Baumbasiertes Dualmodeschlüsselmanagement für die Multicast-Kommunikation

Dissertation

zur Erlangung des Doktorgrads (Dr. rer. nat.)

der Mathematisch-Naturwissenschaftlichen Fakultät

der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Thorsten Aurisch

aus Troisdorf

Bonn 2007

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn.

Erstgutachter: Prof. Dr. Peter Martini, Rheinischen Friedrich-Wilhelms-Universität Bonn

Zweitgutachter: Prof. Dr. Jürgen Grosche, Bergische Universität Wuppertal

Datum der Promotion: 21.12.2007

Erscheinungsjahr: 2008

Kurzfassung

Überall und jederzeit verfügbarer Zugriff auf Informationen und Kommunikation sind von entscheidender Bedeutung im täglichen Leben, in der Wirtschaft und für die nationale Sicherheit. Multimedia-Applikationen wie Audio-/Videokonferenz, Internet-Radio und Video-on-Demand bewirken einen zunehmenden Bedarf an gruppenorientierter Kommunikation. Sie wird von der Internet-Technologie durch den Multicast-Dienst unterstützt und ermöglicht einen effizienten Datenversand an eine Gruppe von Nutzern, die auf verschiedenen Rechnern dislozierte sind.

Die Gewährleistung von Sicherheitsdiensten wie Vertraulichkeit, Authentizität und Integrität für den Datenaustausch ist dabei eine erforderliche Eigenschaft der Gruppenkommunikation, falls diese über offene Rechnernetze betrieben wird. Insbesondere bei einer Gruppenkommunikation auf Multicast-Basis über offene Rechnernetze sind Sicherheitsdienste notwendig, da jeder Nutzer den Empfang von Daten anfordern kann und der Sender dadurch keine Kontrolle über die Zusammensetzung der Gruppe hat. Weiterhin kann jeder Nutzer Daten an eine Gruppe senden.

Die Verfügbarkeit eines gemeinsamen Schlüssels, des so genannten Gruppenschlüssels, bei den Teilnehmern der Gruppenkommunikation bildet die Basis für die Gewährleistung der oben genannten Sicherheitsdienste. Im Rahmen der vorgestellten Arbeit wurde deshalb das grundlegende Problem der Bereitstellung eines Gruppenschlüssels untersucht. Der Schutz der Datenübertragung in einer Gruppe kann z.B. durch den Einsatz der Protokollfamilie IP Security (IPSec) und den bei allen Nutzern verfügbaren Schlüssel gewährleistet werden.

Während die Sicherheit der Punkt-zu-Punkt-Kommunikation ein gut erforschtes Gebiet ist, stellt die sichere Gruppenkommunikation noch zahlreiche Herausforderungen an die Forschung. Sichere Gruppenkommunikation ist keine triviale Erweiterung sicherer Punkt-zu-Punkt-Kommunikation, da sich bei der Gruppenkommunikation auf der Basis von Multicast die Zusammensetzung einer Gruppe durch Gruppenbeitritt bzw. -austritt ändern kann. Ein hierbei eingesetztes Schlüsselmanagement muss deshalb einen dynamischen Gruppenschlüssel verwalten.

Um in diesen Gruppen zu verhindern, dass neue Gruppenteilnehmer bereits durchgeführte Datenübertragungen und ehemalige Teilnehmer zukünftige Datenübertragungen mitlesen können, muss das Schlüsselmanagement den Gruppenschlüssel wechseln, wenn die Anzahl der Teilnehmer der Gruppe sich verändert. Diese Forderung wird als Schlüsselgeheimhaltung bezeichnet. Für den Wechsel des Gruppenschlüssels muss ein Schlüsselmanagement effiziente schnelle Mechanismen bereitstellen. Diese sind notwendig, da während des Schlüsselwechsels keine Nutzdaten ausgetauscht werden können.

Soll ein Datenübertragungsschutz für Multicast auch in den Streitkräften ermöglicht werden, muss das für die Gruppenschlüsselbereitstellung verantwortliche Schlüsselmanagementsystem zusätzliche Forderungen erfüllen. Ein derartiges Schlüsselverwaltungssystem muss eine Schlüsselbereitstellung sowohl in Dynamic Peer Groups (DPG) als auch in großen Gruppen ermöglichen. Dynamic Peer Groups sind kleine Gruppen, in denen der Gruppenkoordinator aus Sicht der Kommunikationstechnik nicht von vornherein fixiert ist und die oftmals über

funkbasierte infrastrukturlose Netzwerke, so genannte mobile Ad hoc Netzwerke (MANET), kommunizieren. Allerdings können sich Dynamic Peer Groups durch den Beitritt einer Vielzahl von Nutzern in große Gruppen umwandeln. Deshalb ist es notwendig, die als Skalierbarkeit bezeichnete Forderung zu erfüllen. In den bisherigen Entwicklungen von Schlüsselmanagementverfahren wurde dieser Aspekt allerdings nicht betrachtet. Weiterhin muss die Verlässlichkeit der Schlüsselbereitstellung gewährleistet sein. Da ein Einsatz des Schlüsselmanagements auch in MANETs möglich sein soll, sind zum einen durch eine robuste Schlüsselübermittlung temporäre Kommunikationsfehler, z.B. Paketverluste, zu kompensieren. Zum anderen ist ein eine Reparierbarkeit bei Prozessausfällen und Verbindungsverlusten sicherzustellen. Oftmals ist es erforderlich, im Einsatz befindliche Streitkräfte umzugliedern. Ein Schlüsselmanagement muss deshalb auch bei einer starken Teilnehmerfluktuation die erforderlichen Schlüssel bereitstellen. Weiterhin muss eine Schlüsselübermittlung auch an Teilnehmer mit temporärer Simplexverbindung möglich sein. Eine derartige Situation ist die Folge des Betriebs der Kommunikationsmittel im Zustand Emission Control, in dem Daten empfangen, aber nicht versendet werden dürfen. Ein derartiger Zustand der Kommunikationsmittel kann erforderlich sein, wenn funkbasierte Kommunikationsmittel eingesetzt werden und ein Schutz vor Aufklärungsmaßnahmen gewährleistet werden soll.

Im Rahmen der Arbeit wurde zur effizienten Verwaltung des Gruppenschlüssels speziell für den Einsatz in den Streitkräften das Konzept Multicast Internet Key Exchange (MIKE) eingeführt. Das Konzept beinhaltet zwei Schlüsselbereitstellungsverfahren. Die beiden Verfahren sind im Konzept MIKE als Betriebsmodus Key Agreement und Key Distribution bezeichnet. Grundlage beider Verfahren ist die Datenstruktur bzw. Methode Schlüsselbaum, die einen effizienten Schlüsselwechsel ermöglicht. Deshalb kann MIKE auch als baumbasiertes Dualmodeschlüsselmanagement bezeichnet werden. Ein Vorteil von zwei Verfahren in einem Schlüsselmanagement ist die Skalierbarkeit des Konzeptes bei steigender Gruppengröße. Im Modus Key Agreement wird einem Nutzer zur Koordination der Schlüsselbereitstellung dynamisch der Status des Gruppenkoordinators zugewiesen. Weil auf den Einsatz eines zentralen Prozesses zur Schlüsselbereitstellung verzichtet wird, zeichnet sich dieser Modus durch Reparierbarkeit im Fehlerfall aus. Der Modus Key Distribution wird zur Schlüsselbereitstellung eingesetzt, wenn die Leistungsfähigkeit des anderen Betriebsmodus an seine Grenzen stößt. Das Konzept MIKE enthält außerdem einen Mechanismus, mit dem zwischen den Schlüsselbereitstellungsverfahren umgeschaltet werden kann, ohne die Gruppen erneut zu initialisieren.

Mit den szenariospezifischen Optimierungen wurde eine weitere Anpassung des Schlüsselmanagements MIKE an die militärische Verwendung vollzogen. Hierzu wird die Sammelverarbeitung von Teilnehmeranfragen entworfen, um einen Schlüsselwechselmechanismus für eine starke Teilnehmerfluktuation bereit zu stellen. Die nutzerverhaltensbasierte Schlüsselbaumkonstruktion berücksichtigt, dass militärische Gruppen aus Teilgruppen mit bekanntem Verhaltensmuster bestehen, und baut daher den Schlüsselbaum entsprechend auf. Um den Betrieb des Schlüsselmanagements auch im Zustand Emission Control zu ermöglichen, wurde für den Betriebsmodus Key Agreement die ressourcengesteuerte Auswahl des Gruppenkoordinators entwickelt.

Zur Verifikation der Tragfähigkeit des Konzepts MIKE wurde dieses mittels einer experimentellen Implementierung evaluiert. Die durchgeführte Effizienzanalyse ergab, dass das als Betriebsmodus Key Agreement entworfene Schlüsselvereinbarungsverfahren insbesondere in Netzwerken mit limitierter Datenübertragungskapazität eine schnellere Schlüsselbereitstellung als bestehende Konzepte ermöglicht. Weiterhin konnte die Skalierbarkeit des Konzepts nachgewiesen werden, indem gezeigt wurde, dass durch die Verfügbarkeit eines zweiten Betriebsmodus, auch in Netzwerken mit sehr geringer Datenübertragungskapazität und auch bei steigender Gruppengröße eine Schlüsselbereitstellung möglich ist.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Überall und jederzeit verfügbarer Zugriff auf Informationen und Kommunikation sind von entscheidender Bedeutung im täglichen Leben, in der Wirtschaft und für die nationale Sicherheit. Multimedia-Applikationen wie Audio-/Videokonferenz, Internet-Radio und Video-on-Demand bewirken einen zunehmenden Bedarf an gruppenorientierter Kommunikation. Der Multicast-Dienst der Internet-Technologie ermöglicht einen effizienten Datenversand an eine Gruppe. Die Gewährleistung von Sicherheitsdiensten wie Vertraulichkeit, Authentizität und Integrität für ausgetauschte Informationen sind dabei notwendige Eigenschaften der Gruppenkommunikation, um die Akzeptanz bei den Anwendern zu erreichen. Ursache der Forderung nach Sicherheitsdiensten ist das anonyme Empfängermodell des Multicast-Diensts. Hiermit wird zum Ausdruck gebracht, dass jeder Nutzer Daten anfordern kann und der Sender dadurch keine Kontrolle über die Zusammensetzung der Gruppe hat. Weiterhin kann jeder Nutzer Daten an eine Gruppe senden. Während die Sicherheit der Punkt-zu-Punkt-Kommunikation ein gut erforschtes Gebiet ist, stellt die sichere Gruppenkommunikation noch zahlreiche Herausforderungen an die Forschung [Can00]. Sichere Gruppenkommunikation ist keine triviale Erweiterung sicherer Punkt-zu-Punkt-Kommunikation. An dieser Stelle werden einige zusätzliche Herausforderungen der Gruppenkommunikation erläutert. Eine Punkt-zu-Punkt-Kommunikation hat einen definierten Startzeitpunkt und endet nach einer bestimmten Dauer. Gruppenkommunikation ist diesbezüglich komplexer. Eine Gruppe wird gebildet und kann sich durch Eintritt, Austritt oder Ausschluss von Teilnehmern ändern. Es gibt nicht notwendigerweise ein fest definiertes Ende. Diese Dynamik macht die Garantie von Sicherheitseigenschaften wesentlich aufwändiger. Die zahlreichen und sehr unterschiedlichen Einsatzgebiete bewirken eine sehr unterschiedliche Dynamik und Größe der Gruppenkommunikation. So haben beispielsweise Gruppen, die für Video-on-Demand gebildet werden, andere Eigenschaften als spontan gebildete Gruppen in Ad-hoc-Netzwerken. Eine weitere Herausforderung liegt in der Kommunikationskomplexität der Protokolle, die abhängig von der Teilnehmeranzahl ist.

Die Verfügbarkeit eines gemeinsamen Schlüssels, des so genannten Gruppenschlüssels bei den Teilnehmern der Gruppenkommunikation bildet die Basis für die Gewährleistung von Sicherheitsdiensten. Im Rahmen dieser Arbeit wird deshalb das grundlegende Problem der Bereitstellung eines Gruppenschlüssels untersucht. Eine dynamische Bereitstellung ist auf Grund der sich verändernden Gruppengröße und Zusammensetzung notwendig.

Die bisherigen Entwicklungen von Schlüsselmanagementsystemen konzentrieren sich auf die Bereitstellung eines Gruppenschlüssels entweder in sehr großen Gruppen (Large Group, LG) oder auf Dynamic Peer Groups (DPG) nicht jedoch beides. DPGs sind kleine Gruppen, in denen der Gruppenkoordinator nicht von vornherein fixiert ist, d.h. es gibt keinen zentralen Prozess, der mehr Möglichkeiten als andere Gruppenteilnehmer hat, und die häufig über mobile Ad hoc Netzwerke (MANET) verbunden sind. Im Gegensatz dazu existiert in großen Gruppen, d.h. in Gruppen mit mehr als 1000 Teilnehmern, meistens ein zentraler Prozess. Soll ein Schlüsselmanagement in den Streitkräften eingesetzt werden, so gelangt man bei der Analyse von Kommunikationsprofilen zu der Erkenntnis, dass beide Arten von Gruppen existieren und aus DPGs durch Teilnehmerzuwachs große Gruppen werden. Die Motivation

der Arbeit besteht darin, die dynamische Bereitstellung eines gemeinsamen Schlüssels in beiden Gruppentypen zu ermöglichen. Weiterhin sollen spezielle aus dem militärischen Einsatzkontext resultierende Forderungen an ein Schlüsselmanagement, z.B. Schlüsselbereitstellung für Teilnehmer mit Simplexkommunikation, erfüllt werden.

Nachfolgende Gliederung soll den „roten Faden“ der Arbeit vermitteln.

Für die Definition von Sicherheitsmechanismen in Gruppen ist es zunächst notwendig, sich allgemein mit Gruppenkommunikation zu beschäftigen. Dies umfasst eine Beschreibung der Eigenschaften von Gruppen sowie der Realisierung von Gruppenkommunikation in paketorientierter Datenübertragung basierend auf der Internet-Technologie. Zur Gewährleistung von Sicherheitsdiensten in Gruppen sind zahlreiche Herausforderungen zu bewältigen. Das Kapitel 2 enthält einen Überblick über diese Herausforderungen und stellt die Entwicklung eines Schlüsselmanagements als zentrale Forschungsaufgabe bei der Gewährleistung von Sicherheitsdiensten in Gruppen dar. Die hierfür benötigten grundlegenden kryptographischen Methoden werden eingeführt und eine Notation dafür definiert.

Das Kapitel 3 widmet sich der Einordnung dieser Arbeit in die Forschung auf dem Gebiet des Gruppenschlüsselmanagements. Als Einstieg in das Thema wird ein Schlüsselmanagement der Punkt-zu-Punkt-Kommunikation rekapituliert. An diesem einfachen Fall wird der Einsatz von Schlüsselbäumen erläutert. Anschließend wird eine Definition für Schlüsselbäume eingeführt, die unabhängig von Schlüsselbereitstellungsverfahren ist. Weiterhin werden existierende Schlüsselmanagementverfahren vorgestellt und im Hinblick auf zuvor definierte Anforderungen bewertet. Diese Anforderungen setzen sich zusammen aus den grundlegenden Anforderungen an ein Schlüsselmanagementsystem, sowie aus denen, die aus dem militärischen Verwendungszweck resultieren.

In Kapitel 4 wird das im Rahmen dieser Arbeit neu entwickelte Schlüsselmanagement Multicast Internet Key Exchange (MIKE) präsentiert. Dieses umfasst eine Beschreibung der Idee des Schlüsselmanagements, der eingesetzten Algorithmen und eine Definition der verwendeten Nachrichten. Idee des Konzeptes ist die Realisierung eines Schlüsselmanagements mit zwei Schlüsselbereitstellungsverfahren, die als Key Agreement und Key Distribution bezeichnet werden. Beide Verfahren basieren auf dem im vorhergehenden Kapitel definierten Schlüsselbaum. Deshalb kann MIKE auch als baumbasiertes Dualmodeschlüsselmanagement bezeichnet werden. Weiterhin werden in diesem Kapitel die Mechanismen zur Erhöhung der Fehlertoleranz von MIKE erläutert.

Um die Tragfähigkeit des Konzeptes zu verifizieren, wurde dieses implementiert. Die Beschreibung der Implementierung ist Bestandteil von Kapitel 5. Zur Unterstützung des Verständnisses der komplexen Abläufe in einem Schlüsselmanagement wird auch eine graphische Darstellung der Schlüsselbereitstellungsmechanismen realisiert.

In Kapitel 6 werden die Methoden zur Leistungsbewertung eines Gruppenschlüsselmanagements festgelegt. Schwerpunkt der Bewertung ist die Analyse der Effizienz des Schlüsselwechsels. Die hierfür notwendigen Metriken werden festgelegt und Messgrößen im Schlüsselmanagement MIKE identifiziert.

Eine Leistungsverbesserung für das Schlüsselmanagement MIKE hinsichtlich der benötigten Rechenleistung und Datenübertragungskapazität kann erzielt werden, wenn es an die militärische Verwendung angepasst wird. Hierzu wird in Kapitel 7 die Sammelverarbeitung von Nutzeranfragen und nutzerverhaltenbasierte Schlüsselbaumkonstruktion eingeführt und deren Effizienzsteigerung untersucht.

Mittels der definierten Metrik wird in Kapitel 8 eine Leistungsbewertung des Schlüsselmanagements MIKE durchgeführt. Die Bewertung der Effizienz erfolgt durch Simulation und Messungen im praktischen Einsatz. Das verwendete Messverfahren und die Messarchitektur werden vorgestellt. Eine theoretische Leistungsbewertung untermauert die praktischen Resultate.

Zusammengefasst werden alle Ideen und Untersuchungen in Kapitel 9. Ein Ausblick beschließt das Kapitel.

Abgeschlossen wird die Arbeit durch ein Verzeichnis auftretender Abkürzungen sowie einer Liste der verwendeten Literatur.

2 Sicherheitsdienste bei der Gruppenkommunikation

In diesem Kapitel wird zunächst allgemein die Gruppenkommunikation erläutert, bevor anschließend Sicherheitsdienste in Gruppen definiert werden.

2.1 Gruppenkommunikation

Als Gruppenkommunikation bezeichnet man eine Kommunikationsform, bei der mehr als zwei Nutzer miteinander kommunizieren. Typische Beispiele für Gruppenkommunikation sind Konferenzen und Taxifunk. Betrachtet man die Veränderungen einer Gruppe, so kann man zwischen statischen und dynamischen Gruppen unterscheiden. In der Literatur werden derartige Gruppen oftmals als geschlossene und offene Gruppen bezeichnet. Bei statischen Gruppen ist die Zusammensetzung der Gruppe vorgegeben und ändert sich im Kommunikationszeitraum nicht. Dynamische Gruppen hingegen unterliegen einer Veränderung in der Gruppengröße und Zusammensetzung. Die Ursache dieser Veränderung sind so genannte Teilnehmeroperationen, die auf Grund einer Anfrage zum Gruppenbeitritt bzw. -austritt durchgeführt werden. Änderungen der Gruppengröße und Zusammensetzung können durch einzelne und multiple Teilnehmeroperationen hervorgerufen werden. Nachfolgende Aufzählung gibt einen Überblick über mögliche Teilnehmeroperationen (Abbildung 1:):

- Einzelne Teilnehmeroperationen
  - JOIN: Beitritt eines Teilnehmers zu einer Gruppe.
  - LEAVE: Austritt eines Teilnehmers aus einer Gruppe.
  - EJECT: Ausschluss eines Teilnehmers von einer Gruppe und Verweigerung eines erneuten Gruppenbeitritts.
- Multiple Teilnehmeroperation
  - MULTIPLE JOIN: Gleichzeitiger Beitritt mehrerer einzelner Teilnehmer zu einer Gruppe.
  - MERGE: Der Beitritt einer bereits organisierten Teilgruppe zu einer bestehenden Gruppe.
  - MULTIPLE LEAVE: Gleichzeitiger Austritt einzelner Teilnehmer aus einer Gruppe.
  - PARTITION: Austritt einer Teilgruppe aus einer bestehenden Gruppe.

Hinsichtlich des Kommunikationsmusters wird in Gruppen zwischen der 1-zu-m und n-zu-m Kommunikation unterschieden. Hierbei bezeichnen n und m die Anzahl der beteiligten Nutzer. Die weiteren im Rahmen dieser Arbeit durchgeführten Betrachtungen konzentrieren sich auf die n-zu-m Gruppenkommunikation in Rechnernetzwerken mit dynamischer Teilnehmerzusammensetzung. Die 1-zu-m Kommunikation ist in der n-zu-m Gruppenkommunikation enthalten und kann als ein Spezialfall angesehen werden. Für die an einer Gruppenkommunikation teilnehmenden räumlich verteilten aktiven Elemente wird zusätzlich zu der Bezeichnung Nutzer häufig die Bezeichnung Prozess verwendet. Dieser Terminologie wird sich auch in dieser Arbeit bedient.

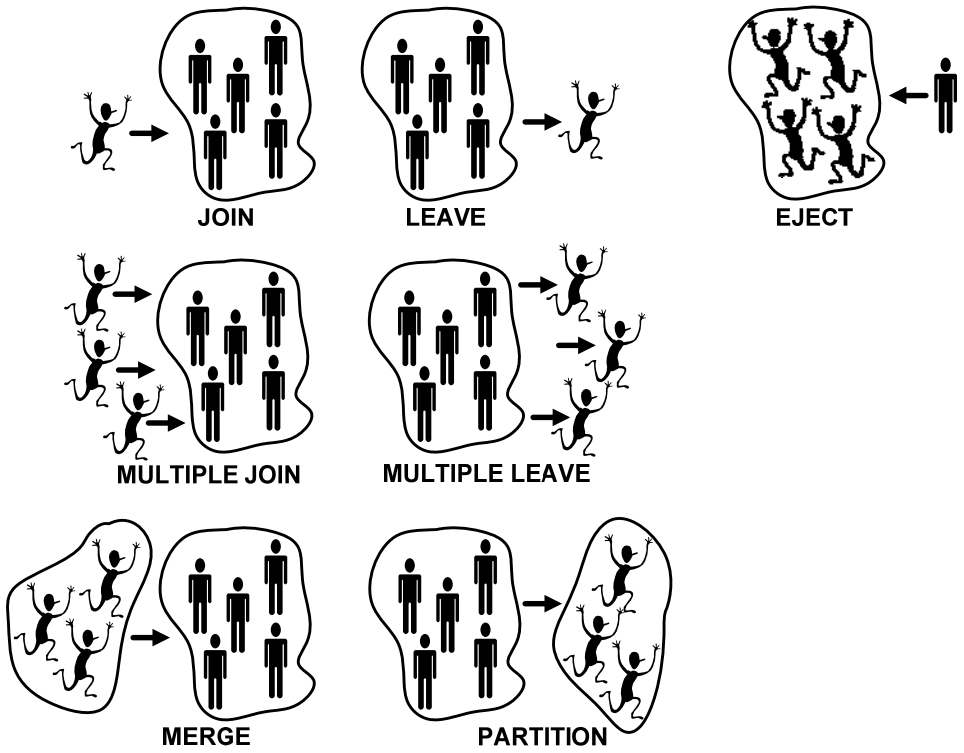

Abbildung 1: Teilnehmeroperationen in dynamischen Gruppen

2.2 Gruppenkommunikation in Rechnernetzwerken

In Netzwerken mit paketorientierter Datenübertragung, die auf der Internet-Technologie basieren, kann Gruppenkommunikation effizient durch IP-Multicast realisiert werden [Mie01]. Meistens wird hierfür nur die Bezeichnung Multicast verwendet. Obwohl Multicast eine Technologie zur Realisierung von Gruppenkommunikation bezeichnet, wird dieser Begriff synonym für Gruppenkommunikation verwendet.

2.2.1 Adressierung beim Multicast

In der Internet-Technologie wird der Sender und Empfänger eines Datenpakets jeweils durch eine IP-Adresse festgelegt. Multicast beinhaltet, dass alle Teilnehmer einer Gruppe zu einer so genannten Multicast-Gruppe zusammengefasst und durch eine spezielle IP-Adresse repräsentiert werden. Anstatt die Pakete mit den Daten an jedes Mitglied der Gruppe einzeln zu senden, werden diese an die Multicast-Adresse geschickt. Die Router des Netzwerkes vervielfältigen und verteilen die Datenpakete dann an die Teilnehmer der Multicast-Gruppe. Mittels dieses Verfahrens wird eine Minimierung der übertragenen Pakete erzielt und damit die benötigte Datenübertragungsrate vermindert. Zusätzlich wird eine Verringerung der Belastung des Senders erreicht, da nur eine Übertragung für alle Empfänger notwendig ist. Multicast ist neben der Punkt-zu-Punkt-Kommunikation, auch IP-Unicast bzw. Unicast genannt, integraler Bestandteil des in der Internet-Technologie eingesetzten Netzwerkprotokolls Internet Protocol Version 4 (IPv4) sowie des als Nachfolger spezifizierten Internet Protocol Version 6 (IPv6). In IPv4 ist für die Adressierung von Multicast-Gruppen der Adressbereich von 224.0.0.0 bis 239.255.255.255 reserviert [Dee89].

Adressen dieses Bereichs werden als Klasse D bezeichnet. Multicast-Adressen für IPv6 erkennt man daran, dass diese mit FF beginnen. In IPv6 steht der Adressbereich von FF01:0:0:0:0:0:0:0 bis FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF zur Verfügung [Hin03].

2.2.2 Teilnehmerverwaltung einer Multicast-Gruppe

Zur Verwaltung von Teilnehmern einer Multicast-Gruppe wird von Routern in IPv4 das Internet Group Management Protocol (IGMP) eingesetzt [Fen97]. Dieses ermöglicht es einem Prozess, den Empfang von Datenpaketen einer Multicast-Gruppe bei einem Router anzufordern und somit einer Multicast-Gruppe beizutreten. Mittels der Nachricht `Host Membership Query` wird durch den Router periodisch überprüft, ob es in seinem Zuständigkeitsbereich Empfänger für Multicast-Pakete gibt. Alle an Multicast Interessierte antworten auf diese Nachricht nach einer zufälligen Verzögerung mit einem `Host Membership Report`. Um überflüssige Mehrfachantworten zu vermeiden, wird ein `Host Membership Report` nur dann abgeschickt, wenn im lokalen Subnetz noch keiner geantwortet hat (Abbildung 2).

