hilfe der übermittelten Nachricht die weiteren erforderlichen t Lagrange-Grundpolynome berechnet:

$$\lambda_i = \prod_{j=0, j\neq i}^{t} \frac{I_{u_j}}{I_{u_j} - I_{u_i}}$$

Die Berechnung des Schlüssels $\mathcal{S}$ erfolgt nach folgenden Berechnungsschritten:

$$\begin{aligned}\mathcal{S} &= (g^r)^{P(0)} \\ &= (g^r)^{\sum_{i=0}^{t} \lambda_i P(I_{u_i})} \\ &= \prod_{i=0}^{t} (g^r)^{\lambda_i P(I_{u_i})} \\ &= \prod_{i=0}^{t} (g^{rP(I_{u_i})})^{\lambda_i}\end{aligned}$$

Mit Hilfe des vom Gruppencontroller übertragenen Wertes g^r und der persönlichen Information $K_{u_0} =< I_{u_0}, P(I_{u_0}) >$ kann jeder Empfänger $u \in \mathcal{N} \setminus \mathcal{R}$ das Polynom eindeutig an der Stelle $g^{rP(0)}$ bestimmen, da nur dieser die erforderlichen $(t+1)$ Wertepaare $< (I_{u_i}, P(I_{u_i}) >$ mit $i = 0, \ldots, t$ besitzen. Einem Empfänger $u_r \in R$ fehlt das notwendige $(t+1)$-te Wertepaar, da seine Stützstelle bereits in der übermittelten Nachricht vorkommt und somit doppelt vorhanden ist.

3.2.3 Aufwand

Um den Rechenaufwand der Lagrange-Interpolation im Naor-Pinkas-Verfahren ermitteln zu können, werden die jeweiligen Berechnungsschritte eines Empfängers genauer betrachtet. Die Formel für die Berechnung eines einzelnen Lagrange-Grundpolynoms lautet:

$$\lambda_i = \prod_{j=0,\, j\neq i}^{t} \frac{I_{u_j}}{I_{u_j} - I_{u_i}} = \frac{I_{u_0} \cdot \ldots \cdot I_{u_{i-1}} \cdot I_{u_{i+1}} \cdot \ldots \cdot I_{u_t}}{(I_{u_0} - I_{u_i}) \cdot \ldots \cdot (I_{u_{i-1}} - I_{u_i}) \cdot (I_{u_{i+1}} - I_{u_i}) \cdot \ldots \cdot (I_{u_t} - I_{u_i})}$$

Dabei werden im Zähler insgesamt $(t-1)$ Mutliplikationen (M) und im Nenner $(t-1)$ M und t Additionen (A) benötigt. Zudem kommt zum Schluss eine Division (D) dazu. Zusammengefasst ergibt das für ein einzelnes Lagrange-Grundpolynome folgenden Aufwand:

$$2 \cdot (t+1)\, M + 1\, D + t\, A.$$

Die Berechnung muss für jede der $(t+1)$ Stützstellen ausgeführt werden. Der Rechenaufwand für alle Lagrange-Grundpolynome steigt dadurch auf insgesamt

$$\begin{aligned}&(t+1) \cdot (2 \cdot (t+1)\, M + 1\, D + t\, A) \\ &= 2 \cdot (t+1)^2\, M + (t+1)\, D + t \cdot (t+1)\, A \end{aligned} \tag{3.1}$$

an. Die nächsten Berechnungen erfolgen bei der Ermittlung des geheimen Schlüssels $\mathcal{S}$. Mit Hilfe des vom Gruppencontroller übertragenen Wertes g^r kann der Empfänger $(g^r)^{P(I_{u_0})}$ berechnen. Offensichtlich kostet dies eine Exponentenberechnung (E). Um das Polynom letztendlich an der Stelle $P(0)$ ermitteln zu können, muss die folgende Formel ausgeführt werden:

$$\prod_{i=0}^{t} (g^{rP(I_{u_i})})^{\lambda_i} = (g^{rP(I_{u_0})})^{\lambda_0} \cdot \ldots \cdot (g^{rP(I_{u_t})})^{\lambda_t}$$

Zum einen sind dank der übertragenen Nachricht die Werte bekannt, zum anderen müssen die Exponenten noch berechnet werden. Dies erhöht den Aufwand auf weitere $(t + 1)$ E. Der Berechnungsaufwand beträgt demnach

$$t\,M + (t + 2)\,E. \tag{3.2}$$

Nach den Ergebnissen aus (3.1) und (3.2) beträgt der Gesamtaufwand für einen Empfänger mit der Lagrange-Interpolation im Naor-Pinkas-Verfahren

$$\begin{aligned} &2 \cdot (t + 1)^2\,M + (t + 1)\,D + t \cdot (t + 1)\,A + t\,M + (t + 2)\,E \\ &= \left(2 \cdot (t + 1)^2 + t\right)\,M + (t + 1)\,D + t \cdot (t + 1)\,A + (t + 2)\,E. \end{aligned} \tag{3.3}$$

3.2.4 Fazit

Die Lagrange-Interpolation selten praktische angewendet. Zum einen ist bei einer großen Anzahl von Stützstellen der Rechenaufwand verhältnismäßig groß, da für alle $(t + 1)$ Stützstellen jeweils ein Lagrange-Grundpolynom berechnet werden muss, was zum Gesamtaufwand wie in (3.3) dargestellt führt. Zum anderen ist bei Hinzunahme lediglich einer einzelnen Stützstelle eine Neuberechnung aller λ_i notwendig.

Ein Vorteil der Lagrange-Darstellung ist jedoch, dass sich die Summen aus den aus Produkten bestehenden Brüchen besonders einfach in Form von ineinander geschachtelten Schleifen programmieren und somit einfacher in Programmen einbeziehen lassen. Zudem werden die Stützwerte unabhängig von den Stützstellen berechnet. Sobald die Lagrange-Grundpolynome λ_i bestimmt wurden, lassen sich verschiedene Sätze von Stützwerten y_i mit gleichen Stützstellen x_i schneller interpolieren. Auf das Verfahren von M. Naor und B. Pinkas bezogen, ein enormer Vorteil, da in einem Zeitraum bei Nichtveränderung der unerlaubten Empfängermenge die weiteren Empfänger $u \notin R$ durch lokale Speicherung der Lagrange-Grundpolynome λ_i nicht mehr neu berechnet werden müssen. Aufgrund dieser Eigenschaft lässt sich nach einmaliger Berechnung der λ_i und in einem Zeitraum ohne Änderung der Menge $\mathcal{R}$ der Rechenaufwand auf t Multiplikationen und $(t + 2)$ Exponentenberechnungen reduzieren. Zudem kann in diesem speziellen Zeitraum auch die zu übertragende Nachrichtenlänge um die Menge der zu sperrenden Empfänger $\mathcal{R} = \{I_{u_1}, \ldots, I_{u_t}\}$ verkürzt werden, da diese wegen den bereits berechneten und lokal gespeicherten Lagrange-Grundpolynomen λ_i nicht mehr benötigt werden. Dadurch reduziert sich die Länge der übermittelten Nachricht um t Elemente aus dem Körper $\mathcal{F}$.

3.3 Naor-Pinkas-Polynominterpolation mit Newton

In diesem Abschnitt soll gezeigt werden, dass die Lagrange-Interpolation nicht die einzige Möglichkeit für die Bestimmung des Interpolationspolynoms ist. Das Ziel in diesem Kapitel wird sein, die Newton-Interpolation im Naor-Pinkas-Verfahren gleichermaßen erfolgreich anzuwenden. Der Vergleich beider Interpolationen wird in Abschnitt 3.5 näher beschrieben.

3.3.1 Newton-Interpolationsaufgabe für Polynome

Mit der Newton-Interpolation wird wie bei der Lagrange-Interpolation nach einem eindeutigen Polynom gesucht, welches exakt durch angegebene, paarweise verschiedene Punkte verläuft.

Dabei seien $(t+1)$ paarweise verschiedene Stützstellen $x_i \in \mathbb{R}$ mit $i = 0, \ldots, t$ und die zugehörigen Stützwerte y_i bekannt. Es wird nach einem Polynom P vom Höchstgrad t gesucht, welches alle Gleichungen $P(x_i) = y_i$ erfüllt. Wir definieren mit

$$N_0(x) \coloneqq 1$$
$$N_i(x) \coloneqq \prod_{j=0}^{i-1}(x - x_j) = (x - x_0) \cdots (x - x_{i-1})$$

für $i = 1, \ldots, t$ die zur Stützstelle x_i gehörenden Newton-Basisfunktionen, sodass P mit der Newton'schen Interpolationsformel

$$P(x) \coloneqq \sum_{i=0}^{t} N_i(x) \cdot c_i \tag{3.4}$$
$$= c_o + c_1(x - x_0) + \cdots + c_t(x - x_0) \cdots (x - x_{t-1}) \tag{3.5}$$

und den Koeffizienten c_i dargestellt werden kann. Das Gleichungssystem für alle $P(x_i)$ besitzt dadurch die Form

$$\begin{pmatrix} 1 & 0 & \ldots & & 0 \\ 1 & (x_1 - x_0) & 0 & \ldots & \\ 1 & (x_2 - x_0) & (x_2 - x_0)(x_2 - x_1) & 0 & \ldots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (x_t - x_0) & \ldots & & \prod_{i=0}^{t-1}(x_n - x_i) \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} P(x_0) \\ P(x_1) \\ \vdots \\ P(x_n) \end{pmatrix}.$$

Das Ergebnis ist eine einfach strukturierte untere Dreiecksmatrix und lässt sich demnach einfach lösen. Auch hier gilt der Fundamentalsatz der Algebra. Folglich existiert ein solches Polynom stets und ist eindeutig bestimmbar. Diese Darstellung wird als Newton'sches Interpolationspolynom in der Newton'schen Darstellung bezeichnet.

Effiziente Bestimmung der Koeffizienten mit dem Schema der dividierten Differenzen

Die Koeffizienten c_i in (3.4) können mit Hilfe des Schemas der dividierten Differenzen effizienter bestimmt werden. Für die Koeffizienten gilt

$$c_i \coloneqq P[x_0, \ldots, x_i], \tag{3.6}$$

wobei $i < j$ und die dividierten Differenzen $P[x_i, \ldots, x_j]$ rekursiv durch

$$P[x_i] \coloneqq P(x_i) \tag{3.7}$$
$$P[x_i, \ldots, x_j] \coloneqq \frac{P[x_{i+1}, \ldots, x_j] - P[x_i, \ldots, x_{j-1}]}{x_j - x_i} \tag{3.8}$$

definiert werden.

Die Berechnung der Koeffizienten lässt sich wie folgt veranschaulichen:

$$
\begin{array}{ccccccccccc}
P[x_0] & & & & & & & & & & \\
 & \searrow & & & & & & & & & \\
P[x_1] & \rightarrow & P[x_0,x_1] & & & & & & & & \\
 & \searrow & & \searrow & & & & & & & \\
P[x_2] & \rightarrow & P[x_1,x_2] & \rightarrow & P[x_0,x_1,x_2] & & & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & & & & & \\
 & \searrow & & \searrow & & & \searrow & & & & \\
P[x_{t-1}] & \rightarrow & P[x_{t-2},x_{t-1}] & \rightarrow & P[x_{t-3},x_{t-2},x_{t-1}] & \cdots & \rightarrow & P[x_0 \ldots x_{t-1}] & & & \\
 & \searrow & & \searrow & & & \searrow & & \searrow & & \\
P[x_t] & \rightarrow & P[x_{t-1},x_t] & \rightarrow & P[x_{t-2},x_{t-1},x_t] & \cdots & \rightarrow & P[x_1 \ldots x_t] & \rightarrow & P[x_0 \ldots x_t]
\end{array}
$$

Dabei entspricht die obere Diagonale genau den gesuchten Koeffizienten c_i für das Polynom P. Hierbei fällt auf, dass bei Hinzunahme eines weiteren Wertepaars (x_{t+1}, y_{t+1}) lediglich die neu hinzukommende Zeile neu berechnet werden muss, um den zusätzlichen Koeffizienten $c_{t+1} = P[x_0, \ldots, x_t]$ ermitteln zu können. Die zuvor bestimmten Koeffizienten $c_0, \ldots, c_t$ müssen nicht erneut berechnet werden. Später wird gezeigt, wann diese Eigenschaft der Newton-Interpolation im Naor-Pinkas-Verfahren von Vorteil sein kann.

3.3.2 Newton-Interpolation im Naor-Pinkas-Verfahren

In diesem Abschnitt wird beschrieben, wie im Naor-Pinkas-Verfahren die Newton-Interpolation angewendet werden kann.

Hierzu sei P wieder ein vom Gruppencontroller zufällig generiertes Polynom vom Grad t über $\mathbb{Z}_q$ und $K_{u_i} =< I_{u_i}, P(I_{u_i}) >$ die über einen privaten Kanal versendete persönliche Information des Empfängers u_i mit $i \in [1, \ldots, |\mathcal{N}|]$. Ferner wählt der Gruppencontroller vor der Übertragung der Nachricht ein zufälliges $r \in \mathbb{Z}_q$ und setzt $g^{rP(0)}$ als neuen Schlüssel $\mathcal{S}$. Im nächsten Schritt versendet der Gruppencontroller folgenden Inhalt an alle Empfänger $u \in \mathcal{N}$:

$$< g^r, g^{rP(I_{u_1})}, g^{rP(I_{u_2})}, \ldots, g^{rP(I_{u_t})}, I_{u_1}, I_{u_2}, \ldots, I_{u_t} > .$$

Mit Hilfe der Newton-Interpolation soll jeder Empfänger $u_0 \in \mathcal{N} \setminus \mathcal{R}$ das vom Gruppencontroller generierte Polynom eindeutig bestimmen können. Hierzu soll der Teilnehmer u_0 das Polynom an der Stelle $\mathcal{S} = (g^r)^{P(0)}$ ermitteln können:

$$
\begin{aligned}
\mathcal{S} &= (g^r)^{P(0)} \\
&= (g^r)^{\sum_{i=0}^{t} N_i \cdot c_i} \\
&= \prod_{i=0}^{t} g^{r \cdot N_i \cdot c_i}.
\end{aligned}
$$

Zu diesem Zweck berechnet u_0 den Wert $(g^r)^{P(I_{u_0})}$ und alle notwendigen Newton-Basisfunktionen

$$N_i := N_i(0) = \prod_{j=0}^{i-1} -I_{u_j} = (-1)^i \prod_{j=0}^{i-1} I_{u_j}. \tag{3.9}$$