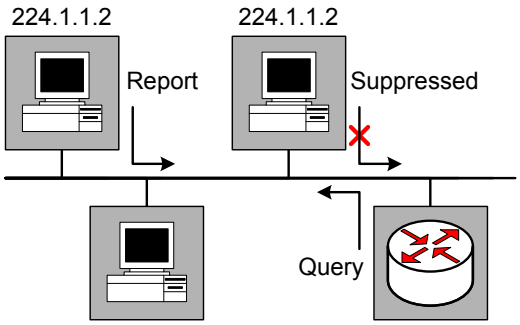

Abbildung 2: IGMP zur Verwaltung von Multicast-Gruppen

Es gibt drei Versionen von IGMP mit den folgenden prinzipiellen Eigenschaften:

- IGMP Version 1: Ein Prozess kann nur einer Multicast-Gruppe beitreten. Ein Abmelden ist bei dieser Version nicht vorgesehen. Erhält ein Router auf die Nachricht `Host Membership Query` keine Antwort, so werden nach dem Überschreiten einer Zeitschranke keine Pakete mehr weitergeleitet.
- IGMP Version 2: Ein Prozess kann sich bei dieser Version auch von der Multicast-Gruppe abmelden. Hierzu wird die Nachricht `Leave Membership Query` verwendet.
- IGMP Version 3: Mit dieser Version ist es möglich, das Interesse an Multicast-Daten von einer vorgegebenen Quelle abzufragen.

Im Gegensatz zum IGMP von IPv4 sind bei IPv6 die Nachrichten zum Verwalten von Multicast-Gruppen kein eigenes Protokoll, sondern in das Internet Control Message Protocol Version 6 (ICMPv6) eingebettet [Con98]. Dieses ICMPv6-Teilprotokoll wird als Multicast Listener Discovery (MLD) [Dee99] bezeichnet und definiert die Nachrichten `Multicast Listener Query`, `Multicast Listener Report` und `Multicast Listener Done`. Ein Router prüft durch ein `Multicast Listener Query` regelmäßig, ob es in seinem Zuständigkeitsbereich Empfänger für Multicast-Pakete gibt. Soll nur eine bestimmte

Multicast-Gruppe geprüft werden, so wird die Anfrage nur an deren Multicast-Adresse geschickt. Soll jedoch nach allen Gruppen gefragt werden, so wird die Nachricht an die Link-Local-Multicast-Adresse geschickt. Interessierte Prozesse antworten auf die Nachricht mit einem `Multicast Listener Report`. Mit Hilfe der Nachricht `Multicast Listener Done` kann ein Prozess aus einer Multicast-Gruppe austreten. Auch vom ICMPv6-Teilprotokoll MLD existieren mehrere Versionen. Die MLD Version 1 ist ähnlich zu IGMP Version 2, während MLD Version 2 vergleichbare Funktionalität wie IGMP Version 3 aufweist. Die beiden Protokolle IGMP und MLD ermöglichen somit eine dynamische Zusammensetzung der Multicast-Gruppen.

2.2.3 Routing-Protokolle zur Verteilung von Multicast-Paketen

Zur Verteilung von Multicast-Paketen werden Multicast-Routing-Protokolle benötigt. Das Ziel dieser Protokolle besteht darin, Multicast-Verteilbäume (Multicast Distribution Tree) zu ermitteln. Ein solcher Verteilbaum enthält alle Router mit den angeschlossenen Rechnern, die Mitglied der Multicast-Gruppe sind. Hierzu wurde eine Vielzahl von Verfahren entworfen. In drahtgebundenen Netzwerken lassen sich diese danach klassifizieren, ob ein quellenbasierter Verteilbaum (Source-rooted Multicast Distribution Tree) oder ein gemeinsamer Verteilbaum (Shared Multicast Distribution Tree) für eine Multicast-Gruppe gebildet wird. Nachfolgend werden die etablierten Multicast-Routing-Protokolle gemäß der genannten Klassifizierung aufgelistet:

- Das Distance Vector Multicast Routing Protocol (DVMRP) [Wai88], das Multicast Open Shortest Path First (MOSPF) [Moy94] und das Protocol Independent Multicast (PIM) Dense Mode [Ada05] bilden einen Multicast-Verteilbaum für jeden Sender von Multicast-Paketen. Die Pakete werden also abhängig vom Sender an die Teilnehmer der Gruppe weitergeleitet.
- Das Protocol Independent Multicast (PIM) Sparse Mode [Est98], das Core Based Tree Protocol (CBT) [Bal01] [Bal02], das Ordered Core Based Tree Protocol (OCBT) [Shi96] und das Hierarchical Multicast Routing (HIP) [Shi98] erzeugten für alle Sender den gleichen Verteilbaum. Die Routing-Protokolle CBT, OCBT, und HIP erzeugen bidirektionale gemeinsame Verteilbäume. Die Auswertung des Baums zur Paketverteilung kann an jedem Punkt des Baums gestartet werden. Hingegen verwendet PIM Sparse Mode einen gemeinsamen unidirektionalen Verteilbaum. Multicast-Pakete werden zuerst zum zentralen Element des Routing-Verfahrens, dem so genannten Rendezvous Point (RP), gesandt und dann unter Auswertung eines gemeinsamen unidirektionalen Verteilbaums an alle Teilnehmer übermittelt.

Einige der Multicast-Routing-Protokolle für drahtgebundene Netzwerke werden im nachfolgenden Abschnitt detaillierter vorgestellt.

Das DVMRP basiert auf dem Verfahren Reverse Path Multicast (RPM). Beim RPM überträgt ein Router ein Multicast-Paket von einer bestimmten Quelle über alle seine Verbindungen außer der Verbindung, über die das Paket empfangen wurde. Eine Übertragung wird aber nur dann durchgeführt, wenn das Paket auf der Verbindung angekommen ist, die auf dem kürzesten Pfad zur Quelle liegt. Ansonsten verwirft der Router das Paket. Beim RPM

ermittelt ein Router den kürzesten Pfad zur Quelle, indem er die Unicast-Routing-Tabellen auswertet. Ein Router, der Multicast-Pakete empfängt, obwohl kein angeschlossener Rechner zu der betreffenden Multicast-Gruppe gehört, sendet eine Prune-Nachricht an seinen benachbarten Router. Die Unterdrückung des Empfangs unerwünschter Pakete durch Router wird als Pruning bezeichnet. Dieses kann durch den Versand einer so genannten Graft-Nachricht explizit rückgängig gemacht werden.

Mit MOSPF wird das Routing-Protokoll Open Shortest Path First (OSPF) für Multicast erweitert. Im speziellen wird festgelegt, wie Router ihre Multicast-Gruppenmitgliedschaft in Link-State-Advertisements einfügen können, mit denen OSPF das Netzwerk flutet. Somit verfügen alle Router neben den für Unicast benötigten Topologieinformationen über die Teilnehmer der verschiedenen Multicast-Gruppen, und können dadurch quellenspezifische Verteilbäume für jede Multicast-Gruppe berechnen. Mit MOSPF wird deshalb Link State Multicast (LSM) realisiert.

Das Protocol Independent Multicast funktioniert ohne Routing-Informationen des Unicast-Dienstes. PIM verfügt über zwei Betriebsarten, die als Dense Mode und Sparse Mode bezeichnet werden. Effizientes Multicast-Routing im Dense Mode ist dann möglich, wenn die Multicast-Gruppenteilnehmer dicht beieinander liegen, d.h. die meisten Router in das Multicast-Routing einbezogen werden. Im Sparse Mode hingegen ist die Anzahl der am Multicast beteiligten Router im Vergleich zur Gesamtzahl gering, und die Gruppenteilnehmer sind großflächig im Netzwerk verteilt. PIM Dense Mode verwendet einen Ansatz wie Reverse Path Multicasting, d.h. die Router fluten zunächst das Rechnernetzwerk. Im Sparse Mode wird eine Multicast-Gruppe vom Rendezvous Point verwaltet. Die Sender von Multicast-Paketen registrieren ihre Multicast-Gruppe am Rendezvous Point. Bekunden Rechner mittels IGMP bzw. MLD Interesse an einer Multicast-Gruppe, so sendet der für diese Rechner zuständige Router Join-Nachrichten an den Rendezvous Point. Dieser sendet daraufhin seinerseits eine Join-Nachricht an den Router, der den Sender verwaltet, und erhält die Multicast-Pakete der betreffenden Multicast-Gruppe. Der Rendezvous Point leitet nun alle erhaltenen Multicast-Pakete an den Empfänger weiter.

Bei Multicast-Routing-Protokollen für funkbasierte, infrastrukturlose Netzwerke, so genannte mobile Ad hoc Netzwerke (MANET), unterscheidet man zwischen positions- und topologiebasierten Routing-Verfahren. Positionsbasierte Routing-Verfahren, z.B. das Scalable Position-Based Multicast (SPBM) [Tra04], nutzen Informationen über die genauen Positionen zum Aufbau der Multicast-Verteilbäume. Die weit verbreiteten topologiebasierten Routing-Verfahren kommen ohne genaue Informationen über die Positionen aus. Ihnen genügen Informationen über die Nachbarschaftsbeziehungen der Rechner, also welche Rechner eine direkte Verbindung haben. Üblicherweise wird topologiebasiertes Routing-Verfahren für mobile Ad-hoc-Netzwerke in pro-aktive und reaktive Verfahren unterteilt:

- Pro-aktive Routing-Verfahren
  Pro-aktive Verfahren bestimmen die zu verwendenden Verteilbäume, bevor diese tatsächlich benötigt werden. Beispiel für Protokolle dieser Klasse sind das Multicast Optimized Link State Routing (MOLSR) [Jac01] sowie das Core-Assisted Mesh Protocol (CAMP) [Gar99].

- Reaktive Routing-Verfahren
  Im Gegensatz zu den pro-aktiven Verfahren bestimmen reaktive Routing-Verfahren die benötigten Verteilbäume erst, wenn diese tatsächlich benötigt werden. Daraus ergibt sich, dass das erste Datenpaket einer Verbindung erst mit einer geringen Verzögerung versendet werden kann. Das Protokoll Multicast Ad hoc On-Demand Distance Vector Routing (MAODV)or [Roy99] und das Protokoll On-Demand Associativity-Based Multicast (ABAM) [Cha00] sind Beispiele für Protokolle dieser Kategorie.

Eine Kombination aus pro-aktiven und reaktiven Routing-Verfahren wird als hybrides Routing bezeichnet.

2.2.4 Gruppenkommunikationssysteme

Gruppenkommunikationssysteme (Group Communication System, GCS) sind verteilte Nachrichtenübermittlungssysteme, die eine zuverlässige Kommunikation zwischen einer Gruppe von Prozessen $\mathcal{P}=\{p_1,\ldots,p_j\}$ ermöglichen. Hierzu stellt das Gruppenkommunikationssystem die nachfolgenden zwei Dienste bereit:

- Gruppenmitgliedschaftsdienst
- Zuverlässigen, geordneten Multicast-Dienst

Der Gruppenmitgliedschaftsdienst administriert eine Liste der aktiven und verbundenen Prozesse. Die Ausgabe des Gruppenmitgliedschaftsdiensts wird Gruppensicht (group view) genannt. Die Gruppensicht kann sich durch den Gruppenbeitritt bzw. Gruppenaustritt ändern. Ungewollte Änderungen der Gruppensicht werden durch Prozessabsturz oder Verbindungsabbruch hervorgerufen. Der zuverlässige, geordnete Multicast-Dienst liefert Multicast-Nachrichten mit dem Ordnungsgrad FIFO (First-In-First-Out), kausal und total an die Teilnehmer der aktuellen Gruppensicht aus. Der Nachrichtenaustausch von Anwendungsprozessen über Gruppenkommunikationssysteme ist in Abbildung 3 dargestellt.

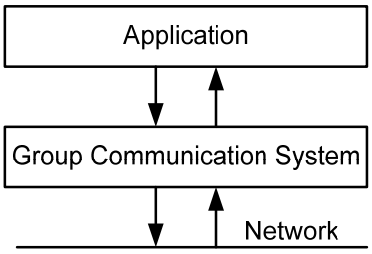

Abbildung 3: Nutzung eines Gruppenkommunikationssystems durch einen Anwendungsprozess

Formale Spezifikationen von Gruppenmitgliedschaftsdienst und zuverlässigem, geordnetem Multicast-Dienst sind Virtual Synchrony (VS) [Fek01] und Extended Virtual Synchrony (EVS) [Mos94]. Die Spezifikationen verknüpfen Nachrichtenübermittlung und Gruppensicht. Die beiden Spezifikationen werden in verkürzter Form rekapituliert. Hierbei wird, wie bereits in Abschnitt 2.1 eingeführt, für die aktiven Elemente bzw. Nutzer der Begriff Prozess verwendet. Zur Definition der Gruppensemantik wird eine Gruppenkommunikation bestehend aus einer Menge von Prozessen $\mathcal{P}=\{p_1,\ldots,p_j\}$, ausgetauschten Nachrichten $\mathcal{M}=\{m_1,\ldots,m_k\}$ und Gruppensichten $\mathcal{I}=\{Id_1,\ldots,Id_\ell\}$ betrachtet. Diese Gruppensichtbezeichnungen können

durch die Relation „>“ geordnet werden. Die Semantik Virtual Synchrony besitzt nachfolgende Eigenschaften:

- Self Inclusion
  Legt ein Prozess $p_i \in \mathcal{P}$ die Gruppensicht $Id_k \in \mathcal{I}$ an, dann ist p_i Teilnehmer von Id_k.
- Local Monotonicity
  Wenn ein Prozess $p_i \in \mathcal{P}$ eine Gruppensicht $Id_k \in \mathcal{I}$ nach der Gruppensicht $Id_j \in \mathcal{VID}$ anlegt, dann gilt $Id_k > Id_j$.
- Initial View
  Ein Nachrichtenaustausch kann nur in einer Gruppensicht $Id_k \in \mathcal{VID}$ stattfinden. Es beginnen alle Prozesse mit einer Anfangsgruppensicht.
- Partitionable Membership
  Teilt sich ein Gruppenkommunikationssystem mit den Prozessen $\mathcal{P}$ in zwei Gruppen mit den Prozessen $\mathcal{P}'$ und $\mathcal{P}''$, dann sind die Gruppensichtbezeichnungen $\mathcal{I}'$ und $\mathcal{I}''$ unabhängig von $\mathcal{I}$.
- Message Delivery Integrity
  Empfängt ein Prozess $p_i \in \mathcal{P}$ die Nachricht $m_\ell \in \mathcal{M}$ in einer Gruppensicht $Id_k \in \mathcal{I}$, dann existiert ein Prozess q_j, der die Nachricht m_l vorher in einer Gruppensicht Id_k gesendet hat.
- No Duplication
  Eine Nachricht $m_\ell \in \mathcal{M}$ wird nur einmal an den Prozess $p_i \in \mathcal{P}$ ausgeliefert.
- Sending View Delivery
  Eine Nachricht wird in der Gruppensicht $Id_k \in \mathcal{I}$ ausgeliefert, in der diese gesendet wird.
- FIFO Delivery
  Wird vom selben Prozess $p_i \in \mathcal{P}$ eine Nachricht $m_\ell \in \mathcal{M}$ vor der Nachricht $m_n \in \mathcal{M}$ in der gleichen Gruppensicht $Id_k \in \mathcal{I}$ gesendet, dann wird von jedem Prozess $p_i \in \mathcal{P}$ m_ℓ vor m_n empfangen.
- Causal Delivery
  Werden die Nachrichten $m_l, m_n \in \mathcal{M}$ in der gleichen Gruppensicht $Id_k \in \mathcal{I}$ gesendet und hängt die Nachricht m_n kausal von der Nachricht m_ℓ ab, dann wird von jedem Prozess $p_i \in \mathcal{P}$ m_ℓ vor m_n empfangen.
- Total Ordered Delivery
  Wird in der gleichen Gruppensicht $Id_k \in \mathcal{I}$ die Nachricht $m_\ell \in \mathcal{M}$ vor der Nachricht $m_n \in \mathcal{M}$ gesendet, dann wird von jedem Prozess $p_i \in \mathcal{P}$ m_ℓ vor m_n empfangen.
- Safety Notification
  Ist die Nachricht $m_\ell \in \mathcal{M}$ des Prozesses $p_i \in \mathcal{P}$ an alle Prozesse $p_j \in \mathcal{P}$ mit $j \neq i$ ausgeliefert worden, dann erhält der Prozess p_i darüber eine Bestätigung.

Im Unterschied dazu enthält die Spezifikation Extended Virtual Synchrony nicht die Eigenschaft Sending View Delivery. Diese wird ersetzt durch:

- Same View Delivery
  Eine Nachricht $m_\ell \in \mathcal{M}$ des Prozesses $p_i \in \mathcal{P}$ an alle Prozesse $p_j \in \mathcal{P}$ mit $j \neq i$ ausgeliefert in der gleichen Gruppensicht $Id_k \in \mathcal{I}$.

Systeme wie Isis [Bir94], Transis [Ami92], Horus [Ren96], Spread [Sta98] und Totem [Aga94] stellen eine zuverlässige Kommunikation in einer Gruppe entsprechend der Spezifikation Virtual Synchrony bzw. Extended Virtual Synchrony bereit. Das Gruppenkommunikationssystem Totem ist ein Vertreter eines Systems, das die Spezifikation Virtual Synchrony erfüllt. Dessen Funktionsweise soll im Folgenden exemplarisch erläutert werden. Die Idee von Totem basiert auf einem Quittungsmechanismus mittels eines Tokens. Dabei wird ein Token in fester Reihenfolge an alle Gruppenteilnehmer weitergegeben, so dass diese einen logischen Ring bilden. Die Reihenfolge wird auf der Basis der eindeutigen Bezeichnung für jeden Totem-Prozess realisiert. Dieses Protokoll wird als Totem-Single-Ring-Protokoll (Totem-SR-Protokoll) und das Token als Regular-Token bezeichnet (vgl. Abbildung 4). Zum Senden muss ein Totem-Prozess der Gruppe im Besitz des Tokens sein. Das Regular-Token enthält dazu die Nachrichtensequenznummer der nächsten total angeordneten Nachricht. Nach dem Versand der Nachricht wird das Token mit erhöhter Nachrichtensequenznummer weitergereicht. Zur zuverlässigen Auslieferung werden keine speziellen Quittungen benötigt, da jedes Gruppenmitglied das Token innerhalb einer Umkreisung bekommt. Jeder ist dann aufgrund der im Token enthaltenen Nachrichtensequenznummer in der Lage, einen eventuellen Nachrichtenverlust festzustellen. In diesem Fall wird eine Anforderung der betreffenden Nachrichten an das Token gehängt. Wenn ein Sender einer Nachricht das Token bekommt und darin eine Anforderung für eine von ihm versandte Nachricht findet, wird diese an die entsprechenden Nutzer gesendet. Im Gegensatz zu negativen Quittungsmechanismen ist bei diesem Verfahren keine dauerhafte Speicherung aller Nachrichten notwendig. Wenn nach einer Umkreisung des Tokens keine Anforderung zur Wiederholung im Token enthalten ist, kann daraus die zuverlässige Zustellung an alle Prozesse gefolgert werden.

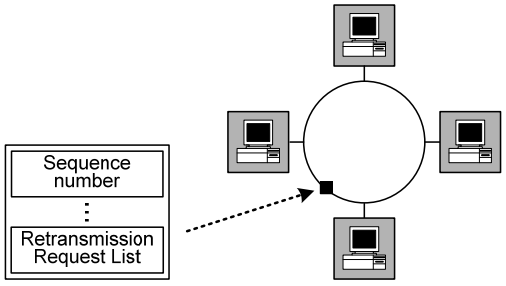

Abbildung 4: Totem-Single-Ring-Protokoll

Zur Aufnahme eines neuen Prozesses in das Totem-SR-Protokoll sowie bei einem Verlust des Regular-Token wird das Totem-Membership-Protokoll (Totem-M-Protokoll) verwendet. Es dient zur Einigung über die neue Ringmitgliedschaft. Zum Erzielen eines Konsenses hierüber wird von jedem Prozess eine Join-Nachricht mit seiner aktuellen Teilnehmerliste so oft versandt, bis die Teilnehmerliste bei empfangener und versandter Join-Nachricht identisch ist. Bei U Teilnehmern werden mindestens $2 \cdot U$ Join-Nachrichten übertragen. Durch Zirkulation des Commit-Tokens bestätigen alle Prozesse die Teilnehmerliste. Insgesamt werden zwei Umläufe des Commit-Tokens durchgeführt. Der zweite Umlauf wird dazu verwendet, noch

nicht an alle Prozesse ausgelieferte Nachrichten zu ermitteln. Diese Information nutzt der im Anschluss verwendete Recovery-Mechanimus.

2.3 Grundlegende Sicherheitsdienste der Kryptographie

Rechnernetzwerke sind eine Ressource, die von vielen Anwendern für verschiedene Zwecke gemeinsam genutzt wird. Um eine gesicherte Kommunikation über ein derartiges offenes Kommunikationsmittel zu ermöglichen, ist es erforderlich, die ausgetauschten Daten zu schützen. Bevor auf die besonderen Herausforderungen zum Schutz von Multicast eingegangen wird, werden in diesem Abschnitt Terminologie und Notation grundlegender Sicherheitsdienste erläutert. Weiterhin werden die zum Schutz von Daten benötigten kryptographischen Methoden eingeführt. Die nachfolgenden Sicherheitsdienste werden betrachtet:

- Vertraulichkeit (Confidentiality)
  Das Schutzziel besteht in der Geheimhaltung von Daten, um ein Abhören durch unberechtigte Nutzer zu verhindern. Durch die Verteilung des zum Schutz verwendeten Schlüssels an autorisierte Nutzer kann somit eine Zugriffskontrolle realisiert werden.
- Integrität (Integrity)
  Schutzziel ist die Erkennung einer Fälschung bzw. Unversehrtheit von Dateninhalten.
- Authentizität (Authenticity)
  Eng verwandt mit dem Schutzziel Integrität ist das Schutzziel Authentizität, d.h. die Gewährleistung der Herkunft von Daten. Bei authentischen Daten lässt sich genau erkennen, von wem diese gesendet oder erzeugt wurden. Ist die Authentizität von Daten gewährleistet, kann der Absender die Erzeugung der Informationen nicht leugnen.

Ein asymmetrisches Verschlüsselungsverfahren ist eine mögliche Basis zur Realisierung des Sicherheitsdienstes Vertraulichkeit. Hierzu wird von einem Nutzer u_2 ein Schlüsselpaar bestehend aus öffentlichem Schlüssel pk und geheimem Schlüssel sk erzeugt (Abbildung 5). Das Schlüsselpaar besitzt die Eigenschaft, dass der geheimen Schlüssel sk nicht aus dem öffentliche Schlüssel pk abgeleitet werden kann. Das bedeutet, es ist nicht effizient möglich, den geheimen Schlüssel aus dem öffentlichen Schlüssel zu ermitteln. Der Nutzer u_1 kann eine Nachricht m bei der Übertragung schützen, indem er die mit dem öffentlichen Schlüssel pk verschlüsselte Nachricht cm=E(m,pk) sendet. Der vom Nutzer u_1 verwendete Schlüssel wurde zu diesem Zweck vom Nutzer u_2 im Vorfeld der Kommunikation übermittelt. Die hierzu verwendete Verbindung kann ungesichert sein. Zum Entschlüsseln der Nachricht muss der Nutzer u_2 dann den geheimen Schlüssel sk verwenden. Die bei diesem Verfahren verwendeten Schlüssel werden als asymmetrische Schlüssel bezeichnet. Der Algorithmus RSA ist ein Beispiel für einen asymmetrischen Verschlüsselungsalgorithmus [Riv78]. RSA ist nach seinen Erfindern R. L. Rivest, A. Shamir und L. Adleman benannt.

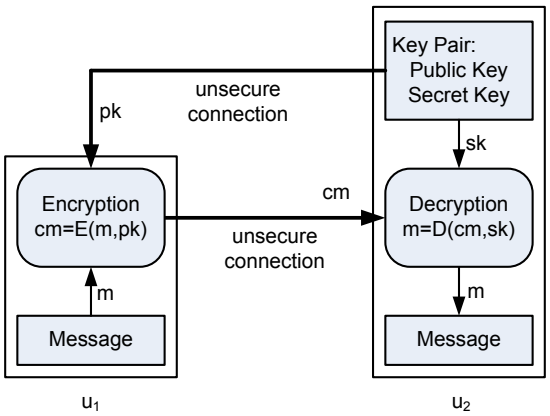

Abbildung 5: Asymmetrische Verschlüsselung

Der Sicherheitsdienst Vertraulichkeit basierend auf einem symmetrischen Verschlüsselungsalgorithmus ist in Abbildung 6 dargestellt. Der Nutzer u_1 verschlüsselt hierzu seine Nachricht m mit dem Schlüssel k, d.h. dieser berechnet cm=E(m,k). Die Nachricht cm kann dann über die ungeschützte Verbindung übermittelt werden. Damit der Empfänger die Nachricht interpretieren kann, entschlüsselt er diese mit demselben Schlüssel k, indem er m=D(cm,k) berechnet. Voraussetzung für dieses Verfahren ist, dass der zum Schutz verwendete Schlüssel k über eine geschützte Verbindung zwischen den Nutzern übermittelt wurde (Abbildung 6). Die bei diesem Verfahren verwendeten Schlüssel werden als symmetrische Schlüssel bezeichnet. Ein Beispiel ist der Algorithmus Advanced Encryption Standard (AES), der in der symmetrischen Kryptographie häufig eingesetzt wird [Nis01].

Zur Erzeugung eines Schlüssels werden meistens Pseudozufallszahlengeneratoren (Pseudo Random Number Generator) verwendet. Sie erzeugen eine Zahlenfolge, die zwar zufällig aussieht, es aber nicht wirklich ist, da sie durch einen deterministischen Algorithmus berechnet wird. Bei jedem Start der Berechnung einer Zufallszahl x wird dem Generator ein Startwert s übergeben, d.h. x=Prng(s). Der Startwert für einen Pseudozufallszahlengenerator wird auch als Seed bezeichnet.

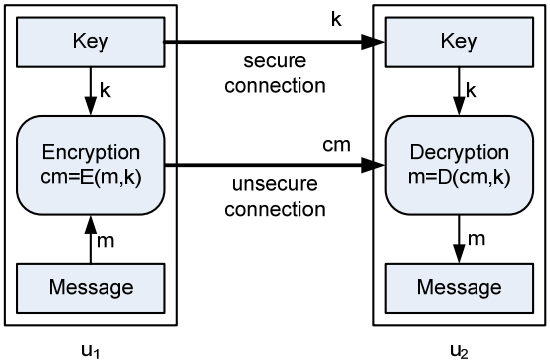

Abbildung 6: Symmetrische Verschlüsselung

Für die Realisierung der Sicherheitsdienste Authentizität und Integrität wird eine zusätzliche Information, die so genannte digitale Signatur, zur Nachricht hinzugefügt. Zu deren Berechnung werden Prüfsummenfunktionen benötigt. Eine Prüfsummenfunktion ist eine Funktion h=Hash(m), die eine Nachricht m auf eine Zeichenkette h mit fester Länge abbildet.