Die Koeffizienten c_i werden in der Newton-Interpolation mit Hilfe der rekursiven Formeln (3.7) und (3.8) ermittelt. Diese direkte Berechnung ist jedoch im Naor-Pinkas-Verfahren nicht möglich,

da den Teilnehmern die notwendigen Stützwerte $P(I_{u_i})$ nicht bekannt sind. Durch geschicktes Umformen kann das Polynom an der Stelle $\mathcal{S}$ berechnet werden. Diese sieht beim Koeffizient c_0 folgendermaßen aus:

$$g^{r \cdot c_0 \cdot N_0} = (g^{r \cdot P(I_{u_0})})^{N_0} \tag{3.10}$$

Alle notwendigen Werte sind bekannt. Beim Koeffizient c_1 sieht die Umformung wie folgt aus:

$$\begin{aligned}
g^{r \cdot c_1 \cdot N_1} &= g^{r \cdot P[I_{u_0}, I_{u_1}] \cdot N_1} \\
&= g^{r \cdot \frac{P(I_{u_1}) - P(I_{u_0})}{I_{u_1} - I_{u_0}} \cdot N_1} \\
&= g^{r \cdot (\frac{P(I_{u_1})}{I_{u_1} - I_{u_0}} - \frac{P(I_{u_0})}{I_{u_1} - I_{u_0}}) \cdot N_1} \\
&= \left((g^r)^{\frac{P(I_{u_1})}{I_{u_1} - I_{u_0}} - \frac{P(I_{u_0})}{I_{u_1} - I_{u_0}}} \right)^{N_1} \\
&= \left(\frac{(g^r)^{\frac{P(I_{u_1})}{I_{u_1} - I_{u_0}}}}{(g^r)^{\frac{P(I_{u_0})}{I_{u_1} - I_{u_0}}}} \right)^{N_1} \\
&= \left(\frac{g^{r \cdot P(I_{u_1})}}{g^{r \cdot P(I_{u_0})}} \right)^{\frac{N_1}{I_{u_1} - I_{u_0}}}
\end{aligned} \tag{3.11}$$

Aufgrund der vom Gruppencontroller übertragenen Nachricht sind alle Werte bekannt. Diese Umformungen müssen für alle c_i durchgeführt werden. Es gilt:

$$g^{r \cdot P[I_{u_0}, \ldots, I_{u_i}]} = \frac{\left(\frac{g^{r \cdot P(I_{u_i})}}{g^{r \cdot P(I_{u_{i-1}})}} \right)^{\frac{N_i}{(I_{u_i} - I_{u_0}) \cdots (I_{u_i} - I_{u_{i-1}})}}}{\begin{array}{c} \vdots \\ \vdots \end{array}} \Bigg/ \left(\frac{g^{r \cdot P(I_{u_1})}}{g^{r \cdot P(I_{u_0})}} \right)^{\frac{N_i}{(I_{u_i} - I_{u_0}) \cdots (I_{u_1} - I_{u_0})}} \tag{3.12}$$

Folglich besitzt ein Empfänger $u_0 \in \mathcal{N} \setminus \mathcal{R}$ alle notwendigen Informationen um das Polynom eindeutig lösen und an der Stelle $x = 0$ auswerten zu können.

3.3.3 Aufwand

Um den Rechenaufwand der Newton-Interpolation im Naor-Pinkas-Verfahren ermitteln zu können, werden die jeweiligen Berechnungsschritte eines Empfängers genauer analysiert. Die Formel für die Berechnung einer einzelnen Newton-Basisfunktion für $i > 0$ lautet:

$$N_i = (-1)^i \prod_{j=0}^{i-1} x_j.$$

Für $i = 0$ muss nichts berechnet werden. Für $i > 0$ werden i M und 1 E benötigt. Ferner muss diese Rechnung für alle t unerlaubten Empfänger bestimmt werden. Der Aufwand wird dementsprechend erweitert:

$$\sum_{i=1}^{t} (i \cdot M + E) = t \cdot E + \sum_{i=1}^{t} i \cdot M \tag{3.13}$$

Sind die einzelnen Newton-Basisfunktionen bestimmt, berechnet der Empfänger u_0 mit Hilfe seiner persönlichen Information $(g^r)^{P(I_{u_0})}$, was offensichtlich 1 E kostet.

Für den restlichen Aufwand müssen, Aufgrund der Umformung für die Berechnung der Koeffizienten, die notwendigen Schritte genauer betrachtet werden. Für den Koeffizienten c_0 gilt:

$$g^{r \cdot c_0 \cdot N_0} \overset{(3.10)}{=} (g^{r \cdot P(I_{u_0})})^{N_0}.$$

Diese Berechnung kostet dem Empfänger 1 E. Für den Koeffizienten c_1 sieht dies folgendermaßen aus:

$$g^{r \cdot c_1 \cdot N_1} \overset{(3.11)}{=} \left(\frac{g^{r \cdot P(I_{u_1})}}{g^{r \cdot P(I_{u_0})}} \right)^{\frac{N_1}{I_{u_1} - I_{u_0}}}$$

Aufgrund der bekannten Werte $g^{r \cdot P(I_{u_1})}$ und $g^{r \cdot P(I_{u_0})}$ beträgt der Aufwand 1A, 2 D und 1 E. Für c_2 erhalten wir mit Hilfe der Formel aus (3.12):

$$g^{r \cdot c_2 \cdot N_2} = \frac{\left(\frac{g^{r \cdot P(I_{u_2})}}{g^{r \cdot P(I_{u_1})}} \right)^{\frac{N_2}{(I_{u_2} - I_{u_0})(I_{u_2} - I_{u_1})}}}{\left(\frac{g^{r \cdot P(I_{u_1})}}{g^{r \cdot P(I_{u_0})}} \right)^{\frac{N_2}{(I_{u_2} - I_{u_0})(I_{u_1} - I_{u_0})}}}$$

was folgenden Berechnungsaufwand kostet:

$$3 \cdot D + 2 \cdot (2A + 1M + 1D + 1E).$$

Allgemein auf einen Koeffizienten c_i mit $i > 0$ bezogen ergibt sich dementsprechend der Aufwand:

$$(2^i - 1) \cdot D + i \cdot (iA + (i-1)M + 1D + 1E).$$

Aufgrund der Umformungen benötigt ein Empfänger u_0 für die Berechnung:

$$1E + \sum_{i=1}^{t} \left(i \cdot (iA + (i-1)\, M + 1D + 1E) + (2^i - 1) \cdot D \right) \tag{3.14}$$

Der Gesamtaufwand für die Newton-Interpolation im Naor-Pinkas-Verfahren für einen einzelnen Empfänger u_0 beträgt nach (3.13) und (3.14) insgesamt

$$1\,E + \sum_{i=1}^{t} (i\,M + E) + 1\,E + \sum_{i=1}^{t} \left(i \cdot (i\,A + (i-1)\ M + 1\,D + 1\,E) + (2^i - 1)\,D \right)$$

$$= (t+2)\,E + \sum_{i=1}^{t} \left(i \cdot (i\,A + i\,M + 1\,D + 1\,E) + (2^i - 1)\,D \right) \tag{3.15}$$

$$= 2\,E + \sum_{i=1}^{t} \left(i \cdot (i\,A + i\,M + 1\,D + 1\,E) + (2^i - 1)\,D + 1\,E \right) \tag{3.16}$$

3.3.4 Fazit

Im Allgemeinen ist die Newton-Interpolation für den praktischen Einsatz eher geeignet, da diese sich numerisch sehr einfach anwenden lässt und im Vergleich zur Lagrange-Interpolation mit geringerem Aufwand zum gleichen Ergebnis kommt. Das Ändern einer Stützstelle führt nicht zur kompletten Neuberechnung der Koeffizienten, da sich die rekursiv definierten Werte lediglich ab der Position der neuen Stützstelle ändern.

Dieser Vorteil ist beim Einsatz der Newton-Interpolation im Noar-Pinkas-Verfahren jedoch nicht möglich. Dies liegt daran, dass die Stützstellen für die rekursive Berechnung der Koeffizienten c_i in (3.6) nicht bekannt sind. Ein Empfänger muss für die alle Koeffizienten die Umformung aus (3.12) durchführen, was zum Gesamtaufwand aus (3.15) führt.

Zudem besitzt die Newton-Interpolation den Nachteil, dass viele Zwischenergebnisse für die Berechnung der Koeffizienten gespeichert werden müssen. Dies kann vor allem bei zustandslosen Empfängern mit beschränkter Speichermöglichkeit zu einem Problem führen. Da die Newton-Basisfunktionen nicht von den Stützwerten abhängig sind, können in einem Zeitraum ohne Veränderung der Menge $\mathcal{R}$ die Werte beim Empfänger gespeichert werden, sodass diese nicht erneut berechnet werden müssen. Aufgrund dieser Eigenschaft lässt sich der Aufwand um der Wert aus (3.13) reduzieren.

3.4 Naor-Pinkas-Polynominterpolation mit kubischen Splines

In diesem Abschnitt soll der Versuch geschildert werden, eine weitere Polynominterpolation in das Naor-Pinkas-Verfahren einzubauen: die sogenannte kubische Spline-Interpolation.

3.4.1 Kubische Splineinterpolation für Funktionen

Im Gegensatz zur Lagrange- und Newton-Interpolation wird bei der Spline-Interpolation versucht, gegeben $(n+1)$ bekannte, paarweise verschiedene Stützstellen $x_i \in \mathbb{R}$ und Stützwerte $y_i \in \mathbb{R}$ mit Hilfe stückweise stetiger Polynome eine möglichst glatte Funktion durch angegebene Punkte zu finden. Anders ausgedrückt soll eine Funktion mit möglichst geringer Krümmung bestimmt werden. Dabei wird jedes Teilintervall zwischen zwei Punkten $[x_i, x_{i+1}]$ mit $i = 0, \ldots, n-1$ durch ein Polynom $s_i : [x_i, x_{i+1}] \to \mathbb{R}$, einem Spline, beschrieben. Der einfache Ansatz besteht aus der Verwendung von Geraden zwischen jeweils zwei benachbarten Punkten. Der verbesserte Ansatz nutzt jedoch kubische Parabeln der Form $a_i x^3 + b_i x^2 + c_i x + d_i$. In diesem Fall werden letztere als kubische Splines bezeichnet. Diese werden in diesem Abschnitt näher betrachtet.

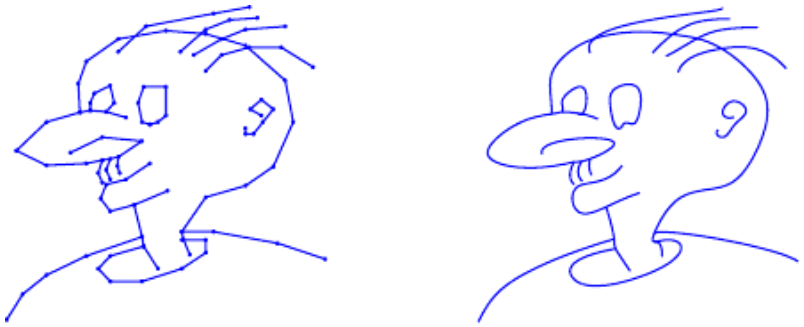

Abbildung 3.1 Glättung durch Splines. Links: Streckenzug mit Hilfe von Geraden zwischen zwei Punkten. Rechts: Streckenzug mit Hilfe von kubischen Polynomen. (Daten aus [C11] von *Gerhard Wanner*).

Was auf den ersten Blick auffällt: Aufgrund kubischer Parabeln entstehen für jedes Intervall $[x_i, x_{i+1}]$ vier unbekannte Koeffizienten a_i, b_i, c_i und d_i. Bei n verschiedenen Teilintervallen bedeutet das, dass mit Hilfe von n Gleichungen $4n$ unbekannte Koeffizienten bestimmt werden sollen. Um alle Koeffizienten eindeutig bestimmen zu können, müssen weitere Bedingungen herangezogen werden. Die erste Bedingung ist, dass der ermittelte kubische Spline zweimal stetig differenzierbar sein muss. Dadurch wird ein durchgehender Polynomzug erzwungen. Eine weitere Bedingung ist, dass kubische Splines eine Minimalitätsbedingung für die zweite Ableitung erfüllen müssen, was sie gegenüber anderen Splines besonders interessant macht.

Mathematisch formuliert wird nach einer Funktion $s : [x_0, x_n] \to \mathbb{R}$ mit $x_0 < \cdots < x_n$ und folgenden Eigenschaften gesucht:

(a) $s(x_i) = y_i$, $i = 0, \ldots, n$,

(b) $s \in C^2([x_0, x_n])$,

(c) $\int_{x_0}^{x_n} (s''(x))^2 \partial x \to min$,

(d) $s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$ mit $x \in [x_i, x_{i+1}]$ und $i = 0, \ldots, n-1$.

Um das Gleichungssystem eindeutig zu lösen, werden $4n$ Bedingungen benötigt. Für jedes der n Intervalle sind zwei Interpolationsbedingungen zu erfüllen:

$$\begin{aligned} s_i(x_i) &= y_i \ , \\ s_i(x_{i+1}) &= y_{i+1} \ . \end{aligned}$$

Dadurch entstehen $2n$ Bedingungen. Eingesetzt bedeutet das für die beiden Endpunkte eines Teilintervalls $[x_i, x_{i+1}]$

$$\begin{aligned} s_i(x_i) &= a_i(x_i - x_i)^3 + b_i(x_i - x_i)^2 + c_i(x_i - x_i) + d_i = d_i = y_i \ , \\ s_i(x_{i+1}) &= a_i(x_{i+1} - x_i)^3 + b_i(x_{i+1} - x_i)^2 + c_i(x_{i+1} - x_i) + d_i = y_{i+1} \ . \end{aligned}$$

Zudem sind alle $(n-2)$ inneren Stützstellen zweimal stetig differenzierbar. Mit den ersten beiden Ableitungen

$$\begin{aligned} s_i'(x) &= 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i \ , \\ s_i''(x) &= 6a_i(x - x_i) + 2b_i \end{aligned}$$

werden weitere $2n-2$ Bedingungen

$$\begin{aligned} s_i'(x_{i+1}) &= s_{i+1}'(x_{i+1}) \quad i = 0, \ldots, n-2 \ , \\ s_i''(x_{i+1}) &= s_{i+1}''(x_{i+1}) \quad i = 0, \ldots, n-2 \end{aligned}$$

erzeugt, die erfüllt werden müssen. Die dadurch entstehenden Gleichungen der zweiten Ableitung werden in Abhängigkeit von s'' gelöst:

$$\begin{aligned} s_i''(x_i) &= 6a_i(x_i - x_i) + 2b_i = 2b_i \\ \Leftrightarrow\ b_i &= \frac{s_i''(x_i)}{2} \\ s_i''(x_{i+1}) &= 6a_i(x_{i+1} - x_i) + 2b_i = 6a_i(x_{i+1} - x_i) + 2\frac{s_i''(x_i)}{2} \\ \Leftrightarrow a_i &= \frac{s_i''(x_{i+1}) - s_i''(x_i)}{6(x_{i+1} - x_i)} \end{aligned}$$

Diese Gleichungen in $s_i(x_{i+1})$ eingesetzt ergeben:

$$\begin{aligned} s_i(x_{i+1}) &= y_{i+1} \\ &= a_i(x_{i+1} - x_i)^3 + b_i(x_{i+1} - x_i)^2 + c_i(x_{i+1} - x_i) + d_i \\ &= \frac{s_i''(x_{i+1}) - s_i''(x_i)}{6}(x_{i+1} - x_i)^2 + \frac{s_i''(x_i)}{2}(x_{i+1} - x_i) + c_i(x_{i+1} - x_i) + y_i \\ \Leftrightarrow c_i &= \frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{2(x_{i+1} - x_i)s_i''(x_i) + (x_{i+1} - x_i)s_{i+1}''(x_{i+1})}{6} \end{aligned}$$