Damit die Funktion einsetzbar ist, muss sie kollisionsfrei sein. Dies bedeutet, es ist nicht effizient möglich, zwei Nachrichten mit derselben Prüfsumme zu finden. Eine weitere Anforderung an die Prüfsummenfunktionen ist die Unumkehrbarkeit, d.h. es darf keine effizient berechenbare inverse Funktion geben, mit der es möglich wäre, für eine gegebene Prüfsumme eine passende Nachricht zu finden. Aus diesem Grund werden Prüfsummenfunktionen oft auch als Einwegfunktionen bezeichnet. Beispiele für Prüfsummenfunktionen sind Message Digest Algorithm 5 (MD5) [Riv92] und Secure Hash Algorithm 1 (SHA-1) [Nis95]. Zur Realisierung digitaler Signaturen können asymmetrische und symmetrische Schlüssel eingesetzt werden. Werden ein asymmetrischer Schlüssel pk_1, sk_1 und ein asymmetrischer Verschlüsselungsalgorithmus verwendet, so berechnet der Absender u_1 der Nachricht m die digitale Signatur s, in dem er die Prüfsumme h=Hash(m) mit dem Schlüssel sk_1 verschlüsselt (Abbildung 7). Die digitale Signatur wird zusammen mit der Nachricht dem Empfänger übermittelt. Der Empfänger u_2 überprüft die digitale Signatur, in dem er die Prüfsumme der erhaltenen Nachricht h′=Hash(m) berechnet. Anschließend vergleicht er die erhaltene, mit dem öffentlichen Schlüssel pk_1 entschlüsselte Signatur, d.h. $D(s,pk_1)$ mit der berechneten Prüfsumme h′. Zeigt das Ergebnis des Vergleichs eine Übereinstimmung der beiden Werte, so wurde die Unversehrtheit der Nachricht nachgewiesen. Mit diesem Verfahren kann Authentizität gewährleistet werden, wenn das Schlüsselpaar pk_1, sk_1 eindeutig dem Nutzer u_1 zugeordnet ist.

Für diese Zuordnung werden digitale Zertifikate verwendet, welche außerdem die Authentizität eines Schlüsselpaares und seinen zulässigen Anwendungs- und Geltungsbereich bestätigen. Ein digitales Zertifikat ist selbst durch eine digitale Signatur geschützt. Für dessen Prüfung wird jedoch wiederum eine Zuordnung des öffentlichen Schlüssels des Zertifikatsausstellers zu seiner Identität, d.h. ein weiteres Zertifikat, benötigt. Auf diese Weise lässt sich eine Kette von digitalen Zertifikaten konstruieren. Auf die Echtheit des letzten Zertifikates muss man sich allerdings ohne ein weiteres Zertifikat verlassen. Eine Hierarchie von Zertifikaten bildet eine Public Key Infrastruktur (PKI) [Smi98]. Bei einer PKI wird der Aussteller eines Zertifikates als Zertifizierungsinstanz bezeichnet. Diese hat die Möglichkeit, durch die Herausgabe einer Zertifikatswiderrufsliste ein Zertifikat für ungültig zu erklären.

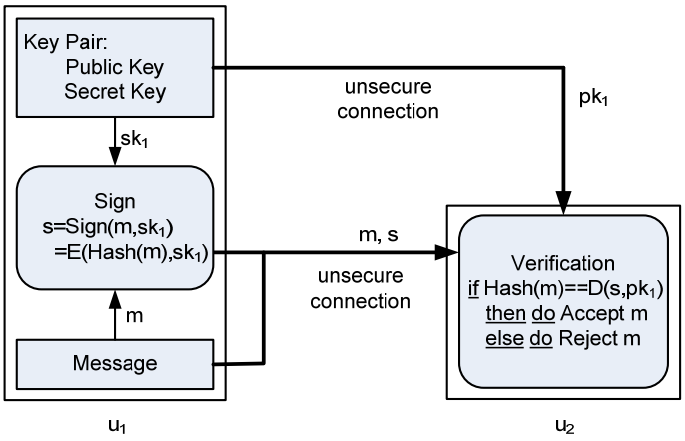

Abbildung 7: Digitale Signatur mit einem asymmetrischen Schlüssel

Eine digitale Signatur kann auch unter Zuhilfenahme eines symmetrischen Schlüssels berechnet werden. Derartige Signaturen werden zur Unterscheidung von Signaturen basierend

auf asymmetrischen Schlüsseln oft als kryptographischer Message Authentication Code (MAC) bezeichnet. Voraussetzung ist hierbei wieder, dass der verwendete Schlüssel k im Vorfeld der Kommunikation den beiden Nutzern u_1 und u_2 bekannt ist. Zur Signaturerzeugung berechnet der Absender der Nachricht die Prüfsumme h über die zu übermittelnde Nachricht m und den nur u_1 und u_2 bekannten Schlüssel k, d.h. h=Mac(m,k)=Hash(m||k) (Abbildung 8). Hierbei wurde die Notation m||k verwendet, um die Nachricht m mit dem Schlüssel k zu verketten. Die digitale Signatur wird auch bei diesem Verfahren zusammen mit der Nachricht dem Empfänger übermittelt. Der Empfänger u_2 überprüft die digitale Signatur, indem er ebenfalls h′=Mac(m,k) über die erhaltene Nachricht m und dem ihm bekannten Schlüssel k berechnet. Anschließend vergleicht er die erhaltene Signatur mit der berechneten digitalen Signatur. Stimmen die beiden Werte überein, so wird die Nachricht als unverändert akzeptiert. Mit diesem Verfahren kann Authentizität gewährleistet werden, wenn der Schlüssel k eindeutig dem Nutzer u_1 zugeordnet und außer u_2 niemandem bekannt ist.

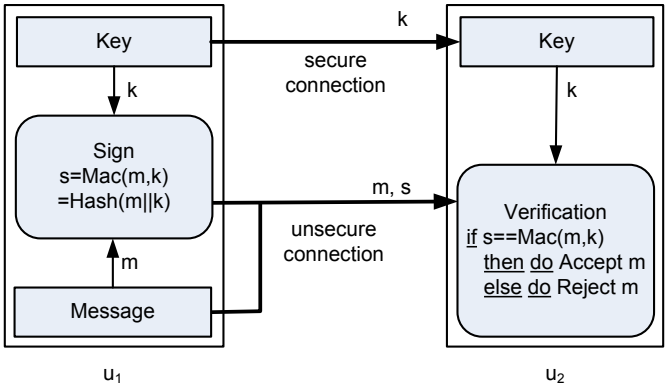

Abbildung 8: Digitale Signatur (kryptographischer MAC) mit einem symmetrischen Schlüssel

Die bei der Beschreibung grundlegender Sicherheitsdienste in dieser Arbeit verwendeten Notationen sind in Tabelle 1 zusammengefasst.

u_i	Nutzer (user) i
m	Nachricht (message)
cm	Verschlüsselte Nachricht (ciphered message)
k	Symmetrischer Schlüssel k (key)
pk	Asymmetrischer öffentlicher Schlüssel pk (public key)
sk	Asymmetrischer geheimer Schlüssel sk (secret key)
$m_1\|\|m_2$	Verkettung der Nachricht m_1 und m_2
E(m,k)	Verschlüsselung (Encryption) der Nachricht m mit Schlüssel k
D(cm,k)	Entschlüsselung (Decryption) der verschlüsselten Nachricht cm mit dem Schlüssel k
Hash(m)	Prüfsumme (hash) der Nachricht m
Prng(s)	Zufallszahlerberechnung mit einem Pseudozufallszahlengeneratoren und dem Seed s
Sig(m,sk)	Signatur der Nachricht m mit dem geheimen Schlüssel sk
Mac(m,k)	Kryptographischer Message Authentication Code der Nachricht m mit dem symmetrischen Schlüssel k

Tabelle 1: Notationen für die Beschreibung grundlegender Sicherheitsdienste

2.4 Multicast-Sicherheit

Multicast ermöglicht eine ressourcen-schonende Realisierung der Gruppenkommunikation in Rechnernetzen. Allerdings ist die Bereitstellung von Sicherheitsmechanismen für Multicast notwendig, um die Sicherheitsanfälligkeiten von Multicast zu beseitigen und so die Akzeptanz bei den Anwendern zu erreichen. Eine Ursache für Sicherheitsanfälligkeiten ist das anonyme Empfängermodell von Multicast. Hiermit wird bezeichnet, dass ein Sender keine Kontrolle über die Zusammensetzung der Gruppe hat, da jeder Nutzer den Datenempfang anfordern kann. Weiterhin wird bei Multicast das Vervielfältigen der Daten von nicht unter dem Einfluss des Senders stehenden Netzwerkkomponenten übernommen. Durch die beiden Eigenschaften ist ein Abhören bzw. Verfälschen von Multicast-Daten leicht möglich. Zusätzlich zur Anforderung des Datenempfangs kann jeder Nutzer Daten an eine Gruppe senden, ohne Teilnehmer der Gruppe zu sein. Durch den unberechtigten Versand bzw. Empfang von Multicast-Daten ist es möglich, dass Netzwerkressourcen verschwendet werden. Ist dies so erheblich, dass die Verfügbarkeit von multicast-basierter Datenübertragung nicht mehr gewährleistet werden kann, liegt ein Denial of Service (DoS) vor. Das wichtige Teilgebiet Multicast-Sicherheit des Forschungsfelds Gruppenkommunikation beinhaltet deshalb die Bereitstellung von Sicherheitsmechanismen für Multicast.

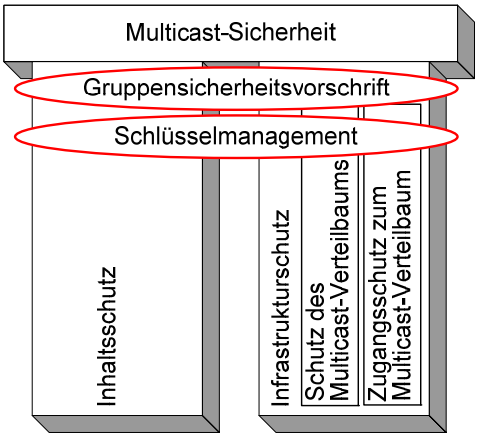

Abbildung 9: Säulen der Multicast-Sicherheit

Es besitzt zwei Säulen (Abbildung 9):

- Multicast-Infrastrukturschutz
  Durch den Multicast-Infrastrukturschutz wird die sichere Vervielfältigung und Verteilung der Multicast-Daten durch die Router des Netzwerkes gewährleistet. Außerdem wird sichergestellt, dass nur berechtigte Nutzer an eine Multicast-Gruppe senden bzw. den Datenempfang anfordern können.
- Multicast-Inhaltsschutz
  Die Aufgabe dieses Themenbereiches besteht darin, eine Anwendung der in Abschnitt 2.3 erläuterten Sicherheitsdienste auch auf Multicast-Datenverkehr zu ermöglichen. Durch den Multicast-Inhaltsschutz wird ein Abhören bzw. Verfälschen verhindert.

Wie in Abbildung 9 verdeutlicht, müssen Lösungen für den Multicast-Infrastrukturschutz und den Multicast-Inhaltsschutz gefunden werden, um ein minimales Maß an Multicast-Sicherheit zu gewährleisten. Die fundamentalen Themenbereiche Gruppensicherheitsvorschrift und Schlüsselmanagement sind in beiden Säulen der Multicast-Sicherheit zu betrachten. In den nachfolgenden Abschnitten werden diese beiden Säulen der Multicast-Sicherheit detaillierter erläutert und der Inhalt aktueller Forschungsarbeiten dargelegt.

2.5 Multicast-Inhaltsschutz

Als Einstieg in den Themenbereich Multicast-Inhaltsschutz wird das Multicast-Sicherheit-Referenzframework der Internet Engineering Task Force (IETF) vorgestellt (Abbildung 10) [Har00]. Das Referenzframework beschreibt die zum Schutz der Multicast-Daten benötigten Elemente und Interaktionen. Die Interaktionen zwischen den Elementen des Frameworks werden durch Pfeile dargestellt. Es wurde entworfen, um die komplexen Herausforderungen, die mit dem Schutz von Multicast-Daten verbunden sind, zu veranschaulichen. Die nachfolgenden drei Elemente sind Bestandteile des Frameworks:

- Gruppenverwalter/Schlüssel-Server
  Aufgabe des Gruppenverwalters/Schlüssel-Servers (Group Controller/Key Server, GCKS) ist die Verwaltung und Verteilung der in der Multicast-Gruppe verwendeten Schlüssel. Um sicherzustellen, dass nur berechtigte Nutzer an der Gruppenkommunikation teilnehmen, werden in der Gruppe eingesetzte Schlüssel in Abhängigkeit von Nutzerauthentisierung und Berechtigungsprüfungen verteilt. Die in der Gruppe geltende Gruppensicherheitsvorschrift erhält der GCKS von dem Sicherheitsvorschrifts-Server.
- Sender, Empfänger
  Die Sender und Empfänger (Sender, Receiver) tauschen Nutzdaten mittels Multicast aus. Außerdem interagieren diese im Rahmen der Schlüsselverwaltung mit dem GCKS. Diese Interaktion beinhaltet Authentisierung sowie den Bezug und falls erforderlich die Erneuerung des Schlüsselmaterials bzw. der Sicherheitsparameter.
- Sicherheitsvorschrifts-Server
  Der Sicherheitsvorschrifts-Server (Policy Server) definiert und verwaltet die Gruppensicherheitsvorschrift. Der Sicherheitsvorschrifts-Server interagiert mit GCKS, um die Sicherheitsvorschrift für die Berechtigung zur Gruppenteilnahme, für die Schlüsselverwaltung und den Schutz der Nutzdaten zu etablieren bzw. zu verwalten.

Eine Schlüsselbereitstellung kann entweder zentral nur durch einen GCKS erfolgen oder dezentral realisiert werden. Im Multicast-Sicherheit-Referenzframework werden beide Ansätze berücksichtigt. Im Fall einer zentralen Schlüsselbereitstellung besteht das Referenzframework nur aus dem linken Teil der Abbildung 10. Da bei dezentralen Schlüsselbereitstellungsverfahren mehrere GCKS die Gruppenschlüsselbereitstellung übernehmen, ist in diesem Fall der rechte Teil der Abbildung 10 zu ergänzen. Beim verteilten Ansatz müssen die GCKS untereinander Informationen austauschen. Die Elemente des Referenzframeworks und deren Interaktionen werden zu drei funktionalen Bereichen zusammengefasst. Nachfolgend werden diese Bereiche detaillierter erläutert. In den Bereichen 1 und 3 des Multicast-Sicherheit-Referenzframeworks wird zusätzlich der Inhalt aktueller

Forschungsarbeiten dargelegt. Schwerpunkt bei der Darstellung des funktionalen Bereichs 2 des Referenzframeworks bildet die Darstellung der Ziele und Aufgaben eines Schlüsselmanagements. Eine Beschreibung und detaillierte Analyse existierender Schlüsselmanagementverfahren erfolgt im nächsten Kapitel, da dieser Themenkomplex Gegenstand der Arbeit ist.

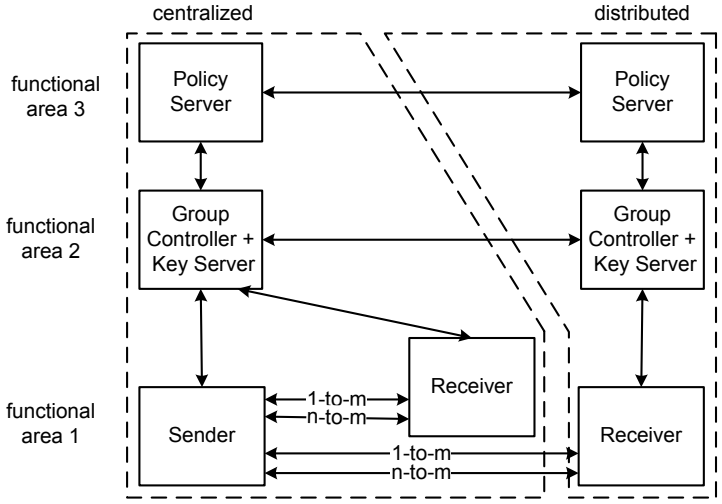

Abbildung 10: Multicast-Sicherheit-Referenzframework [Har00] für den Multicast-Inhaltsschutz

2.5.1 Funktionaler Bereich 1: Sicherung der Nutzdaten

Der Sicherheitsdienst Vertraulichkeit kann durch die Verschlüsselung der übertragenen Daten erzielt werden. Wird zusätzlich der Gruppenschlüssel nur an autorisierte Nutzer verteilt, kann eine Zugriffskontrolle gewährleistet werden. Die Gewährleistung der Authentizität und Integrität von Informationen ist in Gruppen komplexer. Dieser Sicherheitsdienst besitzt zwei Formen:

- Quellenauthentizität und Integrität
  Diese Authentizitäts- bzw. Integritätsform hat den Nachweis zum Ziel, dass die ausgetauschten Daten von der angegebenen Quelle bzw. Sender stammen und weder von einem anderen Gruppenmitglied noch einem externen Angreifer modifiziert wurden.
- Gruppenauthentizität und Integrität
  Dieser Typ der Authentizität bzw. Integrität hat zum Ziel sicherzustellen, dass ausgetauschte Daten von einem Mitglied der Gruppe stammen und nicht durch einen externen Nutzer verändert wurden.

Zum Schutz der Nutzdaten bei Multicast eignen sich Sicherheitsprotokolle der Netzwerk- oder Anwendungsebene des Open Systems Interconnection Models (OSI-Modell), z.B. IP Security (IPSec) oder Secure Real-time Transport Protocol (SRTP) [Nas04]. IPSec hat sich bereits beim Schutz der Punkt-zu-Punkt-Kommunikation bewährt und liegt deshalb im Fokus der Untersuchungen. Nachfolgend sind die Vorteile der Protokollfamilie zusammengefasst:

- Die Sicherheitsdienste sind unabhängig von Anwendungen und Übertragungsmedien.
- Die Sicherheitsdienste werden für eine logische Verbindung festgelegt und nicht pro Übertragungsstrecke.

- Der Schutz der Nutzdaten erfolgt im Hintergrund und erfordert kein Eingreifen durch den Nutzer.
- Parameter und Schlüssel für den Schutz von Nutzdaten werden über dasselbe Kommunikationsmittel übertragen wie die zu schützenden Daten (Inband-Management). Vorteil dieser Vorgehensweise ist, dass keine zweite Infrastruktur zur Schlüsselbereitstellung zur Verfügung gestellt werden muss.

Zum Schutz der übertragenen Daten dienen bei IPSec die Sicherheitsprotokolle Authentication Header (AH) und Encapsulating Security Payload (ESP) [Ken98] [Atk98]. Beide können sowohl separat als auch in Kombination verwendet werden. Durch die Verwendung der IPSec-Sicherheitsprotokolle in Kombination mit einem kryptographischen Message Authentication Code (vgl. Abschnitt 2.3), und den im Request for Comments (RFC) 4305 [Eas05] festgelegten Algorithmen kann aber nur das Schutzziel Gruppenauthentizität und -integrität erreicht werden. Zur Gewährleistung von Quellenauthentizität und -integrität können digitale Signaturen (vgl. Abschnitt 2.3) basierend auf asymmetrischen Algorithmen, wie in [Wei06] spezifiziert, eingesetzt werden. Die derzeitige Spezifikation der Sicherheitsprotokolle AH und ESP bietet nur die Möglichkeit, entweder Gruppen- oder Quellenauthentizität zu gewährleisten. Eine Möglichkeit, beide Schutzziele zu erreichen, bietet der Einsatz des Protokolls Multicast Encapsulating Security Payload (MESP) [Bau03]. Zusätzlich bietet MESP noch Schutz gegen Angriffe durch wiederholtes Senden.

Durch den Berechnungsaufwand einer digitalen Signatur ist deren Verwendung zur Gewährleistung von Quellenauthentizität und Quellenintegrität für einige Datenübertragungen nicht geeignet. Aus diesem Grund wurden effizienter berechenbare Quellenauthentisierungsverfahren vorgeschlagen. Diese können in die Klassen MAC-basierte und verkettungs-basierte Quellenauthentisierungsverfahren unterteilt werden. Beispiel für eine MAC-basierte Quellenauthentisierung ist das Timed Efficient Stream Loss-tolerent Authentication (TESLA) [Per01]. Grundlage von TESLA ist die so genannte Schlüsselkette (key chain). Eine derartige Kette ist nur in einer Richtung berechenbar, d.h. aus einem Kettenelement bzw. Schlüssel lassen sich alle nachfolgenden Schlüssel rekursiv berechnen. Ist man im Besitz des ersten Schlüssels, ist die ganze Kette rekonstruierbar. Der zweite Schlüssel ermöglicht die Rekonstruktion aller Schlüssel der Kette bis auf den ersten Schlüssel. Aus dem vorletzten Schlüssel kann nur der Letzte berechnet werden. Aus diesem ist kein weiterer Schlüssel der Kette zu folgern. In Abbildung 11 ist dargestellt, wie bei TESLA eine Schlüsselkette zur Authentisierung eingesetzt wird. Im Zeitintervall Δt_i wird die Nachricht m_i mittels des symmetrischen Schlüssels k_i authentisiert, indem das Paket p_i zusätzlich zur Nachricht m_i mit dem kryptographischen Message Authentication Code $Mac(m_i,k_i)$ versehen wird. Gleichzeitig wird der nur im Zeitintervall Δt_{i-1} gültige Schlüssel k_{i-1} veröffentlicht, indem dieser ebenfalls dem Paket hinzugefügt wird. Ein Empfänger, der das Paket p_{i-1} bzw. die Nachricht m_{i-1} gepuffert hat, kann deren Quellenauthentizität und Quellenintegrität mittels k_{i-1} verifizieren. Damit dieses Verfahren funktioniert, muss eine lose Zeitsynchronisation zwischen den beteiligten Prozessen bestehen. Hierzu kann zum Beispiel das in [Rei94] beschriebene Verfahren eingesetzt werden. Eine starke Zeitsynchronisation, z.B. mittels des Network Time Protocols (NTP) [Mil92], mit hohem Berechnungs- und Kommunikationsaufwand ist nicht notwendig.

Ein weiteres Quellenauthentisierungsverfahren, das auf MACs basiert, wird in [Can99] vorgeschlagen. Dabei werden j verschiedene symmetrische Schlüssel zur Berechnung von j kryptographischen Message Authentication Codes verwendet. Die Empfänger der Nachricht m_i besitzen jeweils p (mit p<j) verschiedene Schlüssel, mit denen diese p kryptographische Message Authentication Codes verifizieren können. Die j verschiedenen Schlüssel wurden derart verteilt, dass ein Zusammenschluss von w Empfängern nicht in der Lage ist, alle j kryptographischen Message Authentication Codes zu berechnen. Die Sicherheit des Verfahrens basiert auf der Annahme, dass sich maximal w Nutzer zusammenschließen, um die Authentizität der Nachricht m_i zu verfälschen.

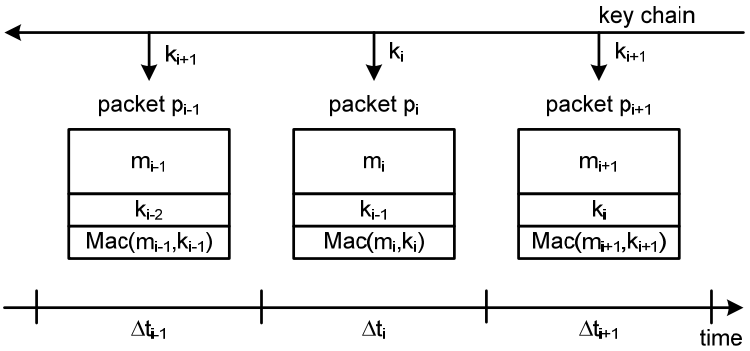

Abbildung 11: Funktionsprinzip der Timed Efficient Stream Loss-Tolerant Authentication

In [Per00] wird mit der Efficient Multicast Stream Signature (EMSS) eine verkettungs-basierte Quellenauthentisierung vorgestellt. Beim Einsatz der Efficient Multicast Stream Signature werden Nachrichten als `EmssPaket` versandt. Jedes `EmssPaket` pkt_k enthält die eigentliche Nachricht m_k und eine Prüfsumme Hash(m_{k-1}) des vorher gesendeten. Hierdurch werden diese untereinander verkettet. Am Ende der Übertragung wird das `EmssPaket` pkt_i, bestehend aus Nachricht m_i und der Prüfsumme Hash(m_{i-1}), gesendet. Um eine Authentifizierung möglich zu machen, wird nun zusätzlich eine digitale Signatur des `EmssPaket` pkt_i berechnet und hinzugefügt (Abbildung 12). Kommt dieses `EmssPaket` an und wird die in ihm enthaltene digitale Signatur verifiziert, sind die gesamten i Pakete damit rekursiv authentifiziert. Die rekursive Authentifizierbarkeit begründet sich über die mitgeführten Prüfsummen. Durch diese steht jedes `EmssPaket` in Abhängigkeit zum `EmssPaket`, das die digitale Signatur enthält. Zum Beispiel kann die Änderung des `EmssPakets` pkt_{i-2} und das Einfügen der Prüfsumme des veränderten Paketes in das `EmssPaket` pkt_{i-1} dadurch aufgedeckt werden, dass das `EmssPaket` pkt_i nicht die Prüfsumme des gefälschten Paketes pkt_{i-1} enthält. Das Konzept Efficient Multicast Stream Signature ist nicht fehlertolerant, da ein ausbleibendes `EmssPaket` die rekursive Authentisierung beendet. Es werden deshalb meistens gleichzeitig mehrere rekursive Verkettungen verwendet. In [Gol01] wird das beschriebene verkettungs-basierte Quellenauthentisierungsverfahren derart erweitert, dass trotz des Verlusts von mehreren aufeinander folgenden Paketen des Protokolls, die Quellenauthentisierung der erhaltenen Pakete verifiziert werden kann. Der in [Won99] spezifizierte Vorgänger von EMSS verwendet so genannte Prüfsummenbäume zur Verkettung. Sie haben allerdings den Nachteil, dass die Daten, die mit dem Verfahren geschützt übertragen werden sollten, vor der Übertragung vollständig bekannt sein müssen.

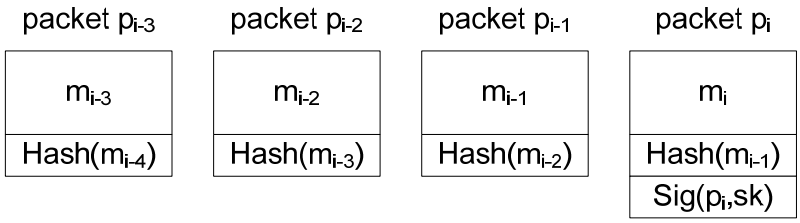

Abbildung 12: Funktionsprinzip der Efficient Multicast Stream Signature

2.5.2 Funktionaler Bereich 2: Schlüsselmanagement

Die in Abbildung 10 dargestellte Interaktion eines Nutzers mit dem GCKS mittels eines Sicherheitsmanagementprotokolls dient zur Etablierung einer Gruppensicherheitsassoziation (Group Security Association, GSA). Eine GSA besteht aus drei Arten von Sicherheitsassoziationen (Security Association, SA). Eine SA legt Umfang und Eigenschaften sicherer Verbindungen fest. Hauptbestandteil einer SA ist der für die sichere Verbindung verwendete Schlüssel. Meistens wird deshalb die Bezeichnung Schlüsselmanagement für die Verwaltung von SAs verwendet, obwohl neben dem Schlüssel auch andere Parameter, wie z.B. verwendete Algorithmen, vereinbart werden. Weiterhin wird meistens auch sprachlich nicht zwischen einer SA und dem enthaltenen Schlüssel differenziert. Die nachfolgenden drei Arten von Schlüsseln bzw. SAs sind durch ein Schlüsselmanagement zu etablieren:

- SA für die Registrierung
  Die SAs für die Registrierung (Register SA) oder Kategorie-1-SA legen den Umfang und die Eigenschaften der sicheren Verbindung zwischen GCKS und dem Nutzer fest. Der Kategorie-1-Schlüssel wird auch als Individualschlüssel bezeichnet. Dieser ist dem Nutzer und dem GCKS bekannt.
- SA für den Schlüsselwechsel
  Die SAs für den Schlüsselwechsel (Rekey SA) bzw. Kategorie-2-SA legen den Umfang und die Eigenschaften der sicheren Verbindungen, die zum Schlüsselwechsel verwendet werden, fest. Die Kategorie-2-Schlüssel werden auch als Key Encryption Keys (KEK) bezeichnet und sind allen Nutzern der Gruppe bekannt.
- SA für den Nutzdatenverkehr
  Die SAs für den Nutzdatenverkehr (Data SA) bzw. Kategorie-3-SAs legen den Umfang und die Eigenschaften der sicheren Verbindungen, die zum Austausch der Nutzdaten verwendet werden, fest. Mit den Kategorie-3-Schlüsseln, die auch als Traffic Encryption Keys (TEK) bezeichnet werden, wird der Nutzdatenverkehr geschützt. Die Bezeichnung TEK wird auch für Schlüssel verwendet, die zum Schutz der Integrität eingesetzt werden. Wird IPSec zum Nutzdatenschutz verwendet, bestimmen die Kategorie-3-SAs die Verwendung der IPSec-Protokolle AH und ESP.

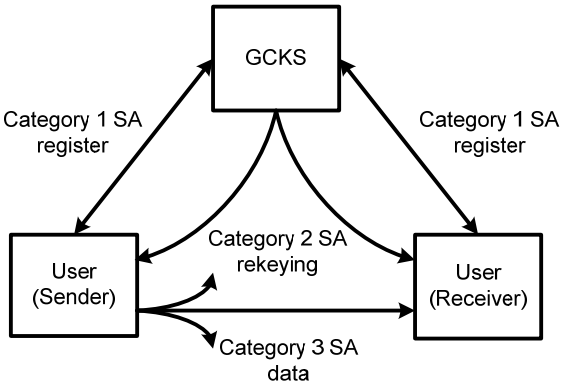

Abbildung 13: Security Associations für Multicast

2.5.3 Funktionaler Bereich 3: Gruppensicherheitsvorschrift

Eine Gruppensicherheitsvorschrift ist ein Satz von Regeln zur Festlegung der Teilnahmeberechtigungen und des sicherheitsrelevanten Verhaltens der Elemente des Multicast-Sicherheit-Referenzframeworks. Die zwei inhaltlichen Bestandteile der Gruppensicherheitsvorschrift sind die nachfolgend genannten Vorschriften:

- Vorschrift zur Teilnahmeberechtigung
- Vorschrift zur Gruppenschlüsselverwaltung und zum Nutzdatenschutz

Zur Festlegung von Format und Inhalt einer Gruppensicherheitsvorschrift wurde die Ismene Policy Description Language (IPDL) [Pra00], das Cryptographic Context Negotiation Template (CCNT) [Bal99] sowie das Group Security Policy Token Version 1 (GSPTv1) [Col06] spezifiziert. Die genannten Spezifikationen übersetzen die Gruppensicherheitsvorschrift in eine maschinenlesbare Form. Das Cryptographic Context Negotiation Template ist Bestandteil des Konzepts Dynamic Cryptographic Context Management (DCCM). Dieses beinhaltet auch die Aushandlung von Gruppensicherheitsvorschriften mit Hilfe des Cryptographic Context Negotiation Protocol (CCNP) [Din00]. Die Ismene Policy Description Language ist in das Antigone Framework eingebunden [Pra99]. In dem Framework ist eine Aushandlung der Vorschrift nicht vorgesehen. An der Umsetzung einer Gruppensicherheitsvorschrift sind alle funktionalen Elemente des Multicast-Sicherheit-Referenzframeworks beteiligt. Der GCKS setzt die festgelegten Zugangsberechtigungen der Gruppensicherheitsvorschrift durch. Außerdem stellt er sicher, dass die richtigen Methoden zur Schlüsselverwaltung verwendet werden. Die Nutzer der sicheren Multicast-Kommunikation überprüfen mit der Gruppensicherheitsvorschrift, ob sie das Schlüsselmaterial von einem zur Verteilung berechtigten Prozess erhalten haben. Außerdem stellen sie sicher, dass die für den Schutz der Nutzdaten festgelegten Sicherheitsmechanismen verwendet werden.

2.6 Multicast-Infrastrukturschutz

Die Betrachtungen dieses Abschnitts konzentrieren sich auf den Themenbereich Multicast-Infrastrukturschutz. Für den Multicast-Infrastrukturschutz sind drei Herausforderungen zu bewältigen. Zum Ersten ist die sichere Vervielfältigung und Verteilung der Multicast-Daten

durch die Router des Netzwerkes zu gewährleisten. Man spricht in diesem Zusammenhang vom Schutz des Verteilbaums für Multicast. Hierzu müssen die von den Multicast-Routing-Protokollen ausgetauschten Kontrollinformationen gegen Verfälschung und Herkunftsänderung geschützt werden. Zum Zweiten ist sicherzustellen, dass nur berechtigte Nutzer an eine Multicast-Gruppe senden. Als Drittes muss gewährleistet werden, dass nur berechtigte Nutzer den Datenempfang anfordern können. Durch die Bewältigung der beiden zuletzt genannten Herausforderungen kann erreicht werden, dass keine unnötigen Netzwerkressourcen verbraucht werden. Dieses wird häufig als Zugangsschutz zum Verteilbaum für Multicast für den Datenversand bzw. Datenempfang bezeichnet. Analog zum Multicast-Inhaltsschutz wurde auch für den Multicast-Infrastrukturschutz ein Multicast-Sicherheit-Referenzframework entworfen. Mittels dieses werden die drei funktionalen Bereiche Schutz des Multicast-Verteilbaums, Zugangsschutz zum Multicast-Verteilbaum für den Datenempfang bzw. den Datenversand graphisch veranschaulicht (vgl. Abbildung 14). Die fundamentalen Themenbereiche Schlüsselmanagement und Gruppensicherheitsvorschrift sind nicht nochmals dargestellt.

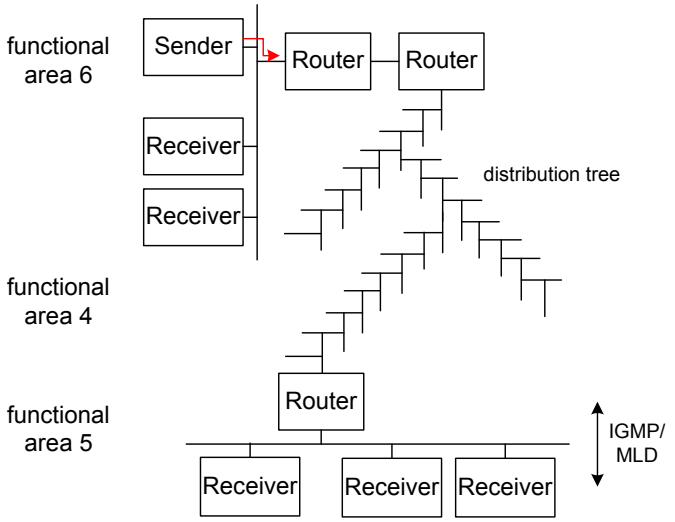

Abbildung 14: Multicast-Sicherheit-Referenzframework für den Multicast-Infrastrukturschutz

2.6.1 Funktionaler Bereich 4: Schutz des Multicast-Verteilbaums

Zur sicheren Verteilung und Vervielfältigung von Multicast-Paketen müssen existierende Multicast-Routing-Protokolle um Sicherheitsmechanismen erweitert wurden. Ziel der Sicherheitserweiterungen ist der Schutz der ausgetauschten Kontrollinformationen des Multicast-Routing-Protokolls. Ein allgemeines Konzept, das übertragbar auf verschiedene Multicast-Routing-Protokolle ist, existiert nicht. Aus diesem Grund werden die Sicherheitsmechanismen einiger ausgewählter Routing-Protokolle vorgestellt.

Das Konzept Scalable Multicast Key Distribution (SMKD) [Bal96] wurde zum Schutz der Verteilung und Vervielfältigung von Multicast-Paketen mit dem Core Based Tree Protocol (CBT) (vgl. Abschnitt 2.2.3) entworfen. Empfängt ein Router eine Beitrittsanfrage, d.h. eine Anfrage auf Multicast-Datenempfang eines Prozesses p_i und ist er nicht bereits Bestandteil des Verteilbaums für Multicast, so sendet er an den benachbarten Router die Nachricht `JoinCBT`. Diese signierte Nachricht enthält das Token des Prozesses p_i sowie sein eigenes

Token. Der benachbarte Router verifiziert diese und ersetzt das Token des benachbarten Routers durch sein eigenes Token. Diese Hop-by-Hop-Verifikation wird so lange durchgeführt, bis die Nachricht `JoinCBT` den so genannten Core des Routing-Protokolls erreicht. Nach einer erfolgreichen Verifikation durch den Core werden alle Router, die die Nachricht `JoinCBT` weitergeleitet haben, Bestandteil des Verteilbaums für Multicast. Erreicht einen Router, der bereits Teil des Multicast-Verteilbaums ist, die Anfrage des Prozesses p_i auf Multicast-Datenempfang, so kann dieser an den Prozess Multicast-Daten weiterleiten ohne den Core zu kontaktieren. Zusätzlich zum Aufbau eines sicheren Multicast-Verteilbaums ist mit SMKD eine Gruppenschlüsselbereitstellung zum Schutz der Multicast-Nutzdaten spezifiziert. Diese wird in Abschnitt 3.4.3 genauer erläutert.

Ein Schutz des Verteilbaums für Multicast kann auch durch den Einsatz des Keyed Hierarchical Multicast Routing Protocol (KHIP) erzielt werden [Shi99]. Hierbei handelt es sich um eine sichere Version des Hierarchical Multicast Routing Protocol (vgl. Abschnitt 2.2.3). Zur Beteiligung am Multicast-Verteilbaum versendet der Router eine Anfrage, die von der bereits bestehenden Multicast-Routing-Infrastruktur an den Center Point weitergeleitet wird. Die Anfrage enthält ein Zertifikat, das der Router vom Authentisierungsserver erhalten hat. Wird die Anfrage akzeptiert, ist der Router nach dem Austausch von weiteren drei Nachrichten Bestandteil des Multicast-Verteilbaums. Zusätzlich zum Aufbau eines sicheren Multicast-Verteilbaums ist in [Shi99] die Bereitstellung von Schlüsseln zum Schutz der Multicast-Nutzdaten spezifiziert. Diese wird in Abschnitt 3.4.3 genauer erläutert.

Zum Schutz der Kontrollinformation des Routing-Protokolls PIM wird in [Wei00] der Einsatz des IPSec-Sicherheitsprotokolls Authentication Header (AH) vorgeschlagen.

2.6.2 Funktionaler Bereich 5: Zugangsschutz zum Multicast-Verteilbaum für den Datenempfang

Eine Kontrolle, welcher Prozess berechtigt ist, Datenpakete von einer Multicast-Gruppe zu empfangen, wird häufig als Zugangsschutz zum Multicast-Verteilbaum für den Datenversand bezeichnet. Eine Realisierung der Empfangskontrolle beinhaltet die Integration von Mechanismen zur Gewährleistung von Authentizität und Integrität in die von den Routern zur Teilnehmerverwaltung eingesetzten Protokolle Internet Control Message Protocol (IGMP) bzw. Multicast Listener Discovery (MLD) (vgl. Abschnitt 2.2.2).

Begonnen wird der Überblick über Protokolle zum Zugangsschutz beim Datenempfang mit dem in [Cai00] beschriebenen Verfahren. Beabsichtigt ein Prozess p_i, den Datenverkehr einer Gruppe zu empfangen, stellt er eine Empfangs- bzw. Beitrittsanfrage beim Key Distributor. Ist der Prozess p_i empfangsberechtigt, erhält er ein `MulticastAccessToken` (Abbildung 15 links, Schritt 1). Der Key Distributor informiert den Authorization Server über die Empfangsberechtigung des Prozesses p_i (Abbildung 15 links, Schritt 2). Der Authorization Server verteilt periodisch oder auf Anfrage eine Liste mit den empfangsberechtigten Prozessen, die sogenannte `MulticastAccessList`, an alle Router (Abbildung 15 links, Schritt 3). Nachdem der Prozess p_i das `MulticastAccessToken` vom Key Distributor erhalten hat, sendet er es an den für ihn zuständigen Router weiter. Der Router überprüft die Authentizität des Tokens, sowie, ob dessen Absender in der `MulticastAccessList`, die

er vom Authorization Server bekommen hat, enthalten ist. Wenn ein Router die Anfrage des Prozesses p_i erfolgreich verifiziert hat, bestätigt er den Gruppenbeitritt und leitet den angeforderten Multicast-Datenverkehr an den Prozess p_i weiter (Abbildung 15 links, Schritt 4).

Das Empfängerkontrollverfahren Group Access Control Architecture (Gothic) [Jud02] kommt ohne die Verteilung einer Liste der empfangsberechtigten Nutzer aus. Ein Prozess p_i der den Datenverkehr einer Gruppe empfangen möchte, sendet ein `AuthorizationRequest` zum Access Conrol Server. Ist der Prozess empfangsberechtigt, erhält er von diesem einen `CapabilityToken` (Abbildung 15 rechts, Schritt 1). Dieses sendet der Prozess p_i anschließend an den für ihn zuständigen Router. Wenn der Router das `CapabilityToken` des Prozesses p_i erfolgreich verifiziert hat, bestätigt er den Gruppenbeitritt und leitet den angeforderten Multicast-Datenverkehr an diesen weiter (Abbildung 15 rechts, Schritt 2).

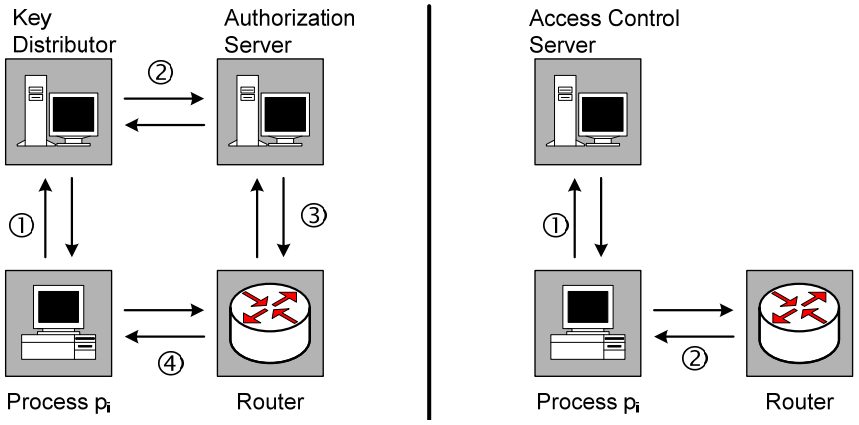

Abbildung 15: Konzept [Cai00] (links) und [Jud02] (rechts) zur Empfängerkontrolle bei Multicast

2.6.3 Funktionaler Bereich 6: Zugangsschutz zum Multicast-Verteilbaum für den Datenversand

Da mit den Protokollen Internet Group Management Protocol (IGMP) bzw. Multicast Listener Discovery (MLD) nur eine Empfängerverwaltung durchgeführt wird, sind zur Kontrolle des Multicast-Datenversands andere Verfahren notwendig. In [Cro95] wird ein Konzept zur Entdeckung und Verhinderung von unberechtigtem Datenversand an eine Multicast-Gruppe vorgeschlagen. Hierzu enthält jedes Multicast-Paket einen Zeit- und Berechtigungsstempel. Empfängt ein Router von einem unbekannten Absender Multicast-Pakete, so überprüft er mittels des Zeit- und Berechtigungsstempels des Datenpakets sowie eines im Netzwerk vorhandenen Authorization Servers die Berechtigung des Absenders zum Paketversand. Nur im Falle einer erfolgreichen Überprüfung werden die Multicast-Pakete weitergeleitet.

Wie bereits erwähnt ist jeder Nutzer in der Lage, Datenpakete an eine Multicast-Gruppe senden. Das Kommunikationsmuster wird deshalb auch als Any Source Multicast (ASM) bezeichnet. Wird die Weiterleitung von Paketen durch Multicast-Routing-Protokolle auf einige Sender beschränkt, so spricht man vom Kommunikationsmuster Source Specific Multicast (SSM). Dieses kann mit Protocol Independent Multicast (PIM) Sparce Mode Version 2 [Fen02] realisiert werden. Gruppen mit dem Kommunikationsmuster SSM enthalten von vornherein eine Senderkontrolle.

2.7 Kapitelzusammenfassung

Das Kapitel diente als Einführung in das Forschungsfeld sichere Gruppenkommunikation. IP-Multicast ist eine effiziente Möglichkeit zur Durchführung einer Gruppenkommunikation in auf der Internet-Technologie basierenden Rechnernetzwerken. Weiterhin wurde die dynamische Veränderung von Gruppengröße und Zusammensetzung als Auswirkung von Teilnehmeroperationen erläutert. Im weiteren Verlauf des Kapitels wurde die Bedeutung des Themenkomplexes Multicast-Sicherheit für die Akzeptanz von Multicast dargestellt. Diese wurde in die Themenbereiche Multicast-Inhaltsschutz und Multicast-Infrastrukturschutz unterteilt und das Schlüsselmanagement als fundamentale Komponente für die Bereitstellung von Multicast-Sicherheit identifiziert. Zur Verdeutlichung der zahlreichen Herausforderungen, die bei der Gewährleistung von Sicherheit für Multicast zu bewältigen sind, wurde das Multicast-Sicherheits-Referenzframework für Inhaltsschutz sowie für Infrastrukturschutz vorgestellt. Diese Herausforderungen lassen sich in die sechs funktionalen Bereiche (1) Sicherung der Nutzdaten, (2) Schlüsselmanagement, (3) Gruppensicherheitsvorschrift (4) Schutz des Multicast-Verteilbaums und (5/6) Zugangsschutz für den Datenempfang bzw. Datenversand einteilen. Gegenstand der Untersuchungen dieser Arbeit sind Forschungsarbeiten im Themenkomplex Schlüsselmanagement, d.h. dem zweiten funktionalen Bereich. Dieser wird im nächsten Kapitel analysiert. In den übrigen funktionalen Bereichen wurde der Stand der Wissenschaft beschrieben mit dem Ziel, die eigenen Arbeiten in die derzeitige Forschungslandschaft der Gruppenkommunikation einzuordnen.

3 Anforderungen an ein Schlüsselmanagement und Bewertung existierender Systeme

Basisvoraussetzung für die Nutzbarkeit von Sicherheitsmechanismen zum Schutz von Anwendungsdaten ist die Verfügbarkeit eines elektronischen Schlüssels bei allen an der Kommunikation beteiligten Nutzern (vgl. Abbildung 9). Neben den Nutzdatenschutzmechanismen benötigen auch Verfahren zum Schutz von Infrastruktur einen Schlüsselbereitstellungsdienst. Die Aufgabe eines Schlüsselmanagementsystems besteht darin, sicherzustellen, dass allen beteiligten und autorisierten Nutzern der korrekte Schlüssel zur Verfügung steht. Aus diesem Grund ist die Entwicklung von Gruppenschlüsselmanagementsystemen eine zentrale Herausforderung bei der Realisierung von Multicast-Sicherheit.

Als Einstieg in das Thema wird ein einfaches Schlüsselmanagement für zwei Nutzern vorgestellt. Ein derartiges System besitzt im Gegensatz zu einem Gruppenschlüsselmanagementsystem eine einfache Struktur. Zur Bewältigung der Schlüsselverwaltung in Gruppen muss eine geeignete Datenstruktur eingesetzt werden. Aus diesem Grund wird das Konzept Schlüsselbaum eingeführt und dessen Prinzipien an dem einfachen Beispiel der Schlüsselverwaltung zwischen zwei Teilnehmern erläutert. Anschließend wird eine Definition für Schlüsselbäume eingeführt, die unabhängig von Schlüsselbereitstellungsverfahren ist. Nach der Vorstellung des Schlüsselmanagements für die Punkt-zu-Punkt-Kommunikation werden die grundlegenden Anforderungen an ein Gruppenschlüsselmanagementsystem aufgestellt. Diese werden ergänzt durch zusätzliche Anforderungen, die zu erfüllen sind, wenn ein solches System innerhalb der Streitkräfte eingesetzt werden soll. Im weiteren Verlauf des Kapitels wird eine Bewertung der Erfüllung der Anforderungen durch existierende Schlüsselmanagementsysteme durchgeführt.

3.1 Schlüsselmanagement bei der Punkt-zu-Punkt-Kommunikation

Die Sicherheit der Punkt-zu-Punkt-Kommunikation ist ein gut erforschtes Gebiet. Es existieren deshalb für dieses Kommunikationsmuster bereits einige Schlüsselmanagementverfahren. Bei der Verwendung von asymmetrischen kryptographischen Algorithmen zum Inhaltsschutz kann ein sehr einfaches Schlüsselmanagement verwendet werden. Der öffentliche Schlüssel des Schlüsselpaares, mit dem die zu übertragenden Daten geschützt werden sollen, kann hierbei einfach über die ungeschützte Verbindung übertragen und muss nicht geheim gehalten werden. Allerdings benötigen asymmetrische Algorithmen sehr viel Rechenleistung, so dass in der Praxis zum Online-Schutz von Übertragungen symmetrische Algorithmen eingesetzt werden. Wichtige Voraussetzung für den Einsatz symmetrischer Algorithmen ist jedoch, dass die Teilnehmer im Besitz eines gemeinsamen Schlüssels sind. Als Hilfsmittel zur sicheren Bereitstellung dieses Schlüssels kann die Datenstruktur Schlüsselbaum verwendet werden. Während beim Schlüsselmanagement mit zwei Partnern der Einsatz eines Schlüsselbaums noch optional ist, ist dieser bei der Schlüsselverwaltung in Gruppen ein sinnvolles Hilfsmittel. Ein Schlüsselbaum ist ein spezieller Graph. Obwohl in der Literatur der Begriff des Schlüsselbaums häufig verwendet wird (z.B. [Kim00], [Won98], [Poo01]), existiert noch keine formale Definition. Nachfolgend

wird ausgehend von der Definition des Baumbegriffs der Graphentheorie der Schlüsselbaum definiert.

Definition: Gerichteter Graph [Toe02]
Ein gerichteter Graph G ist ein Tupel $G=(\mathcal{V},\mathcal{E})$. $\mathcal{V}$ ist eine endliche Knotenmenge (vertices), $\mathcal{E}$ ist eine endliche Kantenmenge (edges) mit $\mathcal{E}\subset\mathcal{V}x\mathcal{V}$. Ein Element $e\in\mathcal{E}$ mit $e_{12}=(v_1,v_2)$ heißt (gerichtete) Kante von v_1 nach v_2. Bei einem ungerichteten Graphen unterscheidet man nicht zwischen einer Kante $e_{12}=(v_1,v_2)$ und der Kante $e_{21}=(v_2,v_1)$. Bei der graphischen Darstellung eines gerichteten Graphen wird jede Kante mit einem Richtungspfeil versehen, welcher die Reihenfolge des zugeordneten Paares beschreibt. Ist $e_{12}=(v_1,v_2)$ eine solche gerichtete Kante, dann wird der Knoten v_1 als Vorgänger des Knotens v_2 und v_2 als Nachfolger von v_1 bezeichnet.

Definition: Pfad [Toe02]
Ein Pfad (path) von v nach v′ ist eine Folge $(v_0,v_1,\ldots,v_k)$ von Knoten mit $v=v_0$, $v'=v_k$ und $(v_i,v_{i+1})\in\mathcal{E}$ für alle $0\le i<k$. k ist die Länge des Pfades.

Der gerichtete Pfad heißt zyklenfrei, wenn er nicht zweimal denselben Knoten durchläuft ($v_m\neq v_k$ für $m\neq k$).

Definition: Zusammenhangskomponente [Toe02]
Ein Graph $G=(\mathcal{V},\mathcal{E})$ heißt zusammenhängend genau dann, wenn es für je zwei Knoten v und v′ einen Pfad von v nach v′ gibt. Eine Zusammenhangskomponente ist ein maximaler, zusammenhängender Teilgraph von G. Für einen Teilgraph $G'=(\mathcal{V}',\mathcal{E}')$ von G gilt $\mathcal{V}'\subset\mathcal{V}$ und $\mathcal{E}'=\mathcal{E}\cap(\mathcal{V}'x\mathcal{V}')$.

Definition: Schlüsselbaum
Ein gerichteter, zyklenfreier und zusammenhängender Graph $KT=(\mathcal{V},\mathcal{E})$ heißt Schlüsselbaum (Abbildung 16) (Key Tree, KT), wenn gilt:

- Es gibt genau einen Knoten $v_{root}\in\mathcal{V}$, der keinen Vorgänger hat. Dieser Knoten heißt Wurzelknoten (root node).
- Es gibt von dem Wurzelknoten v_j zu jedem anderen Knoten v_i, $i\neq j$, genau einen doppelpunktfreien gerichteten Pfad.
- Ein Knoten v_i, der keinen Nachfolger hat, heißt Endknoten (terminaler Knoten) oder Nutzerblatt (user leaf). Die übrigen Knoten werden als innere Knoten oder Schlüsselknoten (key nodes) bezeichnet.
- Jedem Knoten des Baumes wird ein Schlüssel zugeordnet. Der Schlüssel des Wurzelknotens wird als Gruppenschlüssel, der Schlüssel in einem inneren Knoten als Hilfsschlüssel und der Schlüssel in einem Nutzerblatt als Individualschlüssel bezeichnet.
- Bei einer Gruppe mit U Teilnehmern wird jedem Nutzer u_i $i=1,\ldots,U$, dem der Gruppenschlüssel bereitgestellt werden soll, ein Nutzerblatt v_k zugeordnet.
- Ein Nutzer, dem das Nutzerblatt v_i zugeordnet ist, kennt alle Knoten bzw. Schlüssel auf dem gerichteten Pfad $(v_i,\ldots,v_{root})$ vom Startknoten v_i zum Wurzelknoten v_{root}, d.h. insbesondere den Gruppenschlüssel.