Mit Hilfe der ersten Ableitung werden die folgenden Berechnungen möglich:

$$\begin{aligned} s_i'(x_i) &= 3a_i(x_i - x_i)^2 + 2b_i(x_i - x_i) + c_i = c_i \\ s_{i-1}' &= 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1} \end{aligned}$$

Mit der Vorraussetzung $s_i'(x_i) = s_{i-1}'(x_i)$ wird für die Berechnung der benötigten Koeffizienten das folgende Gesamtkonzept erstellt, wobei zur besseren Darstellung $h_i := x_{i+1} - x_i$ gilt:

$$\begin{pmatrix} 2(h_0+h_1) & h_1 & 0 & \cdots & 0 \\ 0 & h_2 & 2(h_2+h_3) & h_3 & \vdots \\ \vdots & & & \ddots & \\ 0 & \cdots & & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{n-1} \end{pmatrix} = 6 \cdot \begin{pmatrix} \frac{y_2-y_1}{h_1} - \frac{y_1-y_0}{h_0} \\ \frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1} \\ \vdots \\ \frac{y_n-y_{n-1}}{h_{n-1}} - \frac{y_{n-1}-y_{n-2}}{h_{n-2}} \end{pmatrix}$$

Nach diesen Rechenschritten wird das Gesamtergebnis in Form einer Funktion mit den ermittelten s_i angegeben:

$$s(x) = \begin{cases} s_0(x), & x \in [x_0, x_1] \\ s_1(x), & x \in [x_1, x_2] \\ \quad \vdots & \\ s_{n-1}(x), & x \in [x_{n-1}, x_n] \end{cases} . \tag{3.17}$$

Da es sich bei der Spline-Interpolation um ein approximatives Verfahren handelt, spielt die Fehlerabschätzung eine wichige Rolle.

Fehlerabschätzung für kubische Splines

Für eine Funktion $f \in C^2([x_0, x_n])$ und einen zugehörigen kubischen Spline s bezüglich $x_0 < \cdots < x_n$ und den Werten $y_k = f(x_k)$, $k = 1, \ldots, n$ mit natürlichen Randbedingungen $s_0''(x_0) = 0$ und $s_{n-1}''(x_n) = 0$ gilt die folgende Fehlerabschätzung:

$$\max_{x_0 \le x \le x_n} |f(x) - s(x)| \le \frac{1}{2} h^2 \max_{x_0 \le x \le x_n} \left| f''(x) \right|$$

mit $h := \max_{0 \le k < n-1}(x_{k+1} - x_k)$. Der Beweis würde über den Rahmen der Studienarbeit hinausgehen, kann aber unter [C11] nachgelesen werden.

3.4.2 Kubische Spline-Interpolation im Naor-Pinkas-Verfahren

Die Umsetzung der Spline-Interpolation im Naor-Pinkas-Verfahren bringt einige Schwierigkeiten mit sich. Diese werden in diesem Abschnitt näher erläutert.

Wertebereich des Splines

Auf den ersten Blick fällt auf, dass die zu ermittelnde Funktion $s(x)$ im Wertebereich $[x_0, x_n]$ beschränkt ist. Jedoch wird im Naor-Pinkas-Verfahren der geheime Schlüssel $\mathcal{S}$ standardmäßig im Punkt $x = 0$ gespeichert. Um $0 \in [x_0, x_n]$ sicherzustellen, muss mindestens ein Stützwert $x_{neg} < 0$ existieren. Dies kann ein vom Gruppencontroller ausgewählter Dummy-Benutzer $u_d \in \mathcal{R}$ sein. Sollte es sich nicht um einen Dummy handeln, muss zu jeder Zeit sichergestellt werden, dass mindestens ein Wertepaar $< I_{u_i}, P(I_{u_i}) >$ eines Empfängers $u_i \in \mathcal{R}$ existiert. Grund hierfür ist, dass ein Empfänger $u \in \mathcal{N} \setminus \mathcal{R}$ nicht in der Menge $\mathcal{R}$ der übertragenen Nachricht vorkommt. Anders ausgedrückt kann in solch einer Konstellation lediglich der Empfänger u_i mit $I_{u_i} < 0$ den Schlüssel $\mathcal{S}$ berechnen, da außer ihm niemand eine Funktion $s(x)$ interpolieren kann, welche im Wertebereich $x < 0$ liegt. Diese Situation wird nochmal in der folgenden Abbildung 3.2 dargestellt.

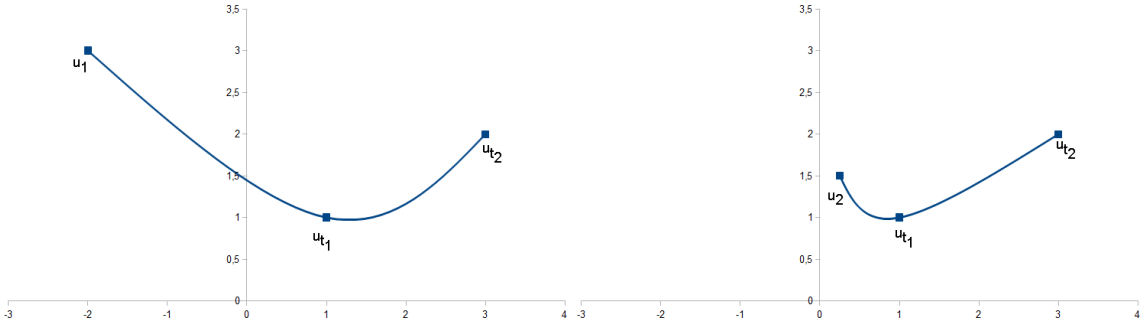

Abbildung 3.2 Wertebereich der Spline-Interpolation. Links aus der Sicht des Empfängers u_1 mit $I_{u_1} < 0$, rechts aus der Sicht des Empfängers u_2

Wert des Schlüssels

Eine weitere Untersuchung der Spline-Interpolation ergibt einen wichtigen Grund, weshalb diese im Naor-Pinkas-Verfahren nicht angewendet werden kann. Hierzu soll die in Abbildung 3.3 definierte Situation mit den Teilnehmern $I_{u_1}, I_{u_2} \in \mathcal{N} \setminus \mathcal{R}$ und $I_{u_1}, I_{u_2} < 0$ genauer betrachtet werden. Seien $I_{u_{t_1}}, I_{u_{t_2}} \in \mathcal{R}$ mit der Sortierung $I_{u_1} < I_{u_2} < 0 < I_{u_{t_1}} < I_{u_{t_2}}$ gegeben. Ferner seien den Empfängern u_1 und u_2 die Werte $s'(I_{u_{t_i}})$ und $s''(I_{u_{t_i}})$ mit $i = 1, 2$ bekannt.

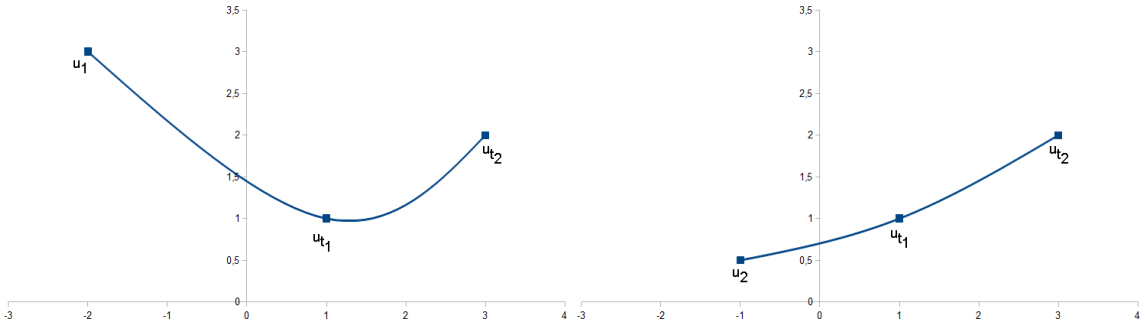

Abbildung 3.3 Wert des Schlüssels. Links aus der Sicht des Empfängers u_1, rechts aus der Sicht des Empfängers u_2

Wie hier bereits mit einigen Beispielwerten gezeigt wird, ist der Schlüssel im Punkt $x = 0$ aus der jeweiligen Sicht der Empfänger u_1 und u_2 nicht eindeutig. Dies soll nochmals betonen, dass die Spline-Interpolation nicht ohne weiteres im Naor-Pinkas-Verfahren angewendet werden kann.

3.4.3 Fazit

Die Spline-Interpolation generiert abschnittsweise glatte und stetig differenzierbare Funktionen und vermeidet starke Oszillationen im Funktionsverlauf. Jedoch kann sie nicht ohne weiteres im Naor-Pinkas-Schema angewendet werden, da es sich bei dieser Interpolation um ein approximatives Verfahren handelt.

Zum einen liegt es am definierten Wertebereich der Funktion, zum anderen am ermittelten Wert des Schlüssels. Ersteres könnte noch einfach durch einen Dummy-Benutzer oder durch Festlegung des Schlüssels an einer anderen Stelle behoben werden. Das zweite Problem ist nicht ohne weiteres offensichtlich lösbar. Durch die Verwendung eines höheren Polynomgrades könnte der ermittelte

Wert näher am tatsächlichen liegen, jedoch erhöht sich dadurch gleichzeitig der Berechnungsaufwand. Eine weitere Möglichkeit wäre, einen Schwellenwert für den Wert des Schlüssels zu benutzen. Jedoch würde sich die Sicherheit des Schemas in diesem Fall reduzieren, da der Schlüssel nicht mehr eindeutig berechnet werden, sondern sich lediglich im Schwellenwertbereich des Schlüssels befinden müsste.

3.5 Vergleich der Interpolationsmöglichkeiten

Wie bisher gezeigt wurde, können im Naor-Pinkas-Verfahren die Lagrange-Interpolation und die Newton-Inteprolation gleichermaßen angewendet werden. In diesem Abschnitt sollen diese miteinander verglichen werden. Dabei wird besonders auf die arithmetischen Fließkommaoperationen pro Sekunde ($FLOPS$ = Floating-point Operations per Second) eingegangen, wobei die übliche Gewichtungen der Operationen in der folgenden Tabelle festgehalten werden:

Gleitpunktoperation	**Gewichtung**
Addition, Subtraktion, Multiplikation	1 FLOPS
Division	4 FLOPS
Exponentialfunktion	8 FLOPS

Werden die ermittelten Werte für die Lagrange-Interpolation aus (3.3) genauer betrachtet, ergibt das

$$\begin{aligned}&(t^2+2t)\,M + t\cdot(t+1)\,D + t\cdot(t+1)\,A + (t+2)\,E\\ &= (t^2+2t) + 4t\cdot(t+1) + t\cdot(t+1) + 8(t+2)\\ &= 6t^2+15t+16\end{aligned}$$

und aufgrund von (3.15) erhält die Newton-Interpolation die Gesamtgewichtung

$$\begin{aligned}&\left(\frac{t\cdot(t+1)}{2}\right)M + \left(\frac{t\cdot(t+1)}{2}\right)D + t\cdot(t+1)\,A + (t+1)\,E\\ &= \frac{t\cdot(t+1)}{2} + 4\cdot\left(\frac{t\cdot(t+1)}{2}\right) + t\cdot(t+1) + 8(t+1)\\ &= 3,5t^2+11,5t+8\,.\end{aligned}$$

Das Ziel ist nun, diese zwei Ergebnisse zueinander in Beziehung zu setzen. Dabei wird mit einer Anzahl an unerlaubten Empfängern von $t > 0$ folgende Abschätzung gemacht:

$$6t^2+15t+16 > 6t^2+15t+8 > 6t^2+11,5t+8 > 3,5t^2+11,5t+8$$

Zusammengefasst bedeutet dieses Ergebnis, dass die Anzahl an notwendigen Operationen eines Empfängers bei der Newton-Interpolation im Vergleich zur Lagrange-Interpolation für jede beliebige Anzahl an unerlaubten Empfängern geringer ausfällt. Dadurch kann das Naor-Pinkas-Verfahren mit Hilfe der Newton-Interpolation effizienter ausgeführt werden.

Dass dieses Ergebnis eine nicht zu vernachlässigende Bedeutung hat, soll die folgende Tabelle darstellen.

Unerlaubte Empfänger t	**Gesamtgewichtung Lagrange**	**Gesamtgewichtung Newton**
100	61516	36158
1000	6015016	3511508
10000	600150016	350115008

Ab einer gewissen Anzahl an unerlaubten Empfängern und mit Hilfe der oben definierten üblichen Gewichtung der Operationen wächst der Unterschied der Anzahl an $FLOPS$ zwischen der Lagrange- und der Newton-Interpolation auf das $\frac{6}{3,5}$-fache an.

4. Asmuth-Bloom-Verfahren

In diesem Abschnitt wird ein kurzer Einblick in das Asmuth-Bloom-Verfahren [AB83] gegeben, welches den aus der Zahlentheorie bekannten chinesischen Restsatz anwendet. Hierzu werden jedoch einige Vorkenntnisse benötigt.

4.1 Simultane Kongruenzen ganzer Zahlen

Eine simultane Kongruenz ganzer Zahlen ist ein System von linearen Kongruenzen

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_n \pmod{m_n}, \end{aligned}$$

wofür der unbekannte Wert x für alle Kongruenzen bestimmt werden soll, falls solch ein Wert überhaupt existiert. Wenn eine Lösung x existiert, dann sind $\mathcal{M} \coloneqq kgV(m_1, \ldots, m_n)$ und die Zahlen $x + k\mathcal{M}$ mit $k \in \mathbb{Z}$ alle Lösungen dieses Systems.

4.2 Chinesischer Restsatz

Mit Hilfe des chinesischen Restsatzes kann unter gewissen Einschränkungen eine Aussage über simultane Kongruenzen getroffen werden. Dabei seien $m_1, \ldots, m_n$ paarweise teilerfremde, natürliche Zahlen. Dann existiert für beliebige Zahlen $a_1, \ldots, a_n \in \mathbb{Z}$ eine Zahl x, die das folgende System von linearen Kongruenzen mit $i = 1, \ldots, n$ erfüllt:

$$x \equiv a_i \pmod{m_i}$$

Alle Lösungen sind kongruent modulo $\mathcal{M} \coloneqq m_1 m_2 \cdots m_n$.

Sei $\mathcal{M}_i = \mathcal{M} \setminus m_i$. Das Finden einer Lösung erfolgt mit dem erweiterten euklidischen Algorithmus, wobei zwei Zahlen r_i und s_i berechnet werden, sodass die Gleichung

$$r_i \cdot m_i + s_i \cdot \mathcal{M}_i = 1$$

stimmt. Ferner gilt durch $e_i \coloneqq s_i \cdot M_i$

$$e_i \equiv 1 \pmod{m_i}$$

$$e_i \equiv 0 \pmod{m_j}$$

für $j \neq i$. Die Lösung der simultanen Kongruenz ist definiert durch:

$$x \coloneqq \sum_{i=1}^{n} a_i e_i \,. \tag{4.1}$$

4.3 Asmuth-Bloom-Secret-Sharing-Schema

Im Asmuth-Bloom-Secret-Sharing-Schema werden zwei Phasen benötigt, um das Geheimnis verteilen und rekonstruieren zu können.