Werden die Definitionen für Bäume aus der Graphentheorie auf Schlüsselbäume übertragen, so lassen sich die nachfolgenden drei Definitionen ableiten.

Definition: Grad eines Schlüsselbaums
Sei KT ein Schlüsselbaum KT=($\mathcal{V}$,$\mathcal{E}$) und $v_i \in \mathcal{V}$ ein Knoten von KT. Die Anzahl der Kanten, die von v_i ausgehen, wird als Ausgangsgrad $d_{out}(v_i)$ dieses Knotens bezeichnet. Die Anzahl der auf v_i zeigenden Kanten heißt Eingangsgrad $d_{in}(v_i)$ dieses Knotens. Der Grad des Schlüsselbaums KT wird definiert durch den maximalen Ausgangsgrad, den seine Knoten haben können.

Definition: Höhe eines Schlüsselbaumes
Sei KT ein Schlüsselbaum KT=($\mathcal{V}$,$\mathcal{E}$). Die Ebene eines Knotens $v_i \in \mathcal{V}$ ist gleich der Länge des Pfades von der Wurzel bis zu diesem Knoten, falls dem Wurzelknoten die Ebene Null zugewiesen wurde. Die höchste auftretende Ebene heißt die Höhe h des Schlüsselbaums KT.

Definition: Ausgeglichener (balancierter) Schlüsselbaum
Sei KT ein Schlüsselbaum KT=($\mathcal{V}$,$\mathcal{E}$). Dieser Schlüsselbaum heißt ausgeglichen oder balanciert, wenn für jeden inneren (nicht terminalen) Knoten $v_i \in \mathcal{V}$ die Pfadlängen bis zu jedem seiner Blätter höchstens um 1 differieren.

Definition: Vollständiger Schlüsselbaum
Sei KT ein Schlüsselbaum KT=($\mathcal{V}$,$\mathcal{E}$) von Grad d. Dieser Schlüsselbaum heißt vollständig, wenn alle Blätter die gleiche Pfadlänge zur Wurzel haben und alle nicht terminalen Knoten d Nachfolger besitzen.

Bei Bäumen spricht man statt von Vorgängern und Nachfolgern, wie es bei Graphen üblich ist, meistens von Vätern und Kindern. Ist eine Kante $e_{ih}=(v_i,v_h)$ vorhanden, dann heißt v_i Vater (father) von v_h und v_h Kind (child) von v_i. Gibt es eine weitere Kante $e_{ik}=(v_i,v_k)$, die von v_i zu einem anderen Knoten v_k (weiteres Kind) führt, dann werden v_h und v_k als Geschwister (sibling) bezeichnet. Zu Knoten, deren Ausgangsgrad kleiner als der Grad des Schlüsselbaumes ist, können Nullknoten (null node) als Platzhalter hinzugefügt werden, bis deren Ausgangsgrad mit dem Grad des Schlüsselbaums übereinstimmt.

Zur eindeutigen Bezeichnung eines Knotens kann diesem eine Knotennummer (Id) zugewiesen werden. Bei dieser Art der Bezeichnung wird dem Wurzelknoten die Nummer null zugeordnet und die übrigen Knoten durchnumeriert (Abbildung 16, links). Zusätzlich zur Id können einem Knoten v die zwei Indizes Ebene ℓ (level) und Ebenenposition p (position), d.h. $v_{\ell,p}$, zugeordnet werden. Die Ebenenposition wird ermittelt, indem die Knoten einer Ebene des Schlüsselbaums von links nach rechts durchnumeriert werden (Abbildung 16, rechts). Die größte im Baum auftretende Ebenenposition wird mit p^*_{max} bezeichnet. Die beiden Knotenbezeichnungen sind äquivalent. Deshalb ist es möglich, aus den beiden Indizes Ebene und Ebenenposition rekursiv die Knotennummer zu berechnen:

$$Id_{child_k} = d \cdot Id_{father} + k \qquad \text{mit } k = 0,...,d-1$$

$$\ell = \ell_{father} + 1$$

$$p_{child_k} = d \cdot p_{father} + k \qquad \text{mit } k = 0,...,d-1$$

Die eingesetzte Knotenbezeichnung ist abhängig vom Schlüsselbereitstellungsverfahren. Im Rahmen dieser Arbeit werden beide Verfahren eingesetzt. Die Verwendung der beiden Knotenbezeichnungen in einem Schlüsselbaum ist in Abbildung 16 veranschaulicht. In einem vollständigen Baum von Grad d gilt für die größte Ebenenposition $p^*_{max}=U$ und für die Höhe $h=\log_d U$.

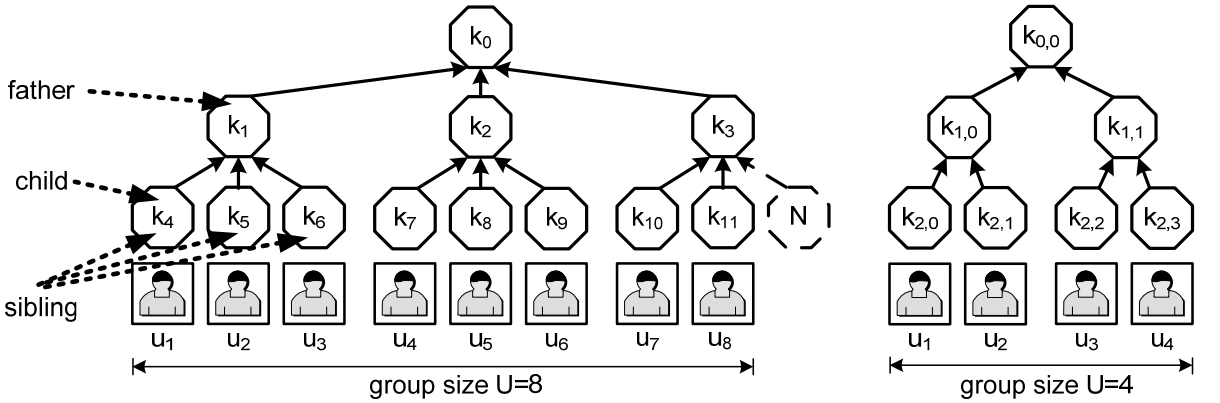

Abbildung 16: Schlüsselbaum vom Grad drei mit Nullknoten (links) und vom Grad zwei (rechts)

Für den Fall, dass der Schlüsselbaum den Grad zwei besitzt, spricht man von einem binären Schlüsselbaum. Ein innerer Knoten $v_{\ell,p}$ besitzt dann genau einen linken Nachfolger $v_{\ell+1,2p}$ und einen rechten Nachfolger $v_{\ell+1,2p+1}$. In diesem Fall sind die Knoten $v_{\ell+1,2p+1}$ und $v_{\ell+1,2p}$ Geschwister. Die bei der Beschreibung eines Schlüsselbaums verwendeten Notationen sind in Tabelle 2 zusammengefasst.

KT	Schlüsselbaum (key tree)
v_i	Knoten (vertex) i des Schlüsselbaums
$v_{\ell,p}$	Knoten der Ebene ℓ (level) und Ebenenposition p (position) des Schlüsselbaums
V	Anzahl der Knoten des Schlüsselbaums
L	Anzahl der Blätter des Schlüsselbaums
$v_{0,0}=v_{root}$	Wurzelknoten (root node) des Schlüsselbaums
d	Grad (degree) des Schlüsselbaums
h	Höhe (high) des Schlüsselbaums = maximale Ebene des Schlüsselbaums
$p_{max}(\ell)$	Größte auf der Ebene ℓ vorhandene Ebenenposition des Schlüsselbaums
p^*_{max}	Größte im Schlüsselbaum auftretende Ebenenposition
$c_{\ell_k,p_k} \in child(v_{\ell,p})$	Nachfolger k des Knoten $v_{\ell,p}$
$w_{\ell_k,p_k} \in path(v_{\ell,p})$	Knoten k auf dem Pfad $(v_{\ell,p},\ldots,v_{0,0})$ vom Startknoten $v_{\ell,p}$ zum Wurzelknoten $v_{0,0}$
$\|path(v_{\ell,p})\|$	Länge des Pfads $(v_{\ell,p},\ldots,v_{0,0})$ vom Startknoten $v_{\ell,p}$ zum Wurzelknoten $v_{0,0}$
u_i	Nutzer i des Schlüsselbaums
U	Anzahl der Nutzer im Schlüsselbaum

Tabelle 2: Notationen für die Beschreibung eines Schlüsselbaums

Die Bereitstellung eines Schlüssels unter Verwendung eines Schlüsselbaums wird nun am Beispiel des Diffie-Hellman-Algorithmus erläutert. Der Diffie-Hellman-Algorithmus (DH-Algorithmus) ist ein Verfahren zur Vereinbarung eines gemeinsamen symmetrischen Schlüssels zwischen zwei Nutzern u_1 und u_2 [Dif76]. Hierzu einigen sich die beteiligten

Nutzer im Vorfeld auf eine Primzahl p und einen Generator g der multiplikativen Gruppe $\mathbb{Z}_p^*$, d.h. ein Element g, für das $\{g^1, g^2, \ldots, g^{p-1}\} = \mathbb{Z}_p^*$ gilt. Weiterhin generiert jeder Nutzer u_i zufällig unter Gleichverteilung ein Element k_i aus $\mathbb{Z}_p^*$. k_i wird als (geheimer) Schlüssel bezeichnet, während $bk_i = BK(k_i) = g^{k_i} \bmod p$ öffentlicher Schlüssel bzw. Blindschlüssel (Blind Key, bk) des Verfahrens genannt wird. Der Begriff öffentlicher Schlüssel wird meistens im Zusammenhang mit digitalen Signaturen bei Verwendung von asymmetrischen Verschlüsselungsalgorithmen verwendet (vgl. Abschnitt 2.3). Um eine Verwechslung auszuschließen, wird in dieser Arbeit die Bezeichnung Blindschlüssel verwendet.

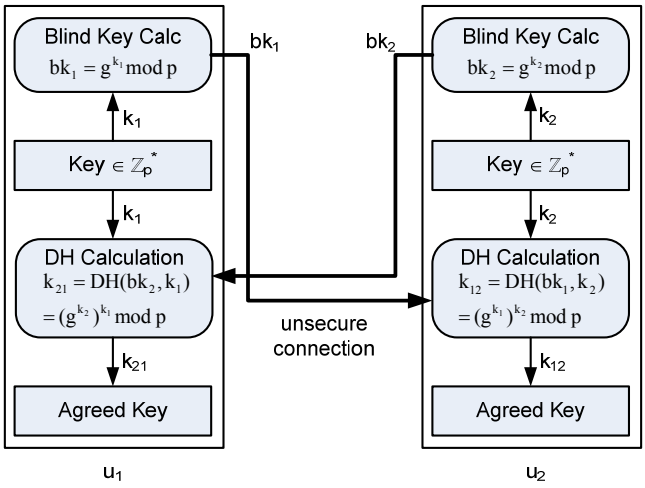

Abbildung 17: Exponentielle Diffie-Hellman-Algorithmus

Mit dem Diffie-Hellman-Algorithmus vereinbaren die beiden Nutzer den gemeinsamen Schlüssel $k_{12} = k_{21}$, ohne diesen oder die Teilschlüssel k_1 bzw. k_2 über die unsichere Verbindung zu übertragen (Abbildung 17). Der gemeinsame Schlüssel wird mittels der Funktion $k_{ij} = (g^{k_i})^{k_j} \bmod p$ berechnet, d.h. die Nutzer müssen zur Schlüsselermittlung eine Exponentiation durchführen. Zur Vereinfachung wird die Notation $k_{ij} = DH(bk_i, k_j) = (g^{k_i})^{k_j} \bmod p$ definiert. Die Sicherheit des DH-Algorithmus beruht auf der hohen Komplexität der Umkehrung der Funktion $f(x) = g^x \bmod p$, wobei p eine Primzahl und g ein Generator von $\mathbb{Z}_p^*$ ist. Das Problem gilt nach heutigem Stand der Wissenschaft als nicht-effizient berechenbar, falls g ein Generator der multiplikativen Gruppe $\mathbb{Z}_p^*$ ist und die Primzahl p hinreichend groß gewählt wird. Die Primzahl p und der Generator g werden als Parameter des Verfahrens bezeichnet. In den letzten Jahren wird der vorgestellte DH-Algorithmus, genauer gesagt der exponentielle (reguläre) Diffie-Hellman-Algorithmus, interpretiert als Beispiel für ein Verfahren, bei dem zwei Prozesse aus geheimem Schlüssel und Blindschlüssel einen gemeinsamen Schlüssel ermitteln. Ein weiteres Beisipiel für ein derartiges Verfahren ist der DH-Algorithmus auf der Basis von elliptischen Kurven. Bei diesem kryptographischen Algorithmus sind die elliptische Kurve, eine Punkt auf der elliptischen Kurve und eine Primzahl p die Parameter des Verfahrens. Wird bei einem Verfahren, bei dem zwei Prozesse aus geheimen Schlüssel und Blindschlüssel einen gemeinsamen Schlüssel ermitteln, ein Schlüsselbaum eingesetzt, so besitzt der Schlüsselbaum des Nutzers u_1 die in Abbildung 18 dargestellte Struktur. Der gemeinsame Schlüssel $k_{12}=k_{21}$ wird der Baumwurzel zugeordnet. Hierbei wurde für den gemeinsamen Schlüssel $k_{12}=k_{21}$ die bei Schlüsselbäumen übliche Bezeichnung $k_{0,0}$ verwendet. Jedem beteiligten Nutzer wird ein

Nutzerblatt des Schlüsselbaumes zugewiesen, z.B. u_1 das Nutzerblatt $v_{1,0}$. Der geheime Schlüssel des Nutzers u_1 ist deshalb in dem Knoten $v_{1,0}$ abgelegt worden.

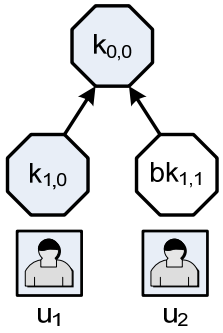

Abbildung 18: Schlüsselbaum des Nutzers u_1 beim Diffie-Hellman-Algorithmus

Die bei der Beschreibung der Schlüsselbereitstellungsmechanismen verwendeten Notationen sind in Tabelle 3 zusammengefasst.

k_i	Schlüssel (key) k_i
$k_{\ell,p}$	Schlüssel des Knotens $v_{\ell,p}$
$bk_i=BK(k_i)$	Blindschlüssel (blind key) von k_i
$bk_{\ell,p}=BK(k_{\ell,p})$	Blindschlüssel des Knotens $v_{\ell,p}$
$DH(bk_i,k_j)$	Vereinbarter Schlüssel bestehend aus den Teilschlüsseln k_i, k_j

Tabelle 3: Notationen für die Beschreibung von Schlüsselbereitstellungsmechanismen

3.2 Grundlegende Anforderungen an ein Gruppenschlüsselmanagement

In diesem Abschnitt werden die grundlegenden Anforderungen an ein Verfahren zur Bereitstellung eines gemeinsamen Schlüssels für mehr als zwei Teilnehmer, d.h. einer Gruppe von Teilnehmern, beschrieben. Soll in einer dynamischen Gruppe der gemeinsame Schlüssel in Kombination mit einem Sicherheitsprotokoll dazu verwendet werden, Integrität, Vertraulichkeit, Authentizität und Zugriffskontrolle zu gewährleisten, so darf dieser nur den beteiligten Prozessen bekannt sein und muss bei einer Veränderung der Gruppenzusammensetzung gewechselt werden. Formal können die Anforderungen an die Schlüsselbereitstellung für eine dynamische Gruppe $\mathcal{U} \in \{U_1,...,U_M\}$ mit $U_k=(u_{k_1},u_{k_2},...,u_{k_i})$ k=1,…,M, deren Größe und Zusammensetzung sich m mal ändern und denen die Gruppenschlüssel $\mathcal{K} \in \{k_1,k_2,\ldots,k_M\}$ bereitgestellt werden, wie folgt definiert werden [Kim00]:

- Group Key Secrecy
  Group Key Secrecy (Geheimhaltung des Gruppenschlüssels) fordert, dass ein passiver Nutzer, der alle Kommunikationsvorgänge abhört, nicht in den Besitz eines Gruppenschlüssels $k_i \in \mathcal{K}$ mit i=1,…,M gelangt.
- Forward Secrecy
  Forward Secrecy (Geheimhaltung Vorwärts) fordert, dass ein Nutzer, der über die zusammenhängende Teilmenge an Gruppenschlüsseln $\{k_i,k_{i+1},\ldots,k_j\}$ verfügt, nicht in den Besitz eines zukünftigen Gruppenschlüssels k_ℓ für alle i,j,ℓ mit $0 \le i < j < \ell$ gelangt.

- Backward Secrecy
  Backward Secrecy (Geheimhaltung Rückwärts) fordert, dass ein Nutzer, der über die zusammenhängende Teilmenge an Gruppenschlüsseln $\{k_i,k_{i+1},\ldots,k_j\}$ verfügt, nicht in den Besitz vorher verwendeter Gruppenschlüssel k_ℓ für alle i,j,ℓ mit $\ell \leq i < j$ gelangt.

Zusätzlich ist in [Kim00] die Sicherheitsvorschrift Key Independence definiert. Diese garantiert, dass ein Nutzer, der im Besitz einer Teilmenge $\mathcal{K}'$ mit $\mathcal{K}' \subset \mathcal{K}$ von Gruppenschlüsseln ist, nicht in den Besitz eines Gruppenschlüssels der Teilmenge $\mathcal{K}'' \in (\mathcal{K} \backslash \mathcal{K}')$ gelangt. Dies ist aber keine zusätzliche Vorschrift sondern eine Konsequenz der drei übrigen Sicherheitsvorschriften an den Schlüsselwechsel. Die Sicherheitsanforderung Key Independence wird auch oftmals als Forderung nach einem kollisionsfreien Schlüssel bezeichnet.

Werden alle der oben genannten Sicherheitsvorschriften erfüllt, so wird ein derartiger Schlüsselwechsel als teilnehmersensitiv bezeichnet. Denkbar ist auch, einen Schlüsselwechsel nur nach einer der Teilnehmeroperationen LEAVE, PARTITION bzw. EJECT durchzuführen. In diesem Fall ist die Vorschrift Backward Secrecy nicht erfüllt und der Schlüsselwechsel wird als austrittssensitiv bezeichnet. Im Gegensatz dazu erfüllt ein beitrittssensitiver Schlüsselwechsel, d.h. ein Wechsel des Schlüssels nur nach den Teilnehmeroperationen JOIN bzw. MERGE, nicht die Vorschrift Forward Secrecy. Tabelle 4 gibt eine Übersicht über die Bezeichnungen und die Kriterien für die Durchführung eines Schlüsselwechsels. Hierbei bedeutet das Kriterium TIME, dass ein Schlüsselwechsel unabhängig von Teilnehmeroperationen nach einer festgesetzten Zeitperiode durchgeführt wird.

	TIME	JOIN	MERGE	LEAVE	PARTITION	EJECT
teilnehmersensitiv	-	X	X	X	X	X
beitrittssensitiv	-	X	X	-	-	-
austrittssensitiv	-	-	-	X	X	X
ausschlusssensitiv	-	-	-	-	-	X
zeitsensitiv	X	-	-	-	-	-

Tabelle 4: Bezeichnung des Schlüsselwechsels und Kriterien für dessen Durchführung

Zusätzlich zu dem Bereitstellungsmechanismus für einen dynamischen Schlüssel, der Sicherheitsvorschriften erfüllt, muss ein störsicheres Verfahren zur Gruppenanmeldung bzw. Gruppenabmeldung verfügbar sein. Dies muss einen Authentisierungsmechanismus enthalten, der bei einer Gruppenanmeldung die Berechtigung eines Nutzers zum Erhalt des Gruppenschlüssels überprüft. Die in diesem Absatz definierten grundlegenden Anforderungen an ein Schlüsselmanagement sind in Tabelle 5 zusammengefasst.

Nr.	Bezeichnung	Kurzbeschreibung der Anforderung
1	Schlüsselgeheimhaltung	Bereitstellung eines dynamischen Gruppenschlüssels und Gewährleistung der Sicherheitsvorschriften Group Key Secrecy, Backward Secrecy, Forward Secrecy
2	Zugriffskontrolle	Störsichere Mechanismen zur Überprüfung der Zugangsberechtigung bei der Gruppenanmeldung

Tabelle 5: Grundlegende Anforderungen an ein Gruppenschlüsselmanagement

3.3 Spezifische Anforderungen an ein Gruppenschlüsselmanagement für die Streitkräfte

Soll ein Multicast-Inhaltsschutz auch in den Streitkräften ermöglicht werden, muss das für die Gruppenschlüsselbereitstellung verantwortliche Schlüsselmanagementsystem zusätzliche Forderungen erfüllen. Diese hängen vom Aufgabenbereich der Streitkräfte ab, obwohl es sicherlich einige Überschneidungen gibt. [Bun04] identifiziert fünf Aufgabenbereiche der Streitkräfte. Ein Aufgabenbereich umfasst Auslandseinsätze im Rahmen von Konfliktverhütung und Krisenbewältigung. Dieser Aufgabenbereich bildet die Grundlage für die Forderungen an ein Schlüsselmanagement. Er wurde ausgewählt, weil derartige Einsätze auf absehbare Zeit am wahrscheinlichsten sind [Bun04]. Im Rahmen einer Studie zur zukünftigen mobilen Kommunikation im Heer wurde das Kommunikationsverhalten der Streitkräfte bei Auslandseinsätzen im Rahmen von Konfliktverhütung und Krisenbewältigung untersucht [Ebe03]. Die Analyse des Kommunikationsverhaltens ergab, dass hierbei siebenunddreißig verschiedene Kommunikationsprofile auftreten. Die absoluten Häufigkeiten der Gruppengrößen bei den Kommunikationsprofilen sind in Abbildung 19 dargestellt. Die Abbildung verdeutlicht, dass Gruppen mit einer Größe von bis zu 20 Nutzern am häufigsten sind. Das Kommunikationsprofil von Großgefechtsständen weist dabei mit 300 Nutzern die größte Teilnehmerzahl auf.

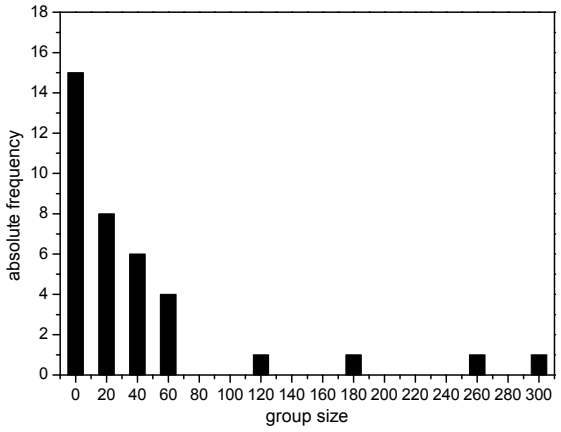

Abbildung 19: Absolute Häufigkeit der Gruppengröße bei den Kommunikationsprofilen in [Ebe03]

Einsätze der Streitkräfte im Rahmen von Konfliktverhütung und Krisenbewältigung erfolgen häufig in Gebieten, in denen keine oder nur eine mangelhafte Kommunikationsinfrastruktur verfügbar ist. Ein im Rahmen des Einsatzes aufgebaute Kommunikationsinfrastruktur besteht im günstigsten anzunehmenden Fall aus einem drahtgebundenen Übertragungsmedium, das auf dem IEEE-Standard 802.3 mit einer Datenübertragungsrate von 100 Mbit/s basiert. Dieses wird im Folgenden mit Ethernet bezeichnet. Im ungünstigsten anzunehmenden Fall wird ein MANET, basierend auf VHF-Funk-Verbindungen, im Folgenden nur mit VHF bezeichnet, eingesetzt [Ebe03]. Typische hierfür verwendete Geräte, z.B. das SEM 90/93, ermöglichen eine Datenübertragungsrate von 16 Kbit/s [Tha06]. In der aufgebauten Kommunikationsinfrastruktur werden Realzeit-Anwendungen, z.B. Sprachkommunikation (Voice over IP, VoIP), Videoübertragung, sowie nachrichtenorientierte Anwendungen, z.B. Email, betrieben [Ebe03]. Nachdem das Einsatzumfeld dargestellt wurde, werden nun Anforderungen an das Schlüsselmanagement aufgestellt und deren Notwendigkeit begründet.

- Schlüsselgeheimhaltung
  Weil sich durch Gruppeneintritt und Gruppenaustritt die Gruppenzusammensetzung ändert, muss ein Schlüsselmanagement einen dynamischen Gruppenschlüssel bereitstellen. Für die Schlüsselbereitstellung ist die Sicherheitsvorschrift Group Key Secrecy, Backward Secrecy, Forward Secrecy zu erfüllen.
- Zugriffskontrolle
  Zur Gruppenanmeldung bzw. Gruppenabmeldung muss ein störsicheres Verfahren vorhanden sein. Dieses muss einen Authentisierungsmechanismus enthalten, der bei einer Gruppenanmeldung die Möglichkeit bietet, die Berechtigung eines Nutzers zum Erhalt des Gruppenschlüssels zu überprüfen. Zusätzlich muss der Authentisierungsmechanismus einen Widerruf der Berechtigung zum Erhalt des Gruppenschlüssels unterstützen. Dieses ist notwendig, um kompromittierte Nutzer vom Erhalt des Gruppenschlüssels auszuschließen. Ein weiterer Aspekt ist die Störsicherheit der Verfahren zur Gruppenanmeldung bzw. Gruppenabmeldung gegen Angriffe durch wiederholtes Senden und durch Fluten des Systems mit Nutzeranfragen.
- Skalierbarkeit
  Die Forderung Skalierbarkeit wird unter den Gesichtspunkten schnelle Mechanismen zum Schlüsselwechsel bei steigender Gruppengröße und bei geringen verfügbaren Kommunikationsressourcen erläutert. Ein schneller Schlüsselwechselmechanismus ist erforderlich, da zur Zeit des Schlüsselwechsels keine Nutzdaten ausgetauscht werden können. Begonnen wird die Erläuterung mit der Effizienz bei steigender Gruppengröße. Eine Analyse der Kommunikationsprofile aus [Ebe03] zeigt, dass in vielen militärischen Szenarios, z.B. Spähtrupp zu Fuß, nur eine geringe Anzahl von Nutzern miteinander kommuniziert. In den meisten Fällen ist deshalb eine Schlüsselbereitstellung für kleine Gruppen, in denen der Schlüsselverwalter aus Sicht der elektronischen Kommunikation nicht von vornherein fixiert, notwendig. Allerdings findet auch eine Kommunikation in großen Gruppen mit bis zu 300 Nutzern statt, für die ein Gruppenschlüssel bereitgestellt werden muss. Insbesondere sind kleine Gruppen zu berücksichtigen, die während ihrer Existenz einem erheblichen Teilnehmerzuwachs unterliegen. Nachfolgend wird auf den Aspekt geringe verfügbare Kommunikationsressourcen für den Schlüsselwechsel eingegangen. Um auch bei geringen Ressourcen einen Schlüsselwechsel zu ermöglichen, muss dieser bezüglich benötigter Datenübertragungskapazität und Rechenleistung effizient durchgeführt werden. Insbesondere bezüglich der Datenübertragungsraten darf der Schlüsselwechsel im ungünstigen Fall nur wenig Ressourcen erfordern, da in einigen militärischen Szenarios nur eine MANET mit einer Datenübertragungsrate von 16 Kbit/s (VHF) verfügbar sein kann. Geringe Rechenleistung in Kombination mit geringer Datenmenge für den Schlüsselwechsel stellen sicher, dass ein Einsatz auch bei limitierten Energieressourcen möglich ist.
- Verlässlichkeit
  Bei Verwendung eines Schlüsselmanagements in einem militärischen Einsatzbereich muss eine hohe Verlässlichkeit gewährleistet sein. Da in diesem Bereich neben drahtgebundenen Netzwerken auch MANETs eingesetzt werden, sind zwei Aspekte zu berücksichtigen.