- **Initialisierungsphase**
  In dieser Phase wählt der Gruppencontroller eine Menge von paarweisen teilerfremden Zahlen $m_0 < m_1 < \cdots < m_n$, sodass folgende Ungleichung gilt:

  $$\prod_{i=1}^{t} m_i > m_o \prod_{i=1}^{t-1} m_{n-i+1} \tag{4.2}$$

  Ferner werden $\mathcal{M} \coloneqq \prod_{i=1}^{t} m_i$ und das Geheimnis $\mathcal{S}$ definiert. Letzteres wird so gewählt, dass $m_0 > \mathcal{S}$ gilt. Im nächsten Schritt berechnet der Gruppencontroller

  $$y = \mathcal{S} + Am_0 \,,$$

  wobei A eine positive, zufällig ausgewählte Zahl ist, sodass die Bedingung $0 \leq y < \mathcal{M}$ erfüllt wird. Im letzten Schritt sendet der Gruppencontroller jedem Empfänger i das ihm zugewiesenen Share

  $$y_i = y \mod m_i \,.$$

- **Kombinierungsphase**
  Gegeben sei ein Zusammenschluss $\mathcal{N}'$ von $t < n$ Empfängern, welche das Geheimnis in dieser Phase rekonstruieren sollen. Ferner sei noch $M_{\mathcal{N}'} \coloneqq \prod_{i \in \mathcal{N}'} m_i$ und das System von Kongruenzen

  $$y \equiv y_i \pmod{m_i}$$

  für $i \in \mathcal{N}'$ gegeben. Mit Hilfe des chinesischen Restsatzes wird der Wert y aus $\mathbb{Z}_{\mathcal{M}_{\mathcal{N}'}}$ eindeutig bestimmt. Das Geheimnis $\mathcal{S}$ kann im letzten Schritt wie folgt ermittelt werden:

  $$\mathcal{S} = y \mod m_0 \,.$$

  Diese Lösung ist wegen der Bedingung $y < \mathcal{M} \leq \mathcal{M}_{\mathcal{N}'}$ in $\mathbb{Z}_{\mathcal{M}}$ eindeutig.

Das Schema ist sicher, solange eine Vereinigung von maximal $t - 1$ unerlaubten Empfängern sichergestellt wird. Eine detaillierte Analyse der Sicherheit kann in der veröffentlichten Arbeit von Quisquater et al. [QPV02] nachgelesen werden. In dieser wird unter anderem das Asmuth-Bloom- und weitere Secret-Sharing-Schemas basierend auf dem chinesischen Restsatz analysiert.

4.4 Ausblick: Anwendung als Revocation-Schema

Die Anwendung Asmuth-Bloom-Verfahrens als Revocation-Schema analog zum Noar-Pinkas-Schema bringt eine Problematik mit sich. Der für die Berechnung notwendige Wert y_i des Empfängers i kann lediglich vom Gruppencontroller bestimmt werden, da nur ihm der Session-Schlüssel $y = \mathcal{S} + Am_0$ bekannt ist. Im Umkehrschluss bedeutet das, dass der Gruppencontroller vor der Übertragung der Nachricht jedem erlaubten Empfänger seinen Share, beispielsweise über einen privaten Kanal, mitteilen müsste.

Beim Naor-Pinkas-Schema kann jeder erlaubte Empfänger i anhand der Übermittlung von g^r und der gespeicherten Information $P(I_i)$ den erforderlichen $(t+1)$-ten Wert $g^{rP(I_i)}$ für die eindeutige Bestimmung des Polynoms ermitteln. Dies ist im Asmuth-Bloom-Verfahren nicht möglich, da der Wert y_i direkt über den Session-Schlüssel mit Hilfe einer Modulo-Operation berechnet wird.

Kann jedoch das Naor-Pinkas-Schema im Asmuth-Bloom-Verfahren angewendet werden, falls jeder erlaubter Empfänger i seinen persönlichen Share y_i kennt?

Angenommen das Geheimnis $\mathcal{S}$ wird mit Hilfe eines Generators g und einem beliebigen und zufälligen Wert r in den Expontenten verlagert, sodass der neue Schlüssel durch g^{rS} definiert wird. Zudem existieren paarweise, teilerfremde Zahlen $m_0 < m_1 < \cdots < m_n$, wobei $m_0 > \mathcal{S}$ gilt. Aufgrund der letzt genannten Bedingung könnte durch das Verlagern des Schlüssels in den Exponenten das Finden der Werte $m_1, \ldots, m_n$ aufwendiger werden als geplant, da bereits der Startwert m_0 groß sein kann. Ferner berechnet der Gruppencontroller $y = \mathcal{S} + Am_0$. Zuzüglich seien die Ungleichung aus 4.2 und die Bedingung $0 \leq y < \mathcal{M}$ erfüllt. Das gewünschte Ziel ist, die gleiche Übertragung wie beim Naor-Pinkas-Schema zu erreichen:

$$< g^r, g^{r\,y_{u_1}}, g^{r\,y_{u_2}}, \ldots, g^{r\,y_{u_t}}, m_{u_1}, m_{u_2}, \ldots, m_{u_t} > .$$

Hierzu wird die Berechnung eines einzelnen Shares y_i genauer betrachtet:

$$y_i = y \mod m_i$$

Um den eindeutigen Wert y bestimmen zu können, wird ein Kongruenzsystem von t Kongruenzen benötigt:

$$y \equiv y_i \pmod{m_i}\,.$$

Einem Empfänger wird bei der übertragenen Nachricht jedoch nur die in den Exponenten verlagerten Shares übermittelt. Dieser müsste folglich folgende Berechnung

$$y \equiv g^{r\,y_i} \pmod{m_i}$$

durchführen. Dies führt jedoch nicht zum gewünschten Ergebnis. Anders ausgedrückt benötigt jeder Empfänger die für die Berechnung notwendigen Shares der anderen Empfänger in Klartext. Es wurde im Rahmen der Studienarbeit kein Weg gefunden, das Asmuth-Bloom-Verfahren als Revocation-Schema basierend auf dem Prinzip von Naor und Pinkas aufzubauen.

4.5 Fazit

Im Rahmen dieser Studienarbeit wurde kein Weg gefunden, dass Asmuth-Bloom-Verfahren als Revocation-Schema einzusetzen. Zum einen liegt das an den Shares, welche nur beim Gruppencontroller berechnet und übermittelt werden können. Die Unkenntnis des persönlichen Geheimnisses könnte relativ einfach durch Hinzunahme eines weiteren Kommunikationsweges mit dem Empfänger gelöst werden, was jedoch die Effizienz des Verfahrens verringern würde. Zum anderen führen die notwendigen Bedingungen dazu, dass der Aufwand für die Suche nach geeigneten, paarweisen teilerfremden Zahlen, durch das Verlagern des Schlüssels in den Exponenten, steigen.

Das Ziel im Revocation-Schema von Naor und Pinkas hatte den Ursprung, dass der Schlüssel im Shamir's Secret-Sharing-Verfahren nach einmaliger Berechnung im Klartext bestimmt werden konnte. Aufgrund dieser Situation wurde der Schlüssel und die notwendigen Shares $P(I_i)$ in den Exponenten verlagert. Anhand der mathematischen Eigenschaft der Lagrange-Interolation konnte dies geschickt ausgenutzt werden. Dies ist im Asmuth-Bloom-Verfahren nicht nötig, da der Session-Schlüssel bereits durch das Addieren von Am_0 geheim gehalten wird.