(1) Robuste Schlüsselübermittlung beim Schlüsselwechsel: Zum einen ist durch eine robuste Schlüsselübermittlung sicherzustellen, dass in einer Kommunikationsinfrastruktur mit temporären Kommunikationsfehlern, d.h. Paketverlusten, eine Schlüsselbereitstellung gewährleistet werden kann.
(2) Reparierbarkeit: Außerdem muss im Fehlerfall, z.B. Prozessausfall oder Verbindungsverlust, eine Möglichkeit zur Reparatur des Schlüsselbereitstellungssystems durch Redundanzen bestehen. Deshalb ist in einem Schlüsselmanagementkonzept zu berücksichtigen, dass eine Schlüsselbereitstellung, die keinen Single Point of Failure (SPoF) aufweist, möglich sein muss.

- Mehrfachanfragen
  Beim Einsatz von Streitkräften können Truppenteile umgegliedert werden. Hierbei werden Teile aus militärischen Verbänden herausgenommen und entsprechend der zu erwartenden Aufgaben anderen Verbänden zugeteilt. Dieses bedeutet, dass es Zeiträume gibt, in denen eine starke Fluktuation in den Verbänden auftritt. Für derartige umfangreiche Änderungen muss ein effizienter Mechanismus verfügbar sein.

- EMCON
  Eine Schlüsselübermittlung muss auch an Teilnehmer mit temporärer Simplexverbindung möglich sein. Eine derartige Situation ist die Folge des Betriebs der Kommunikationsmittel im Zustand Emission Control (EMCON), bei dem Daten empfangen, aber nicht versendet werden dürfen. Ein derartiger Zustand der Kommunikationsmittel kann erforderlich sein, wenn eine funkbasierte Kommunikationsinfrastruktur eingesetzt wird und ein Schutz vor Aufkärungsmaßnahmen gewährleistet werden soll.

- IPSec-Adaption
  Zum Schutz der Nutzdaten bei Multicast soll IPSec eingesetzt werden. Dieses hat sich bereits beim Schutz der Punkt-zu-Punkt-Kommunikation bewährt. Eine Gruppenschlüsselbereitstellung für dieses Sicherheitsprotokoll ist deshalb die Hauptaufgabe des Schlüsselmanagements.

- Realzeit
  Eine wichtige Anwendung in den Streitkräften ist die Sprachkommunikation. Das Gruppenschlüsselmanagement muss deshalb auch in der Lage sein, Schlüssel für die Übertragung von Realzeitdaten bereitzustellen.

Die in diesem Abschnitt definierten Anforderungen an ein Schlüsselmanagement bei Verwendung in den Streitkräften sind in Tabelle 6 zusammengefasst. Die ersten beiden Anforderungen in der Tabelle sind identisch mit den grundlegenden Anforderungen an ein Gruppenschlüsselmanagement. Die restlichen Anforderungen resultieren aus dem spezifischen Einsatzumfeld. Außerdem sind in der Tabelle die Anforderungen unter dem Gesichtspunkt Verlässlichkeit als zwei getrennte Anforderungen aufgeführt.

Nr.	Bezeichnung der Anforderung	Kurzbeschreibung der Anforderung
1	Schlüsselgeheimhaltung	Bereitstellung eines dynamischen Gruppenschlüssels und Gewährleistung der Vorschriften Group Key Secrecy, Backward Secrecy, Forward Secrecy. Diese Anforderung wird auch oft als Sicherheitsvorschrift für die Schlüsselbereitstellung bezeichnet.
2	Zugriffskontrolle	Störsichere Mechanismen zur Überprüfung der Zugangsberechtigung bei der Gruppenanmeldung
3	Skalierbarkeit	Gewährleistung der Schlüsselbereitstellung durch einen effizienten Mechanismus zum Gruppenschlüsselwechsel in kleinen Gruppen als auch bei steigender Gruppengröße und bei geringen verfügbaren Kommunikationsressourcen
4	Robuste Schlüsselübermittlung	Kompensation temporärer Kommunikationsfehler bei Schlüsselübermittlung
5	Reparierbarkeit	Fehlertoleranz gegenüber einem Zusammenbruchfehler oder Verbindungsverlust eines Nutzers bzw. des Schlüsselverwalters
6	Mehrfachanfragen	Schlüsselbereitstellung bei kurzzeitiger starker Fluktuation in der Gruppe
7	EMCON	Schlüsselübermittlung an Nutzer mit Simplexverbindung
8	IPSec-Adaption	Schlüsselbereitstellung für das Sicherheitsprotokoll IPSec
9	Realzeit	Schlüsselbereitstellung zum Schutz der Datenübertragung von Realzeit-Anwendungen

Tabelle 6: Anforderungen an ein Gruppenschlüsselmanagement bei Verwendung in den Streitkräften

Zur Strukturierung der Anforderungen sind diese in Abbildung 20 in die vier Kategorien Sicherheit, Effizienz, Verlässlichkeit und Sonstige eingeteilt.

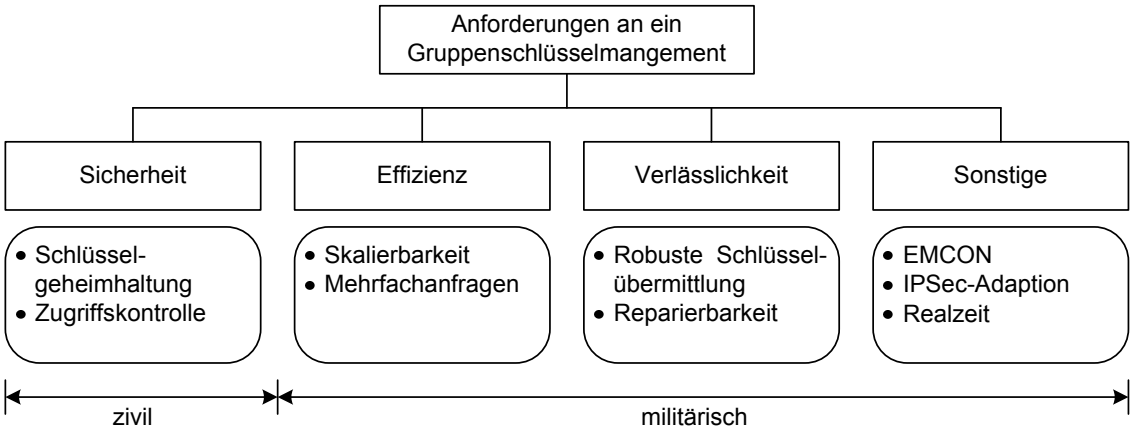

Abbildung 20: Strukturierung der Anforderungen an ein Schlüsselmanagement für die Streitkräfte

Die Tabelle 6 dient im Abschnitt 3.4 als Prüfliste zur Bewertung existierender Schlüsselmanagementverfahren. Die hierbei verwendeten Bewertungsstufen sind in Tabelle 7 zusammengefasst. Für die Bewertung der Skalierbarkeit (Forderung 3, Tabelle 6) wird die Effizienz des Schlüsselwechsels analysiert. Hierzu wird die Anzahl der übertragenen Nachrichten für den Schlüsselwechsel betrachtet. Eine detaillierte Einführung der Metriken zur Effizienzbewertung wird in Abschnitt 6.2 gegeben.

Bewertungsstufe	Kennzeichen	Erläuterung
erfüllt	+	Diese Bewertungsstufe wird verwendet für den Fall, dass ein Schlüsselmanagementsystem eine Anforderung erfüllt.
prinzipiell erfüllbar	0	Bei prinzipiell erfüllbaren Forderungen kann durch Modifikationen des Konzepts diese Forderung erfüllt werden.
nicht erfüllt	-	Im Gegensatz zu prinzipiell erfüllbaren Forderungen kann bei nicht erfüllten Anforderungen auch durch eine Modifikation des Konzeptes diese Anforderung nicht erfüllt werden.
unklar	?	Diese Bewertungsstufe wird verwendet für den Fall, dass der Anforderung keine der übrigen Bewertungsstufen zugeordnet werden kann.

Tabelle 7: Bewertungsstufen bei der Analyse der Anforderungen an ein Gruppenschlüsselmanagement

3.4 Bewertung existierender Gruppenschlüsselmanagementkonzepte

In diesem Abschnitt werden existierende Gruppenschlüsselmanagementkonzepte vorgestellt. Hierbei wird kurz das Funktionsprinzip des Verfahrens vorgestellt. Anschließend wird bewertet, in welchem Umfang die in Abschnitt 3.3 definierten Anforderungen erfüllt sind. Es werden nur Schlüsselverwaltungssysteme, die symmetrische Schlüssel bereitstellen, analysiert, da in der Praxis meistens symmetrische Algorithmen zum Online-Schutz von Übertragungen eingesetzt werden. Zur Strukturierung werden die existierenden Systeme in drei Kategorien eingeteilt. Jede dieser Kategorien wird in zwei Unterkategorien aufgeteilt:

- Zentrale Verfahren zur Bereitstellung von Gruppenschlüsseln
  Eine einzelner Schlüsselverwaltern als Group Controller/Key Server (GCKS) bezeichnet, koordiniert die Gruppe und stellt den Gruppenschlüssel bereit, d.h. verteilt diesen. Zentrale Schlüsselverwaltungsverfahren können weiter unterteilt werden in Verfahren, die einen statischen bzw. dynamischen Gruppenschlüssel bereitstellen.
- Hierarchische Verfahren zur Bereitstellung von Gruppenschlüsseln
  Zur Verbesserung der Effizienz bei steigenden Nutzeranzahlen wird bei hierarchischen Verfahren die Gruppe in Teilgruppen eingeteilt. Die Teilgruppen werden dann von einem einzelnen Prozess verwaltet. Eine detaillierte Einteilung kann vorgenommen werden, wenn unterschieden wird, ob ein gemeinsamer oder ein unterschiedlicher Gruppenschlüssel für die Teilgruppen bereitgestellt wird.
- Verteilte Verfahren zur Bereitstellung von Gruppenschlüsseln
  Bei verteilten Verfahren ist kein Koordinator fixiert, der die Schlüsselbereitstellung durchführt. Bei einer weiteren Unterteilung dieser Kategorie wird unterschieden, ob der Gruppenschlüssel vereinbart wird, d.h. jeder Nutzer liefert einen Beitrag zum Gruppenschlüssel, oder ob es sich um eine dezentrale Schlüsselverteilung handelt.

3.4.1 Beschreibung zentraler Verfahren zur Bereitstellung von Gruppenschlüsseln

Zunächst werden die zentralen Verfahren zur Bereitstellung von Gruppenschlüsseln vorgestellt. Das Verfahren Pre-distributed Group Key stellt nur einen statischen

Gruppenschlüssel bereit, während alle übrigen Verfahren eine dynamische Schlüsselbereitstellung ermöglichen.

Pre-distributed Group Key

Eine einfache Möglichkeit der Verteilung eines Gruppenschlüssels besteht darin, im Vorfeld der Kommunikation allen einen statischen Schlüssel zu übermitteln. Man spricht dabei von der Methode Pre-distributed Group Key (PGK).

Group Key Management Protocol

Beim Group Key Management Protocol (GKMP) [Har97] wird vom GCKS mittels eines sogenannten Group Key Packets der Group Traffic Encryption Key k_{GTEK} und der Group Key Encryption Key k_{GKEK} verteilt. Während der Schlüssel k_{GTEK} zum Schutz der Nutzdaten eingesetzt wird, dient der Group Key Encryption Key als Hilfsschlüssel beim Schlüsselwechsel. Neben diesen beiden Schlüsseln verfügt ein Nutzer u_ℓ über den Individual Key Encryption Key k_{IKEK_ℓ}, mit $\ell \in \{1,\ldots,U\}$. Beim Beitritt eines Nutzers u_{U+1} zu einer Gruppe mit U Teilnehmern wird die Schlüsselwechselinformation $E(\tilde{k}_{GTEK},\tilde{k}_{GKEK})$, $E(\tilde{k}_{GKEK},k_{GKEK})$, $E(\tilde{k}_{GKEK},k_{ITEK_U})$ verteilt. Diese übermittelt insbesondere den aktuellen Schlüssel $\tilde{k}_{GTEK}$ zum Multicast-Inhaltsschutz der Nutzdaten. Beim Austritt des Nutzer u_U versendet der GCKS die Schlüsselwechselinformation $E(\tilde{k}_{GTEK},\tilde{k}_{GKEK})$, $E(\tilde{k}_{GKEK},k_{ITEK_\ell})$ mit $\ell \in \{1,\ldots,U-1\}$. Aus diesem Grund skaliert das Verfahren nicht für große Gruppen.

Conference Key Agreement

Das Verfahren Conference Key Agreement (CKA) [Boy97] legt fest, dass in einer Gruppe mit U Nutzern jeder Nutzer u_ℓ mit $\ell \in \{2,\ldots,U\}$ seinen zufällig erzeugten Schlüssel k_ℓ offen an alle übrigen Nutzer $\{u_1,\ldots,u_{\ell-1},u_{\ell+1},\ldots,u_U\}$ sendet. Im Gegensatz dazu überträgt der Nutzer u_1, der als GCKS fungiert, seinen zufällig erzeugten Schlüssel k_1 geschützt an die übrigen Nutzer $\{u_2,\ldots,u_U\}$. Um den zufällig erzeugten Schlüssel k_1 an den Nutzer u_ℓ mit $\ell \in \{2,\ldots,U\}$ geschützt zu übertragen, verschlüsselt der Nutzer u_1 diesen mit dem öffentlichen Schlüssel des Schlüsselpaars sk_ℓ,pk_ℓ jedes Nutzers u_ℓ, d.h. u_1 übermittelt dem Nutzer u_ℓ die Information $E(k_1,pk_\ell)$. Das asymmetrische Schlüsselpaar wird für jeden Nutzer im Vorfeld der Schlüsselbereitstellung generiert und der öffentliche Schlüssel dem Nutzer u_1 mitgeteilt. Bei diesem Verfahren wird der Gruppenschlüssel berechnet mit $k_{group}=Mac((Hash(k_2)||Hash(k_3)||\ldots||Hash(k_U)),k_1)$. Beim CKA erfolgt die Schlüsselbereitstellung mittels eines GCKS, allerdings liefert jeder Nutzer einen Beitrag zum Gruppenschlüssel.

Group Domain of Interpretation

Mit dem Verfahren Group Domain of Interpretation (GDOI) [Har03] wird das Internet Security Association and Key Management Protocol (ISAKMP) [Mau98] für die Verwendung in Gruppen konkretisiert. Das im Rahmen des Verfahrens spezifizierte Protokoll wurde zur Etablierung der in Abschnitt 2.5.2 eingeführten GSA unter Verwendung eines zentralen GCKS entwickelt. Die Nachrichten des Protokolls setzen sich aus dem Nachrichtenkopf, dem so genannten ISAKMP-Header, und Nutzlastfeldern zusammen. Das spezifizierte Protokoll verwendet einen 2-Phasen-Ansatz (Abbildung 21). In Phase I wird das bereits existierende Protokoll Internet Key Exchange (IKE) [Kau05] eingesetzt. Mittels dieses Protokolls wird der Kategorie-1-Schlüssel, d.h. der Individualschlüssel, zwischen Nutzer und GCKS vereinbart und der Nutzer authentifiziert. In der Phase 1 kann der Main Mode oder der Aggressive Mode

benutzt werden. Beide haben das gleiche Ergebnis, doch verbirgt der Aggressive Mode die Identität der Kommunikationspartner nicht. Im Main Mode Phase I werden sechs und im Aggressive Mode drei Nachrichten ausgetauscht. Beim Gruppenaustritt wird vom Nutzer eine Nachricht mit dem Nutzlastfeld `Delete` übertragen.

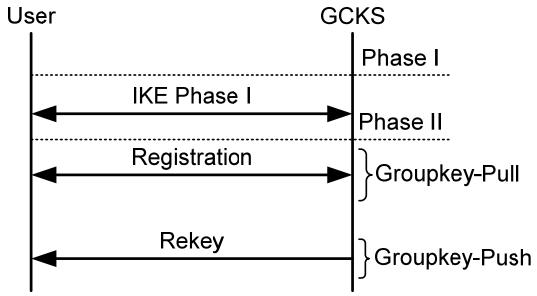

Abbildung 21: Rahmenkonzept von Group Domain of Interpretation

Die Phase II wird in die Protokollabschnitte Groupkey-Push und Groupkey-Pull unterteilt. Im Rahmen des Anmeldevorgangs wird der Protokollabschnitt Groupkey-Pull durchlaufen (Abbildung 22). Dieser aus vier Nachrichten bestehende Protokollabschnitt dient zur Registrierung eines Nutzers und zur Initialisierung des Gruppenschlüsselmanagements. Alle Nachrichten sind durch die in Protokollphase I zwischen Nutzer und GCKS vereinbarten Schlüssel authentisiert und verschlüsselt. Mit der ersten Nachricht des Protokollabschnitts stellt der Nutzer eine Gruppenbeitrittsanfrage. Zur Gewährleistung der Authentizität enthält diese im Feld `Hash(1)` einen MAC. Der im Nutzlastfeld `NONCE` enthaltene Wert trägt dazu bei, die Aktualität des Nachrichtenaustauschs zu gewährleisten. Auf die Anfrage zum Gruppenbeitritt bzw. zur Gruppenregistrierung antwortet der GCKS unter Einsatz des Nutzlastfelds `SA` durch Übermittlung der in der Gruppe verwendeten Algorithmen und Sicherheitsprotokolle sowie einer von ihm generierten Zufallszahl. Der Nutzer antwortet hierauf mit einem MAC (Feld `Hash(3)`) über die beiden Zufallszahlen und dem in der Phase I vereinbarten Schlüssel. Das in der Gruppe verwendete Schlüsselmaterial wird im Nutzlastfeld `KD` erst übermittelt, nach dem der GCKS den Inhalt des Feldes `HASH(3)` verifiziert und damit neben der Zutrittsberechtigung auch die Aktualität der Anfrage überprüft hat. Weiterhin wird mit der Übermittlung des Schlüsselmaterials eine Sequenznummer beim Nutzer initialisiert, die diesen in die Lage versetzt, eine Übertragungswiederholung der Groupkey-Push-Nachricht zu erkennen. Bei Bedarf, z.B. bei Änderung der Gruppengröße/-zusammensetzung oder beim Ablauf der Gültigkeitsdauer des TEK, können vom GCKS die KEKs (Kategorie-2-SA) und die TEKs (Kategorie-3-SA) erneuert werden. Hierzu wird die Groupkey-Push-Nachricht verwendet. Diese besitzt zum Transport der Aktualisierungsinformation für Algorithmen und Schlüssellebensdauer das Nutzlastfeld `SA` sowie das Feld `KD` zur Übermittlung des neuen Schlüsselmaterials. Das GCKS verwaltet eine Sequenznummer, die jedes Mal erhöht wird, wenn eine Groupkey-Push-Nachricht gesendet wird. Der aktuelle Wert der Sequenznummer ist im Feld `SEQ` der Groupkey-Push-Nachricht enthalten. Dieses erlaubt den Gruppenteilnehmern, festzustellen, ob die Nachricht nicht eine Wiederholung einer bereits empfangenen Nachricht ist. Die Groupkey-Push-Nachricht wird vom GCKS mit einer digitalen Signatur versehen (Feld `SIG`). Dadurch können die Empfänger überprüfen, dass die Nachricht vom GCKS und nicht von einem anderen Nutzer gesendet wurde.

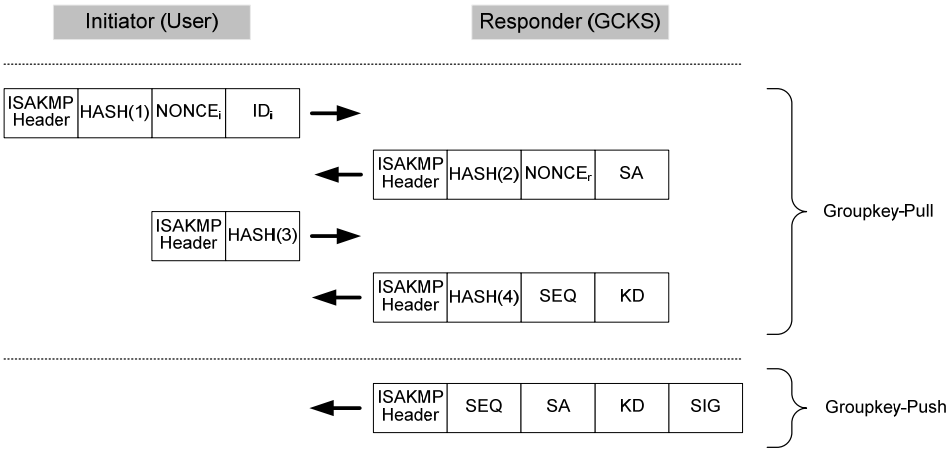

Abbildung 22: Protokollabschnitt Groupkey-Pull und Groupkey-Push des Verfahrens GDOI

Group Secure Association Key Management Protocol

Das Group Secure Association Key Management Protocol (GSAKMP) [Har06] ist dem Verfahren Group Domain of Interpretation sehr ähnlich. Der Unterschied besteht darin, dass das Konzept Group Domain of Interpretation das Internet Security Association and Key Management Protocol (ISAKMP) für die Verwendung in Gruppen erweitert, während GSAKMP eigene Nachrichtentypen und Formate verwendet.

Shared Secret Key Encryption Key

Das Verfahren Shared Secret Key Encryption Key (SSKEK) [Chu02] setzt voraus, dass jeder Nutzer u_ℓ über einen Individual Key Encryption Key k_{IKEK_ℓ} verfügt, der nur ihm und dem GCKS bekannt ist. Wird vom Nutzer u_ℓ eine Nachricht m an eine Gruppe mit U Teilnehmern versandt, so wird diese mit einem zufällig generierten Schlüssel k′ verschlüsselt. Dieser wird mit k_{IKEK_ℓ} verschlüsselt und zusammen mit der Nachricht m versandt, d.h. der Nutzer u_ℓ versendet $E(m,k'),E(k',k_{IKEK_\ell})$ an die Gruppe inklusive des GCKS. Erhält der GCKS diese Nachricht, versendet er eine `ValidationMessage`, die $E(k',k_{IKEK_i})$ mit $i \in \{1,\ldots,\ell\text{-}1,\ell\text{+}1,\ldots,U\}$ enthält. Die Teilnehmer können die Nachricht m entschlüsseln, indem sie den Schlüssel k′ aus der `ValidationMessage` entnehmen. Dadurch, dass die Teilnehmer diese zur Entschlüsselung benötigen, können Nachrichten erst verzögert ausgeliefert werden. Außerdem skaliert das Verfahren nicht, da die Nachrichtengröße der `ValidationMessage` bei linear steigender Gruppengröße ebenfalls linear steigt.

Logical Key Hierarchy

Beim Verfahren Logical Key Hierarchy (LKH) [Won98] fasst ein GCKS die Nutzer zu Teilgruppen zusammen und weist diesen mittels eines Schlüsselbaums Hilfsschlüssel zu. Durch die Hilfsschlüssel ist es möglich, den in der Wurzel abgelegten Gruppenschlüssel sicher und effizient zu verteilten, indem dieser mit den Hilfsschlüsseln verschlüsselt übermittelt wird. Eine formalere Funktionsbeschreibung enthält der Abschnitt 4.4.2, da dieses Verfahren auch Bestandteil des in dieser Arbeit entwickelten Schlüsselmanagements ist. An dieser Stelle wird die Idee des Verfahrens LHK an einfachen Beispielen verdeutlicht. In Abbildung 23, links wird die Bildung von Teilgruppen für die Nutzer $\{u_1,u_2,u_3,u_5,u_6,u_7,u_8\}$ und eine Zuweisung von Hilfsschlüsseln mittels eines Schlüsselbaums vom Grad d=2 illustriert. Zum Beispiel werden die Nutzer $\{u_5,u_6,u_7,u_8\}$ zu einer Teilgruppe zusammengefasst und es wird ihnen der Hilfsschlüssel $k_{1,1}$ zugewiesen. Die Verwendung

eines anderen Baumgrads ist auch möglich. Nun wird der Ablauf des LKH beim Gruppenbeitritt erläutert. Abbildung 23, links zeigt die Platzierung des neuen Nutzers u_4 im Schlüsselbaum infolge des Beitritts. In dem dargestellten Beispiel wird der neue Nutzer an den Knoten $v_{2,1}$ angehängt und ihm der Schlüssel $k_{3,3}$ zugeordnet. Die bei Schlüsselbaumaktualisierung neu erzeugten Hilfsschlüssel $\tilde{k}_{1,0}$, $\tilde{k}_{2,1}$ und der neue Gruppenschlüssel $\tilde{k}_{0,0}$ werden durch $E(\tilde{k}_{0,0},k_{0,0})$, $E(\tilde{k}_{1,0},k_{1,0})$, $E(\tilde{k}_{2,1},k_{3,2})$, $E(\tilde{k}_{0,0},k_{3,3})$, $E(\tilde{k}_{1,0},k_{3,3})$, $E(\tilde{k}_{2,1},k_{3,3})$ an die bisherigen und den neuen Teilnehmer übermittelt (vgl. Abbildung 23, Mitte). In Abbildung 23, rechts ist dargestellt, wie der Schlüsselbaum aktualisiert werden muss, wenn der Nutzer u_4 die Gruppe verlässt. In diesem Fall müssen im Schlüsselbaum die Schlüssel $\tilde{k}_{1,0}$ und $\tilde{k}_{0,0}$ aktualisiert werden. Die in der Gruppe verbleibenden Nutzer werden über den aktualisierten Gruppenschlüssel informiert, ohne dass der ehemalige Teilnehmer in dessen Besitz gelangt. Hierzu werden die aktualisierten Schlüssel mit den in den Kindern abgelegten Schlüsseln verschlüsselt, d.h. $E(\tilde{k}_{0,0}, \tilde{k}_{1,0})$, $E(\tilde{k}_{1,0},k_{1,1})$, $E(\tilde{k}_{1,0},k_{2,0})$, $E(\tilde{k}_{1,0},k_{3,2})$, zugesandt.

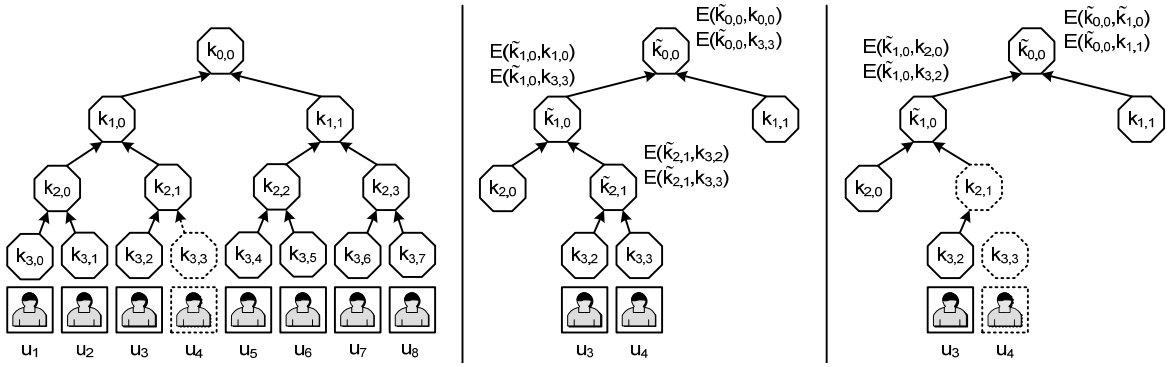

Abbildung 23: Einfügen des Nutzers u_4 in den Schlüsselbaum beim Beitritt (links) und die erforderlichen Verschlüsselungen (Mitte) sowie die erforderlichen Verschlüsselungen beim Austritt des Nutzers u_4 (rechts) beim Verfahren LKH

<u>One-way Function Tree</u>

Das Verfahren One-way Function Tree (OFT) [She03] verwendet einen Schlüsselbaum vom Grad d=2. Bei diesem werden ähnlich wie beim Verfahren Logical Key Hierarchy mittels eines Schlüsselbaums verwaltete Hilfsschlüssel zur effizienten Verteilung des Gruppenschlüssels verwendet. Allerdings werden die Hilfsschlüssel rekursiv aus den in den Kindern gespeicherten Schlüsseln berechnet und nicht nur verteilt, um die Anzahl der übertragenen Hilfsschlüssel zu vermindern. Hierzu wird die Berechnungsvorschrift $k_{\ell,p}=f(g(k_{\ell+1,2p}),g(k_{\ell+1,2p+1}))$ mit der Einwegfunktion g(x) und der Verknüpfungsfunktion $f(x_1,x_2)$ verwendet. Genau wie beim Verfahren Logical Key Hierarchy kennt ein Nutzer alle Schlüssel auf dem Pfad von dem ihm zugeordneten Blatt bis zur Wurzel des Schlüsselbaums. Allerdings muss er zusätzlich die mit $g(k_{\ell,p})$ abgeleiteten Schlüssel der Geschwister aller Knoten auf dem Pfad zur Wurzel kennen. Wird zum Beispiel der Baum in Abbildung 23 links bei diesem Verfahren zur Schlüsselverwaltung verwendet, so kennt der Nutzer u_8 die Schlüssel $\{k_{3,7},g(k_{3,6}),k_{2,3},g(k_{2,2}),k_{1,1},g(k_{1,0}),k_{0,0}\}$. Zur Illustration wird der Beitritt des Nutzers u_3 genauer erläutert. In dem dargestellten Beispiel wird der neue Nutzer an den Knoten $v_{2,1}$ angehängt und ihm der Schlüssel $k_{3,3}$ zugeordnet. Die bei der Schlüsselbaumaktualisierung neu erzeugten Hilfsschlüssel $\tilde{k}_{1,0}$, $\tilde{k}_{2,1}$ bzw. der neue Gruppenschlüssel $\tilde{k}_{0,0}$ werden vom GCKS

durch $E(g(\tilde{k}_{1,0}),k_{1,1})$, $E(g(\tilde{k}_{2,1}),k_{2,0})$, $E(g(\tilde{k}_{3,2}),k_{3,3})$ an die bisherigen und den neuen Teilnehmer übermittelt. Durch $\tilde{k}_{0,0}=f(g(\tilde{k}_{1,0}),g(k_{1,1}))$, $\tilde{k}_{1,0}=f(,g(k_{2,0}),g(\tilde{k}_{2,1}))$ kann zum Beispiel der Nutzer u_1 den neuen Gruppenschlüssel berechnen.

One-way Function Chain Tree

Das Verfahren One-way Function Chain Tree (OFCT) [Can99] ist der Methode OFT sehr ähnlich. Es nutzt allerdings einen Pseudozufallszahlengenerator zur Berechnung der Hilfsschlüssel.

Hierarchical a-ary Tree with Clustering

Beim Mechanismus Hierarchical a-ary Tree with Clustering (HTC) [Ber01] wird für die Schlüsselbereitstellung mittels Schlüsselbaum eine Verminderung des Speicherbedarfs zu Lasten des Kommunikationsaufwands erzielt. Hierzu wird eine Gruppe, bestehend aus U Nutzern in Cluster der Größe M unterteilt. Jedem Cluster wird ein Blatt des Baums zugewiesen (vgl. Abbildung 24). Der in diesem Blatt abgelegte Schlüssel wird als Cluster Key Encryption Key bezeichnet. Jeder Nutzer kennt bei diesem Verfahren den seinem Cluster zugewiesen Cluster Key Encryption Key sowie alle in Knoten auf dem Pfad zur Wurzel abgelegten Schlüssel. Zusätzlich verfügt jeder Nutzer u_i über einen Individualschlüssel k_i, der nur ihm und dem GCKS bekannt ist. In Abbildung 24 sind die Nutzer in vier Cluster eingeteilt. Die zu einem Cluster zusammengefassten Nutzer u_1, u_2 und u_3 kennen den Cluster Key Encryption Key $k_{2,0}$. Verlässt zum Beispiel der Nutzer u_3 die Gruppe, wird zur Aktualisierung des Gruppenschlüssels $E(\tilde{k}_{0,0},\tilde{k}_{1,0})$, $E(\tilde{k}_{0,0},k_{1,1})$, $E(\tilde{k}_{1,0},\tilde{k}_{2,0})$, $E(\tilde{k}_{1,0},k_{2,1})$ sowie des Cluster Key Encryption Key $E(\tilde{k}_{2,0},k_1)$, $E(\tilde{k}_{2,0},k_2)$ versandt.

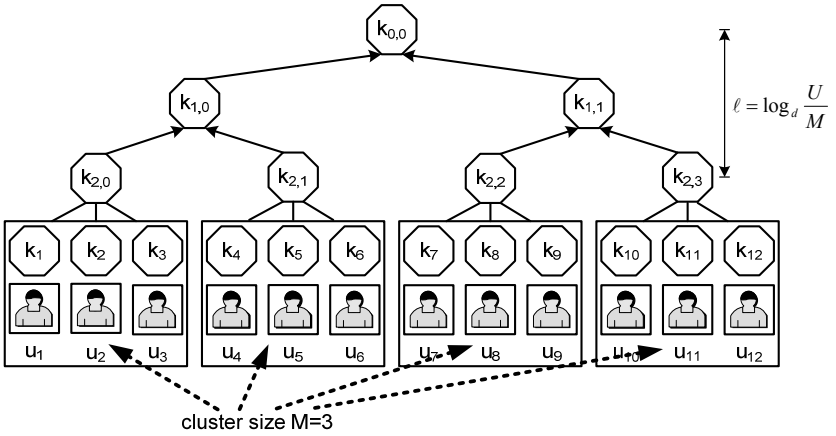

Abbildung 24: Zuordnung der Cluster zu den Blättern des Schlüsselbaums beim Verfahren HTC

Pre-Positioned Secret Sharing

Das Verfahren Pre-Positioned Secret Sharing (PSS) [Del06] basiert auf Shamir's Secret Sharing Scheme. Beim PPS werden vom GCKS jedem Nutzer (m-1) Gruppenschlüsselteile s_i, so genannte Pre-Positioned Shares, verschlüsselt übermittelt. Der Gruppenschlüssel kann nach der unverschlüsselten Übermittlung des m-ten Gruppenschlüsselanteils ermittelt werden. Dieses wird als Activation Shares bezeichnet. Zur Verwaltung der Gruppenschlüsselanteile wird ein Schlüsselbaum eingesetzt. Das Verfahren wurde zur Gruppenschlüsselbereitstellung in zellularen Netzwerken, z.B. dem Global System for Mobile Communication (GSM), entwickelt. Bestandteil des Verfahrens ist deshalb ein Mechanismus zum Schlüsselwechsel

beim Wechsel einer Zelle des Netzwerks. Das Verfahren wird nun anhand eines Beispiels verdeutlicht. In Abbildung 25 ist ein Schlüsselbaum für das Verfahren PSS mit neun Nutzern dargestellt. In jedem Knoten v_i ist ein Satz aus acht Pre-Positioned Shares s_i abgespeichert. Mit k_i wird der Schlüssel bezeichnet, der aus dem Satz s_i des Knoten v_i nach Erhalt des Activation Shares berechnet werden kann. Beim Austritt des Nutzers u_9 übermittelt der GCKS verschlüsselt die neuen Sätze Pre-Positioned Shares, d.h. $E(\tilde{s}_0,k_1)$, $E(\tilde{s}_0,k_2)$, $E(\tilde{s}_0,\tilde{k}_3)$, $E(\tilde{s}_3,k_{10})$, $E(\tilde{s}_3,k_{11})$, sowie ein unverschlüsseltes Activation Share.

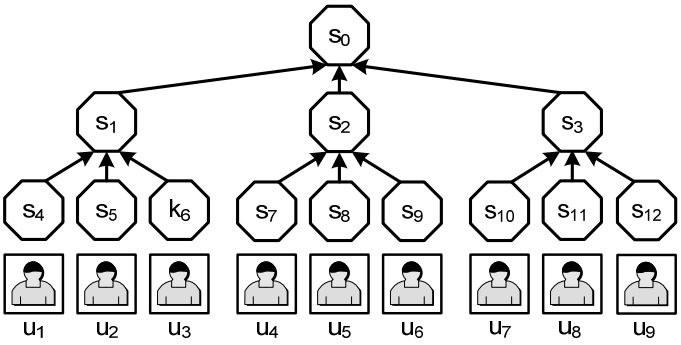

Abbildung 25: Schlüsselbaum für das Verfahren Pre-Positioned Secret Sharing

Efficient Large Group Key

Das Verfahren Efficient Large Group Key (ELK) [Son01] wurde entwickelt, um speziell in sehr großen Gruppen eine Schlüsselbereitstellung zu ermöglichen. Auch bei diesem Verfahren werden die Nutzer zu Teilgruppen zusammengefasst und ihnen mittels eines Schlüsselbaums Hilfsschlüssel zur effizienten Gruppenschlüsselbereitstellung zugewiesen. Im Prinzip basiert ELK auf dem Verfahren Logical Key Hierarchy. Allerdings wurden zur Verbesserung der Effizienz und Zuverlässigkeit in großen Gruppen die Mechanismen Evolving Tree Protocol, Time-structured Tree Protocol, Entropy Injection Key Update und Very Important Bit ergänzt. Das Evolving Tree Protocol verbessert die Effizienz des Gruppenbeitritts, in dem die Schlüssel auf dem Pfad von der Position des neuen Nutzers durch Anwendung einer Prüfsummenfunktion auf die Schlüssel aktualisiert werden. Das Time-structured Tree Protocol ermöglicht eine Effizienzsteigerung, wenn der Zeitpunkt, an dem ein Nutzer die Gruppe verlässt, bekannt ist. Zu diesem Zweck teilt der GCKS die Dauer der Kommunikation in Zeitscheiben Δt ein. Allen Nutzern, die die Gruppe in der Zeitscheibe Δt_i verlassen, wird ein Knoten aus dem Teilbaum ΔKT_i zugewiesen. In dieser Zeitscheibe ist der in der Wurzel des Teilbaums ΔKT_i gespeicherte Schlüssel der Gruppenschlüssel. Nutzer, die die Gruppe in der Zeitscheibe Δt_{i+x} ($x>0$) verlassen, wird ein Knoten aus dem Teilbaum ΔKT_{i+x} zugewiesen. Außerdem wird der Teilbaum ΔKT_{i+1} jeweils an ein Blatt des Teilbaums ΔKT_i angehängt. Mit einer derartigen Baumstruktur ist ein effizienter Schlüsselwechsel möglich, wenn zu Beginn der Zeitscheibe Δt_ℓ der GCKS den kompletten Teilbaum ΔKT_ℓ abschneidet. In dem Mechanismus Entropy Injection Key Update ist das so genannte Hint Protocol enthalten. Mit diesem wird eine Effizienzverbesserung erzielt, in dem die Größe der übertragenen Hilfsschlüssel beim Schlüsselwechsel halbiert wird. Zur robusten Schlüsselübermittlung wird Übertragungswiederholung verwendet. Allerdings wird die Größe der erneut übertragenen Nachricht vermindert, indem vom Mechanismus Very Important Bit nur Schlüssel zur erneuten Übertragung ausgewählt werden, die von einer Vielzahl von Nutzern zur Ermittlung des Gruppenschlüssels benötigt werden.

Centralized Flat Table

Zur Gruppenschlüsselverwaltung setzt der GCKS beim Verfahren Centralized Flat Table (CFT) [Wal99] eine Tabelle ein. Diese enthält den Gruppenschlüssel, der bei diesem Verfahren mit k_{TEK} bezeichnet wird, sowie 2·w Hilfsschlüssel. Der Buchstabe w repräsentiert bei diesem Verfahren die Anzahl der Bits der Nutzerkennzeichnung. Jedem Bit der Nutzerkennzeichnung sind zwei Schlüssel zugeordnet. Bei diesem Verfahren werden die Hilfsschlüssel mit $k_{u/v}$ bezeichnet, wobei der erste Index u die Position des Bits mit $u \in \{0,\ldots,w\}$ beschreibt und der zweite Index v mit $v \in \{0,1\}$ den Wert des Bits enthält. Durch die Hilfsschlüssel ist es möglich, den in der Tabelle abgelegten Gruppenschlüssel k_{TEK} sicher und effizient zu verteilen, indem er mit den Hilfsschlüsseln verschlüsselt übermittelt wird. Das Verfahren wird nun durch ein Beispiel illustriert.

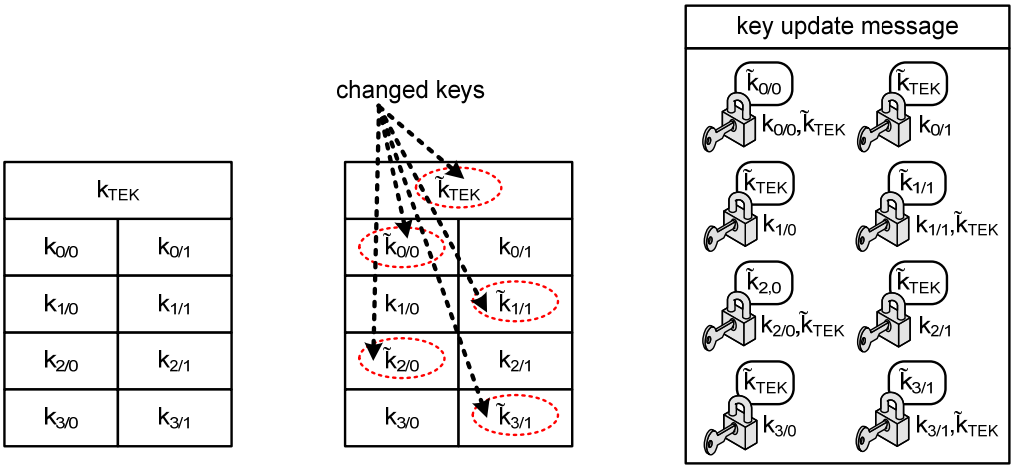

Abbildung 26: Schlüsseltabelle (links) und deren Aktualisierung beim Austritt des Nutzers u_5 (Mitte) sowie die Nachricht für den Schlüsselwechsel (rechts) beim Verfahren CFT

In Abbildung 26, links ist die Schlüsseltabelle dargestellt, bei der vier Bits zur Nutzerkennzeichnung verwendet werden. Bei dieser Größe der Nutzerkennzeichnung können 16 Nutzer verwaltet werden. In diesem Fall gilt für die Indizes der Schlüsselbezeichnung $u \in \{0,1,2,3\}$ und $v \in \{0,1\}$. Der Nutzer u_5 mit der Nutzerkennzeichnung $Id_{u_5}=0101$ ist bei diesem Beispiel im Besitz der Schlüssel k_{TEK}, $k_{0/0}$, $k_{1/1}$, $k_{2/0}$ und $k_{3/1}$. Verlässt dieser Nutzer die Gruppe, so müssen alle mit Rot markierten Schlüssel der Abbildung 26, Mitte erneuert werden. Zu diesem Zweck versendet der GCKS die in Abbildung 26, rechts dargestellte Nachricht. Diese enthält den neuen Gruppenschlüssel verschlüsselt mit dem gültigen Hilfsschlüssel. Weiterhin enthält die Nachricht die neuen Hilfsschlüssel verschlüsselt mit den alten Hilfsschlüssel und dem neuen Gruppenschlüssel $\tilde{k}_{TEK}$.

MARKS

MARKS [Bri99] ist ein Verfahren zur Schlüsselbereitstellung in einer Gruppe bei dem die Nutzer diese zu vorher bekannten Zeitpunkten verlassen. Hierzu wird von einem GCKS die Dauer der Kommunikation in M Zeitscheiben Δt_i mit $i \in \{1,\ldots,M\}$ eingeteilt. Jede dieser Zeitscheibe Δt_i wird ein Gruppenschlüssel k_i zugewiesen. Die Erzeugung der Schlüssel und deren Zuordnung zu einer Zeitscheibe erfolgt mittels eines so genannten Seed-Baums. Dieser besitzt den Grad d=2 und mindestens so viele Blätter wie Zeitscheiben. In jedem Knoten des Baums wird ein Seed $s_{\ell,p}$ gespeichert. In der Wurzel des Baums wird ein zufällig generierter

Seed gespeichert. Außerdem werden mit Hilfe von zwei verschiedenen Prüfsummenfunktionen der Seed des linken Nachfolgers mit $s_{\ell+1,2p}=Hash_L(s_{\ell,p})$ und der des rechten Nachfolgers mit $s_{\ell+1,2p+1}=Hash_R(s_{\ell,p})$ berechnet. Aus den in den Blättern abgelegten Seeds werden die Schlüssel k_i, $i\in\{1,\ldots,M\}$ ermittelt, indem sie als Eingabe für einen Zufallszahlengenerator Prng(x) verwendet werden. Ein Schlüssel k_i wird berechnet gemäß $k_i=Prng(s_{\ell,p})$ mit $p=0,\ldots,p^*_{max}$ und $i=p+1$. Beim Gruppenbeitritt übermittelt ein Nutzer dem GCKS seine Teilnahmedauer. Aus den daraufhin vom GCKS enthaltenen Seeds ermittelt jeder Nutzer eigenständig die gültigen Gruppenschlüssel. Als Beispiel wird die Einteilung der Kommunikationsdauer bei der Initialisierung in acht Zeitscheiben betrachtet. Vom GCKS wird der in Abbildung 27 dargestellte Seed-Baum erzeugt. Dem Nutzer u_1 mit der in der Abbildung dargestellten Teilnahmedauer werden die Seeds $s_{3,3}$ und $s_{1,1}$ mitgeteilt. Aus diesen kann er unter Anwendung von $Hash_L(s_{\ell,p})$, $Hash_R(s_{\ell,p})$ und $Prng(s_{\ell,p})$ die zur Entschlüsselung des Datenverkehrs notwendige Schlüsselmenge $\{k_4,k_5,k_6,k_7,k_8\}$ berechnen.

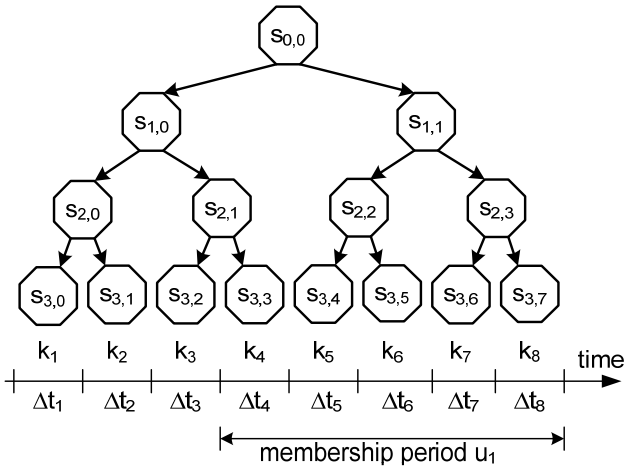

Abbildung 27: Seed-Baum zur Ermittlung der Gruppenschlüssel beim Verfahren MARKS

Broadcast Encryption

Bei dem Verfahren Broadcast Encryption (BE) [Fia93] wird für jede mögliche Teilgruppe B der Gruppe G ein Hilfsschlüssel k_B erzeugt. Dieser Schlüssel wird an alle Teilnehmer verteilt, die nicht zu dieser Teilgruppe gehören, also an alle Teilnehmer $u\in G\backslash B$. Soll ein Gruppenschlüssel erzeugt werden, den die Teilgruppe T nicht ermitteln kann, so wird die XOR-Summe aller Hilfsschlüssel k_B, die keinen Teilnehmer aus der Teilgruppe T enthalten, gebildet:

$$k_{group} = \bigoplus_{B\subset U\backslash T} k_b$$

Diese Gruppenschlüssel können alle Teilnehmer den Gruppenschlüssel berechnen, da sie alle benötigten k_B kennen. Es ist außerdem nicht nötig, eine gesonderte Nachricht zur Verteilung des Gruppenschlüssels zu versenden. Ein Sender fügt der Nutzdatenübertragung lediglich eine Bezeichnung für die aktuelle Teilgruppe hinzu. Ein derartiges Verfahren wird als k-resilient bezeichnet, da keine Teilgruppe $S\subset U$ mit $|S|<k$ in den Besitz des Gruppenschlüssels gelangen kann. Allerdings ist es einer Teilgruppe $|S'|\geq k$ durchaus möglich, in den Besitz des Gruppenschlüssels zu gelangen. Im nachfolgenden Beispiel wird ein Verfahren, das 1-resilient ist, dargestellt. Hierzu werden an jeden Nutzer, wie in Abbildung 28 dargestellt,

fünf Schlüssel verteilt. Als Gruppenschlüssel wird zunächst der Schlüssel k_6 verwendet, d.h. $k_{group}=k_6$. Nach dem Austritt des Nutzers u_5 wird der neue Gruppenschlüssel $\tilde{k}_{group}=k_1 \oplus k_2 \oplus k_3 \oplus k_4 \oplus k_5$ eingesetzt.

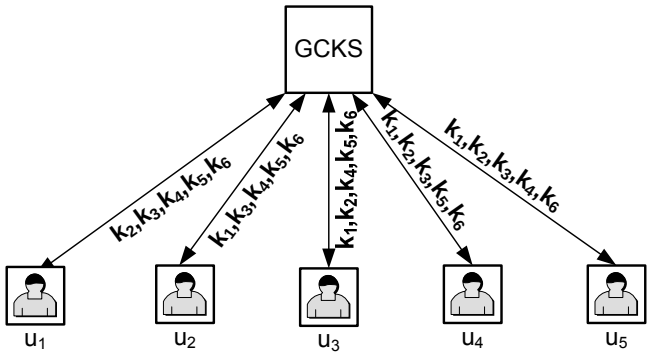

Abbildung 28: 1-resilient Broadcast Encryption

3.4.2 Bewertung zentraler Verfahren zur Bereitstellung eines Gruppenschlüssels

Der Nachteil des Verfahrens Pre-distributed Group Key (PGK) besteht darin, dass alle Teilnehmer vor der Kommunikation mit dem Schlüssel versorgt werden müssen und damit die Gruppengröße festgelegt ist. Da alle Beteiligten im Besitz des statischen Gruppenschlüssels sind, ist beim Ausschluss eines kompromittierten Nutzers eine Schlüsselgeheimhaltung (Forderung 1, Tabelle 6) nicht möglich. Mit den Verfahren Group Key Management Protocol (GKMP), Conference Key Agreement (CKA) und Shared Secret Key Encryption Key (SSKEK) kann ein Gruppenschlüssel bereitgestellt werden, so dass auch in dynamischen Gruppen eine Schlüsselgeheimhaltung (Forderung 1, Tabelle 6) möglich ist. Die Verfahren enthalten eine Beitrittsprozedur, die eine Zugriffskontrolle (Forderung 2, Tabelle 6) sicherstellt. Allerdings zeigen diese Verfahren auf Grund einer linear steigenden Nachrichtenanzahl bzw. Nachrichtengröße bei der Übertragung der Schlüsselwechselinformationen geringe Skalierbarkeit (Forderung 3, Tabelle 6) bei steigender Gruppengröße. Zusätzlich besitzt das Verfahren Shared Secret Key Encryption Key den Nachteil einer verzögerten Schlüsselauslieferung. Mit den baum-basierten Schlüsselbereitstellungsverfahren Logical Key Hierarchy (LKH), One-way Function Tree (OFT), One-way Function Chain Tree (OFCT), Hierarchical a-ary Tree with Clustering (HTC) und Pre-Positioned Secret Sharing (PSS) lassen sich ebenfalls in dynamischen Gruppen die Sicherheitsvorschriften Group Key Secrecy, Forward Secrecy und Backward Secrecy (Forderung 1, Tabelle 6) erfüllen und eine Zugriffskontrolle (Forderung 2, Tabelle 6) realisieren. Weiterhin sind diese drei Verfahren in kleinen Gruppen einsetzbar und besitzen eine gute Effizienz der ausgetauschten Schlüsselwechselinformationen auch bei steigender Gruppengröße (Forderung 3, Tabelle 6). Allerdings ist beim HTC die Effizienz des Schlüsselwechsels zugunsten eines verminderten Speicherbedarfs etwas verschlechtert. Wie in [Yan01] gezeigt wird, lassen sich Mechanismen zur Verarbeitung von Mehrfachanfragen (Forderung 6, Tabelle 6) leicht integrieren. Die in diesen Verfahren fehlende IPSec-Adaption (Forderung 8, Tabelle 6) kann ebenfalls durch Kombination mit den Verfahren Group Secure Association Key Management Protocol (GSAKMP) oder Group Domain of Interpretation (GDOI) leicht erfüllt werden. Auf diese Möglichkeit der Kombination mit anderen Verfahren

wird bei den Verfahren GSAKMP und GDOI hingewiesen. Die Verfahren Efficient Large Group Key (ELK), MARKS und Broadcast Encryption (BE) bilden einen Kompromiss zwischen Erfüllung der Sicherheitsvorschriften und der Bereitstellung eines Gruppenschlüssels in sehr großen Gruppen. Die Verfahren ELK und MARKS erfüllen auf Grund ihres periodischen Schlüsselwechsels nicht strikt die Forderung nach Schlüsselgeheimhaltung (Forderung 1, Tabelle 6). Diese wird ebenfalls von dem Verfahren Broadcast Encryption (BE) nicht eingehalten, da eine Kooperation aus mehreren ausgetretenen Nutzern den Gruppenschlüssel ermitteln kann. Der hauptsächliche Nachteil aller in diesem Abschnitt vorgestellten Schlüsselbereitstellungsverfahren besteht darin, dass sie durch den zentralen GCKS einen Single Point of Failure enthalten und somit die für den Einsatzbereich notwendige Reparierbarkeit (Forderung 5, Tabelle 6) nicht erfüllen. Außerdem kann der GCKS die Gruppe nicht verlassen. Die Bewertung der zentralen Verfahren zur Bereitstellung von Gruppenschlüsseln ist in Tabelle 8 zusammengefasst.