Ein weiterer Nachteil des Verfahrens wird beim Finden von paarweise teilerfremde Zahlen bemerkbar, wenn die Anzahl an Teilnehmern groß ist. Die im Abschnitt 4.3 definierten Bedingungen müssen zu jederzeit erfüllt werden. Beim Naor-Pinkas-Verfahren muss der Gruppencontroller lediglich eine Bedingung erfüllen: Eine Stützstelle darf nicht an zwei verschiedenen Teilnehmer vergeben werden. Hierbei fallen bis auf die Suche, ob die Stützstelle bereits vergeben wurde, keine weiteren Kosten an.

Trotzdem stellt das Asmuth-Bloom-Verfahren eine interessante Grundlage für ein praktisches Secret Sharing dar. In der veröffentlichten Arbeit von Kaya und Selçuk [KS07] können weitere interessante Methoden zum Asmuth-Bloom-Verfahren nachgelesen werden. Unter anderem wird in der Arbeit die Verteilung der Shares mit Hilfe der RSA-Funktion im Asmuth-Bloom-Verfahrens näher untersucht.

5. Zusammenfassung und Ausblick

In dieser Studienarbeit wurden die Grundlagen des Shamir's Secret Sharing-Verfahrens und das darauf basierende Naor-Pinkas-Revocation-Schema genauer erläutert. Zudem wurde nach alternativen Interpolationsverfahren gesucht, die im Schema angewendet und effizienter ausgeführt werden konnten. Die zur Lagrange- sehr ähnliche Newton-Interpolation konnte ohne weitere Probleme in das Schema eingebaut werden, wobei der notwendige Schlüsselberechnungsaufwand eines Empfängers zu jedem Zeitpunkt geringer ausfällt als bei der etwas langsameren Interpolation mit Hilfe der Lagrange-Basisfunktionen.

Die approximative Interpolation mit Splines führte zum Ergebnis, dass diese nicht ohne weiteres als Interpolationsmöglichkeit im Naor-Pinkas-Revocation-Schema angewendet werden kann. Bei der Umsetzung entstanden zwei Probleme. Das erste war die Sicherstellung von $x = 0 \in [x_0, x_n]$, was relativ einfach behoben werden konnte, indem ein Dummy-Benutzer $x_d < 0$ hinzugefügt wurde. Eine weitere Lösung ist die Verwendung eines anderen Schlüssels $\mathcal{S} \in [x_0, x_n]$. Das zweite Problem kann jedoch nicht ohne weiteres behoben werden. Es handelt sich hierbei um die Nichteindeutigkeit des berechneten Wertes für den Schlüssel $\mathcal{S}$. Als Lösungsvorschlag wurden eine Schwellenwertfunktion und Polynome von höherem Grad in den Intervallabschnitten beschrieben, welche jedoch die Effizienz und die Sicherheit des Revocations-Schemas reduzieren würden.

Im darauf folgenden Kapitel wurde das Asmuth-Bloom-Secret-Sharing-Schema genauer beschrieben. Das Ziel war, das Naor-Pinkas-Revocation-Schema so einzubauen, dass es als effizientes Broadcasting-Schema angewendet werden kann. In der Analyse fiel auf, dass dies nicht ohne weiteres möglich ist. Grund hierfür sind die vom Gruppencontroller berechneten Shares des Session-Schlüssels. Diese können nicht von einem Empfänger selber berechnet oder hergeleitet werden. Zu einem weiteren Problem führte die Modulo-Eigenschaft. Diese führte aufgrund der Verlagung der Shares in den Exponenten nicht zum gewünschten Ergebnis. Weitere Untersuchungen zu diesem Bereich können in der vorgestellten Arbeit von Kaya und Selçuk [KS07] nachgelesen werden.

Es existieren weitere Secret-Sharing-Verfahren, welche außerhalb des Bereichs der Zahlentheorie und der Interpolationen liegen. Darunter fällt auch die von Blakley [B79] veröffentlichte Arbeit. In diesem Schema werden in einem n-dimensionalen Raum n nichtparallele Hyperebenen definiert. Dadurch wird ein eindeutiger Schnittpunkt im Raum generiert, welcher jedoch nur mit Hilfe der n Hyperebenen eindeutig ermittelt werden kann. Auch hier stellt sich die Frage, ob mit Hilfe des Naor-Pinkas-Revocation-Schemas die notwendigen Informationen in den Exponenten gezogen

werden können, um auf diese Art und Weise ein effizientes Revocation-Schema zu erstellen.

Aufgrund der mathematischen Eigenschaften bieten Secret-Sharing-Verfahren eine gute und effiziente Möglichkeit, den Schlüssel vor unerlaubten Empfängern geheim zu halten. Mit veröffentlichten Arbeiten wie beispielsweise der von Naor und Pinkas werden solche Schemata weiter verbessert und optimiert, um in der heutigen digitalen Welt mehr Sicherheit gewähleisten zu können.

Literaturverzeichnis

[NP00] M. Naor und B. Pinkas: *Efficient Trace and Revoke Schemes.* In *Financial Cryptography - FC 2000*, pages 1-13. Springer-Verlag, 2000.

[FN93] A. Fiat und M. Naor: *Broadcast Encryption.* Advances in Cryptology - CRYPTO '93, Springer-Verlag LNCS vol. 773, 1994, pp. 427-437

[KRS99] R. Kumpar, S. Rajagopalan und A. Sahai: *Coding constructions for blacklisting problems without computational assumptions*. Advances in Cryptology - CRYPTO '99, Springer-Verlag LNCS 1666, pp. 609-623, 1999

[S79] A. Shamir: *How to share a secret*. Comm. ACM, Vol 22, No. 11, pp. 612-613, 1979

[G15] Carl Friedrich Gauss: *Another new proof of the theorem that every integral rational algebraic function of one variable can be resolved into real factors of the first or second degree.* Göttingen 1815.

[DH79] Diffie W. und Hellman M.E., New Directions in Cryptography, *IEEE Trans. on Information Theroy*, pp. 644-654, 1979

[E85] T. ElGamal, *A public key cryptosystem and a sognature scheme based on discrete logarithmus* , Proc. Advances in Cryptology - Grypto '84, Springer-Verlag LNCS 196, pp 10-18, 1985

[NR97] M. Naor und O. Reingold, *Number-Theoretic constructions of efficient pseudo-random functions*, Proc. 38th IEEE Symp. on Foundations of Computer Science, pp. 458-467, 1997

[CS98] R. Cramer und V. Shoup, *A practical public key cryptosystem provably secure against adaptove chosen ciphertext attacks*, Proc. Advances in Cryptology - Crypto '98, Springer-Verlag LNCS 1462, pp. 13-25, 1998

[C11] R. Cramer, *Numerik Skript*, pp. 644-654, 2011

[AB83] C. Asmuth und J. Bloom, *A modular approach to key safeguarding*, IEEE Trans. Information Theory, 208-210, 1983.

[KS07] K. Kaya und A. Selçuk, *Threshold cryptography based on Asmuth-Bloom secret sharing*, Bilkent University, pp. 5-6, 2007

[QPV02] M. Quisquater, B. Preneel und J. Vandewalle. *On the security of the secret sharing scheme based on the chinese remainder theorem* . In Proc. of PKC 2002, Vol. 2274 of LNCS, pp. 199-210. Springer-Verlag, 2002

[F87] P. FELDMAN, *A practival scheme for non-interactive verifiable secret sharing*, Proc. 28th IEEE Symp. on Foundations of Computer Science, pp. 427-437, 1987

[B79] G. R. BLAKLEY. *Safeguarding cryptographic keys* . Proc. of the National Computer Conference 48, pp. 313-317, 1979