Verfahren/ Anforderung	1 Schlüssel-geheim-haltung	2 Zugriffs-kontrolle	3 Skalier-barkeit	4 Robuste Schlüssel-über-mittlung	5 Reparier-barkeit	6 Mehrfach-anfragen	7 EMCON	8 IPSec-Adaption	9 Realzeit
PGK	-	+	-	0	-	-	+	+	+
GKMP	+	+	-	0	-	0	0	+	+
CKA	+	+	-	0	-	0	-	0	+
GDOI	+	+	0	0	-	0	0	+	+
GSAKMP	+	+	0	0	-	0	0	+	+
SSKEK	+	+	-	0	-	0	0	0	+
LKH	+	+	+	0	-	+	0	0	+
OFT	+	+	+	0	-	0	0	0	+
OFCT	+	+	+	0	-	0	0	0	+
HTC	+	+	+	0	-	0	0	0	+
PSS	+	+	+	0	-	0	0	0	+
ELK	-	+	+	+	-	+	0	0	+
CFT	+	+	+	0	-	0	0	0	+
MARKS	-	+	+	+	-	0	+	0	+
BE	-	+	+	+	-	0	+	0	+

+ erfüllt 0 prinzipiell erfüllbar - nicht erfüllt ? unklar

Tabelle 8: Bewertung der zentralen Verfahren zur Bereitstellung von Gruppenschlüsseln

3.4.3 Beschreibung hierarchischer Verfahren zur Bereitstellung von Gruppenschlüsseln

Basiskonzept aller hierarchischen Verfahren zur Bereitstellung von Gruppenschlüsseln ist die Aufteilung der zu verwaltenden Gruppe in Teilgruppen. Aus diesem Grund werden hierarchische Schlüsselbereitstellungsverfahren unterteilt in Verfahren, die in allen gebildeten Teilgruppen den gleichen Gruppenschlüssel bereitstellen und in solche, die einen unterschiedlichen Gruppenschlüssel zur Verfügung stellen. Begonnen wird die Darstellung

existierender hierarchischer Verfahren mit denen, die in allen Teilgruppen den gleichen Schlüssel verteilen.

Scalable Multicast Key Distribution

Das Verfahren Scalable Multicast Key Distribution (SMKD) wurde in Abschnitt 2.6.1 unter dem Aspekt des Schutzes des Multicast-Verteilbaums vorgestellt. Wie bereits erwähnt, kann mit dem Konzept auch eine Schlüsselbereitstellung für den Schutz von Multicast-Daten vorgenommen werden. Fordert ein Nutzer den Empfang von Multicast-Datenverkehr bei dem für ihn zuständigen Router an, erhält er von diesem den Gruppenschlüssel. Ist der Router noch nicht Bestandteil des Multicast-Verteilbaums, führt er, wie in Abschnitt 2.6.1 beschrieben, eine Gruppenmeldung durch. Ein Schlüsselwechsel infolge eines Gruppenaustritts kann allerdings nur über einen Neuaufbau der Gruppe durchgeführt werden.

Inter-Domain Group Key Management

Das Verfahren Inter-Domain Group Key Management (IGKMP) [Dec01] legt administrative Zonen fest, in denen jeweils ein Area Key Distributor (AKD) für die Schlüsselbereitstellung verantwortlich ist. Weiterhin wird zur Schlüsselbereitstellung ein Domain Key Distributor (DKD) benötigt, der alle Area Key Distributor koordiniert. Jeder Area Key Distributor verwendet zum Datenaustausch mit den von ihm verwalteten Nutzern eine eigene Multicast-Gruppe. Der Area Key Distributor AKD_ℓ verwendet zum Beispiel die Multicast-Gruppe `LocalAreaGroup`$_\ell$. Der Domain Key Distributor sowie alle AKD gehören der Multicast-Gruppe `AllKDGroup` an. Nach einer Änderung der Zusammensetzung der Gruppe übermittelt der Domain Key Distributor den neuen Gruppenschlüssel $\tilde{k}_{group}$ mit $E(\tilde{k}_{group},k_{KD})$ an alle Area Key Distributor unter Verwendung der Multicast-Gruppe `AllKDGroup`. Anschließend übermittelt jeder AKD_ℓ den Gruppenschlüssel $\tilde{k}_{group}$ durch den Versand von $E(\tilde{k}_{group},k_\ell)$ an die von ihm verwalteten Nutzer. In Abbildung 29 ist ein Beispiel dargestellt, bei dem drei Area Key Distributor zur Schlüsselverwaltung eingesetzt werden.

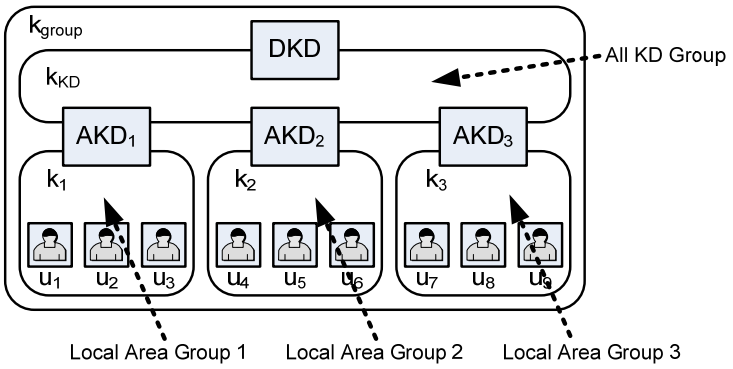

Abbildung 29: Elemente des Verfahrens Inter-Domain Group Key Management

Group Key Management with Network Mobility

Das Verfahren Group Key Management with Network Mobility (GKMNM) [Sun05] ist vergleichbar mit dem Verfahren Inter-Domain Group Key Management. Auch bei diesem werden administrative Zonen festgelegt, die von so genannten Area Group Controller verwaltet werden. Bei diesem Konzept werden die Gruppenschlüssel vom Domain Group Controller mittels der Area Group Controller den Nutzern zur Verfügung gestellt. Die Mobilität von Nutzern wird durch das Delayed Network Mobility Rekeying Scheme

(DNMRS) und das Mobile Group Controller Scheme (MGCS) berücksichtigt. Der erste Mechanismus beinhaltet, dass die Area Group Controller bei mobilen Nutzern, die die administrative Zone wechseln, erst nach einer Verzögerung einen Schlüsselwechsel durchführen. Der zweite Mechanismus sieht vor, dass mobile Nutzer zu einer administrativen Zone zusammengefasst werden, die von einem Mobile Group Controller verwaltet wird.

KRONOS

Ähnlich wie beim IGKMP definiert das Konzept KRONOS [Set00] administrative Zonen, in denen jeweils ein Area Key Distributor für die Schlüsselbereitstellung verantwortlich ist. Diese führen in periodischen Zeitintervallen jeweils in der von ihnen verwalteten Zone einen Wechsel des Gruppenschlüssels durch. Damit in allen Zonen gleichzeitig der Schlüssel gewechselt wird, müssen diese über eine Zeitsynchronisation zum Beispiel mittels des Network Time Protocols (NTP) [Mil92] verfügen. Jeder Area Key Distributor ermittelt rekursiv den im folgenden Zeitintervall gültigen Gruppenschlüssel. Hierzu verteilt der Domain Key Distributor den Hauptschlüssel k_{master} sowie den im Zeitintervall t_0 gültigen Gruppenschlüssel k_0. Der im Zeitintervall $t_{\ell+1}$ gültige Gruppenschlüssel $k_{group_{\ell+1}}$ wird aus dem im Zeitintervall t_ℓ gültigen Gruppenschlüssel k_{group_ℓ} berechnet mit $k_{group_{\ell+1}}=E(k_{group_\ell},k_{master})$.

Dual Encryption Protocol

Das Verfahren Dual Encryption Protocol (DEP) [Don00] schlägt die Bildung von Teilgruppen SG_i und deren Verwaltung durch einen so genannten Teilgruppenverwalter SGM_i vor. Zusätzlich existiert noch ein Key Server KS. In Tabelle 9 sind die gemeinsamen Schlüssel dieser Komponenten dargestellt. Ein neuer Gruppenschlüssel zum Nutzdatenschutz k_{TEK} wird vom Key Server zweimal verschlüsselt an jeden Teilgruppenverwalter versandt, d.h. SGM_i erhält $E(E(k_{TEK},k2_i),k3_i)$. Jeder Teilgruppenverwalter entschlüsselt das Paket mit dem nur ihm und dem Key Server bekannten Schlüssel, d.h. SGM_i berechnet $D(E(E(k_{TEK},k2_i),k3_i),k3_i)$. Anschließend wird der Gruppenschlüssel zum Nutzdatenschutz k_{TEK} von SGM_i mit $k1_i$ verschlüsselt, d.h. $E(E(k_{TEK},k2_i),k1_i)$, an die Teilgruppe SG_i weitergeleitet. Das Verfahren wurde entworfen, um die Schlüsselbereitstellung auf mehrere Prozesse zu verteilen, ohne dass diese in den Besitz des Gruppenschlüssels gelangen.

	KS	SGM_i	SG_i
KS	-	$k3_i$	$k2_i$
SGM_i	$k3_i$	-	$k1_i$
SG_i	$k2_i$	$k1_i$	-

Tabelle 9: Gemeinsame Schlüssel beim Verfahren Dual Encryption Protocol

HYDRA

Das Konzept HYDRA [Raf02] gliedert die zu verwaltende Gruppe in Teilgruppen, für die jeweils ein Hydra-Server zuständig ist. Zur Synchronisation der Hydra-Server untereinander wird das Synchronized Group Key Distribution Protocol (SGKDP) verwendet. Das Protokoll stellt sicher, dass alle Hydra-Server den gleichen Gruppenschlüssel in ihrer Teilgruppe bereitstellen. Die Synchronisation durch SGKDP erfolgt ohne zentralen Server. Allerdings wird das Gruppenkommunikationssystem Spread (vgl. Abschnitt 2.2.4) vorausgesetzt. Findet in einer der Teilgruppen eine Teilnehmeroperation statt, initiiert der für die Teilgruppe

zuständige Hydra-Server einen Schlüsselwechsel. Zur Synchronisation des Schlüsselwechsels mit den übrigen Hydra-Servern setzt er das SGKDP ein.

BAAL

Das Konzept BAAL [Chi01] basiert auf einem ähnlichen Konzept wie HYDRA. Es verwendet zur Schlüsselverwaltung so genannte Local Controllers und einen Group Controller. Die Local Controller sind für die Schlüsselverwaltung in einem ihnen zugewiesenen Teilnetzwerk zuständig. Findet in ihrem Zuständigkeitsbereich eine Teilnehmeroperation statt, können sie einen Schlüsselwechsel initiieren. Die Synchronisation des Gruppenschlüsselwechsels stellt der Group Controller durch Zuweisung einer Priorität zu jedem Local Controller sicher.

Distributed Registration and Key distribution

Bei Verfahren Distributed Registration and Key distribution (DiRK) [Opp96] wird zwischen aktiven und passiven Prozessen unterschieden. Eine Gruppenanmeldung kann bei jedem aktiven Prozess durchgeführt werden. Ein neuer Teilnehmer erhält dabei ein Zertifikat, das seine Gruppenmitgliedschaft bescheinigt. Um die übrigen Nutzer über die Gruppenmitgliedschaft zu informieren, sendet jeder Teilnehmer periodisch die Nachricht `RegistrationValidation`. Diese Nachricht enthält auch den öffentlichen Schlüssel eines vom Teilnehmer erzeugten Schlüsselpaares. Ein Schlüsselwechsel innerhalb der Gruppe wird vom Initiator durchgeführt, indem er den neuen Gruppenschlüssel mit den öffentlichen Schlüsseln aus der Nachricht `RegistrationValidation` jedes Nutzers verschlüsselt.

Die nachfolgend vorgestellten Schlüsselbereitstellungsverfahren setzen unterschiedliche Schlüssel in jeder Teilgruppe ein.

IOLUS

Beim Konzept IOLUS gliedert ein Group Security Controller (GSC) [Mit97] die zu verwaltende Gruppe in Teilgruppen. In diesen wird von Group Security Intermediaries (GSI) unabhängig voneinander ein Gruppenschlüssel bereitgestellt. Die Teilgruppen sind hierarchisch organisiert und jeder Group Security Intermediary ist Teilnehmer in der übergeordneten Teilgruppe. Zur sicheren Datenübertragung werden die Nutzdaten vom Sender mit einem von ihm selbst erzeugten Sitzungsschlüssel verschlüsselt. Der Sitzungsschlüssel wiederum wird mit dem in der Teilgruppe gültigen Gruppenschlüssel verschlüsselt und den Nutzdaten beigefügt. Die Group Security Intermediaries führen eine so genannte Sitzungsschlüsselübersetzung durch. Hierzu entschlüsseln sie den Sitzungsschlüssel und verschlüsseln ihn wieder mit dem Gruppenschlüssel der untergeordneten bzw. übergeordneten Teilgruppe. In Abbildung 30 ist die Gruppe in drei Teilgruppen eingeteilt. Die beiden Group Security Intermediaries GSI_2 und GSI_3 entschlüsseln jeweils die vom Nutzer u_6 verwendeten und mit dem Schlüssel k_{TEK_2} verschlüsselten Sitzungsschlüssel. Anschließend wird dieser von GSI_2 mit k_{TEK_3} und von GSI_2 mit k_{TEK_1} verschlüsselt und in die entsprechende Teilgruppe weitergeleitet.

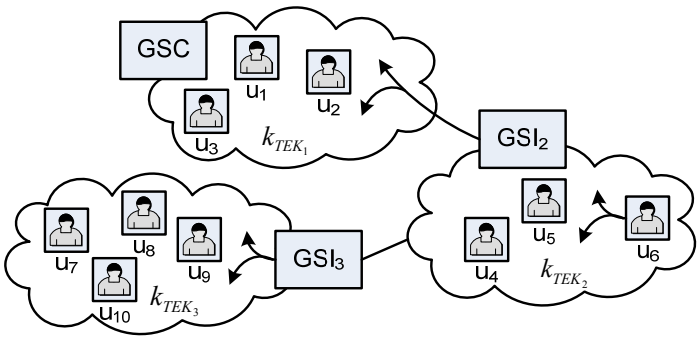

Abbildung 30: Nutzdatenübertragung beim Konzept IOLUS

Cipher Sequences Technique

Das in [Mol00] vorgestellte Verfahren basiert auf der Cipher Sequences Technique (CST). Die Funktion f(S,a) wird Cipher Group genannt, wenn für a_i eine Folge von N Elementen sowie für S_i eine Folge von N+1 Elementen mit S_0 als Startwert gilt $S_i=f(S_{i-1},a_i)$ mit $i>0$. Weiterhin gilt für das Zahlenpaar k,ℓ $S_k=h_{k,\ell}(S_\ell)$ mit $k>\ell$. Insbesondere kann bei einer Cipher Group der Startwert S_0 mit $S_0=h_{0,\ell}(S_\ell)$ wiederhergestellt werden. Zur Anwendung der Cipher Sequences Technique wird jedem für die Verteilung von Multicast-Daten zuständigen Prozess p_i ein Wert a_i mitgeteilt und jede Multicast-Nachricht als Startwert S_0 einer Cipher Group betrachtet. Außerdem findet eine Aufteilung der Nutzer in Teilgruppen statt und es wird festgelegt, welcher Prozess eine Nachricht an welche Teilgruppe weiterleitet. Empfängt ein Prozess S_{i-1}, sendet $S_i=f(S_{i-1},a_i)$ weiter. Nur die Nutzer, die die Funktion $h_{0,\ell}(x)$ kennen, können die Multicast-Nachricht mit $S_0=h_{0,\ell}(S_\ell)$ wiederherstellen. Findet eine Teilnehmeroperation statt, wird dem für die Teilgruppe zuständigen Prozess der neue Wert $\tilde{a}_\ell$ und der Teilgruppe $\tilde{h}_{0,\ell}(x)$ mitgeteilt.

Keyed Hierarchical Multicast Protocol

In Abschnitt 2.6.1 wurde auch das Konzept Keyed Hierarchical Multicast Protocol (KHIP) unter dem Aspekt des Schutzes des Multicast-Verteilbaums dargestellt. Mit dem Konzept kann auch eine Schlüsselbereitstellung für den Schutz von Multicast-Daten vorgenommen werden. Zu diesem Zweck wird von den Routern unabhängig voneinander in jedem Zweig des Multicast-Verteilbaums ein Gruppenschlüssel bereitgestellt. Zur sicheren Datenübertragung werden die Nutzdaten vom Sender mit einem von ihm selbst erzeugten Sitzungsschlüssel verschlüsselt. Der Sitzungsschlüssel wiederum wird mit dem im Zweig des Multicast-Verteilbaums gültigen Gruppenschlüssel verschlüsselt und den Nutzdaten beigefügt. Router die sich an einer Schnittstelle zwischen zwei Zweigen des Multicast-Verteilbaums befinden, führen genau wie beim Konzept IOLUS eine Sitzungsschlüsselübersetzung durch.

Scalable and Adaptive Key Management Scheme

Das Verfahren Scalable and Adaptive Key Management Scheme (SAKM) [Cha04] verwendet das gleiche Konzept zum Nutzdatenschutz bzw. zur Gruppenschlüsselbereitstellung wie beim Konzept IOLUS. Beim Verfahren SAKM werden allerdings zusätzlich in Abhängigkeit von der Teilnehmerdynamik Teilgruppen zu Clustern zusammengefasst. Die Teilgruppen in einem Cluster verwenden alle den gleichen Gruppenschlüssel, sodass zwischen diesen keine Schlüsselübersetzung durchgeführt werden muss.

Scalable Infrastructure for Multicast Key Management

Dem IOLUS-Konzept ebenfalls sehr ähnlich ist das Verfahren Scalable Infrastructure for Multicast Key Management (SIM-KM) [Muk04]. Die Schlüsselübersetzung findet bei diesem Konzept mit der so genannten Proxy Encryption statt, bei der der Prozess, der die Schlüsselübersetzung durchführt, die geschützten Nutzdaten nicht entschlüsseln kann.

3.4.4 Bewertung der hierarchischen Verfahren zur Bereitstellung von Gruppenschlüsseln

Mittels der Konzepte Scalable Multicast Key Distribution (SMKD) und KRONOS läßt sich die Schlüsselgeheimhaltung in einer dynamischen Gruppe (Forderung 1, Tabelle 6) nicht sicherstellen. Die Ursache besteht darin, dass beim Konzept SMKD kein Mechanismus zum Schlüsselwechsel nach der Teilnehmeroperation LEAVE vorhanden ist und beim Konzept KRONOS nur periodisch einen Schlüsselwechsel durchführt. Die Konzepte Inter-Domain Group Key Management (IGKMP), Group Key Management with Network Mobility (GKMNM), Dual Encryption Protocol (DEP) und Distributed Registration and Key Distribution (DiRK) stellen die Schlüsselgeheimhaltung (Forderung 1, Tabelle 6) durch entsprechende Methoden sicher und überprüfen beim Gruppenbeitritt die Zugangsberechtigung (Forderung 2, Tabelle 6). Durch die Unterteilung der Gruppen bieten sie einen effizienten Schlüsselwechsel bei steigender Gruppengröße. Hierbei wird bei DEP sichergestellt, dass die Sub-group Manager keinen Zugriff auf den Gruppenschlüssel haben, während GKMNM speziell die Mobilität von Nutzern berücksichtigt. Allerdings sind derartige Verfahren nicht in kleinen Gruppen einsetzbar und daher ist die Forderung 3, Tabelle 6 nicht erfüllt. Weiterhin besitzen diese Konzepte eine hierarchische Struktur und damit einen Single Point of Failure, so dass auch die notwendige Reparierbarkeit (Forderung 5, Tabelle 6) nicht gegeben ist. Im Gegensatz dazu ist bei den Konzepten HYDRA, BAAL, IOLUS, Cipher Sequences Technique (CST), Keyed Hierarchical Multicast Protocol (KHIP), Scalable and Adaptive Key Management Scheme (SAKM) und Scalable Infrastructure for Multicast Key Management (SIM-KM) die Reparierbarkeit (Forderung 5, Tabelle 6) des Schlüsselmanagementsystem gegeben. Nachteil dieser Verfahren ist die fehlende Möglichkeit, sie in kleinen Gruppen (Forderung 3, Tabelle 6) einzusetzen. Außerdem findet eine Übersetzung der verschlüsselten Multicast-Nutzdaten bzw. der Sitzungsschlüssel statt, so dass sie für eine Gruppenschlüsselbereitstellung zum Schutz von Realzeitdatentransfer (Forderung 9, Tabelle 6) nur bedingt geeignet sind. Die Bewertung kann wie folgt zusammengefasst werden. Die in diesem Abschnitt vorgestellten Schlüsselbereitstellungsverfahren erfüllen durch Redundanz in gewissem Umfang die Forderung der Reparierbarkeit. Allerdings sind die Verfahren zur Schlüsselverwaltung in großen Gruppen entworfen und eignen sich nicht für kleine Gruppen, so dass die für den Einsatzbereich notwendige Skalierbarkeit (Forderung 3, Tabelle 6) nicht erfüllt ist. Die Bewertung der hierarchischen Verfahren zur Bereitstellung von Gruppenschlüsseln ist in Tabelle 10 zusammengefasst.

Verfahren/ Anforderung	1 Schlüssel-geheim-haltung	2 Zugriffs-kontrolle	3 Skalierbar-keit	4 Robuste Schlüssel-über-mittlung	5 Reparier-barkeit	6 Mehrfach-anfragen	7 EMCON	8 IPSec-Adaption	9 Realzeit
SMKD	-	+	-	0	-	0	0	0	+
IGKMP	+	+	-	0	-	0	0	0	+
GKMNM	+	+	-	0	-	0	0	0	+
KRONOS	-	+	-	0	-	0	0	0	+
DEP	+	+	-	0	-	0	0	0	+
HYDRA	+	+	-	0	0	0	0	0	+
BAAL	+	+	-	0	0	0	0	0	+
DiRK	+	+	-	0	-	0	0	0	+
IOLUS	+	+	-	0	+	0	0	0	-
CST	+	+	-	0	?	0	0	0	-
KHIP	+	+	-	0	+	0	0	0	-
SAKM	+	+	-	0	+	0	0	0	-
SIM-KM	+	+	-	0	+	0	-	0	-

+ erfüllt 0 prinzipiell erfüllbar - nicht erfüllt ? unklar

Tabelle 10: Bewertung hierarchischer Verfahren zur Bereitstellung von Gruppenschlüsseln

3.4.5 Beschreibung verteilter Verfahren zur Bereitstellung von Gruppenschlüsseln

Die Verfahren zur verteilten Schlüsselbereitstellung können in die Unterkategorien Schlüsselvereinbarung und dezentrale Schlüsselverteilung separiert werden. Mit der Darstellung der zuerst genannten Verfahren wird begonnen.

Ingemarsson-Tang-Wong Group Diffie-Hellman

Bei dem Verfahren Ingemarsson-Tang-Wong Group Diffie-Hellman (ITW) [Ing82] sind die Nutzer $\{u_1,\ldots,u_U\}$ als logischer Ring organisiert, in dem ein Token kreist. Das Token wird dazu verwendet, Hilfsschlüssel zwischen den Nutzern auszutauschen. Auf Grund der logischen Anordnung der Nutzer als Ring übermittelt der Nutzer u_i das Token immer an den Nutzer u_{i+1}.

(1) Zu Beginn generiert jeder Nutzer u_i den (geheimen) Schlüssel k_i entsprechend der Vorgaben des DH-Algorithmus. Während des ersten Umlaufs des Tokens sendet der Nutzer u_i den abgeleiteten Blindschlüssel $bk_i=BK(k_i)$ an den Nutzer u_{i+1}.

(2) In den nun folgenden (U-2) Umläufen des Tokens wird jeweils mit dem DH-Algorithmus ein weiterer Hilfsschlüssel berechnet und an den Nachbar im logischen Ring übermittelt. Als Eingaben für den DH-Algorithmus dienen der im vorherigen Umlauf des Tokens empfangene Hilfsschlüssel und der eigene zu Beginn generierte Schlüssel. Im letzten Umlauf des Tokens erhält der Nutzer u_i den Hilfsschlüssel $z_{1\ldots i-1 i+1\ldots U} = g^{\prod_{j=1, j\neq i}^{U} k_j}$ und kann den Gruppenschlüssel mit $k_{group}=DH(z_{1\ldots i\text{-}1\ i+1\ldots U},k_i)$ berechnen.

Diese beiden Verarbeitungsschritte müssen sowohl bei der Teilnehmeroperation JOIN als auch bei der Operation LEAVE durchgeführt werden. In Abbildung 31 ist ein Beispiel mit

vier Nutzern $\{u_1,u_2,u_3,u_4\}$ dargestellt, bei dem das Token mit den Hilfsschlüsseln dreimal den logischen Ring durchläuft, bis zum Beispiel der Nutzer u_3 den Gruppenschlüssel mit $k_{group}=DH(z_{124},k_3)$ berechnen kann.

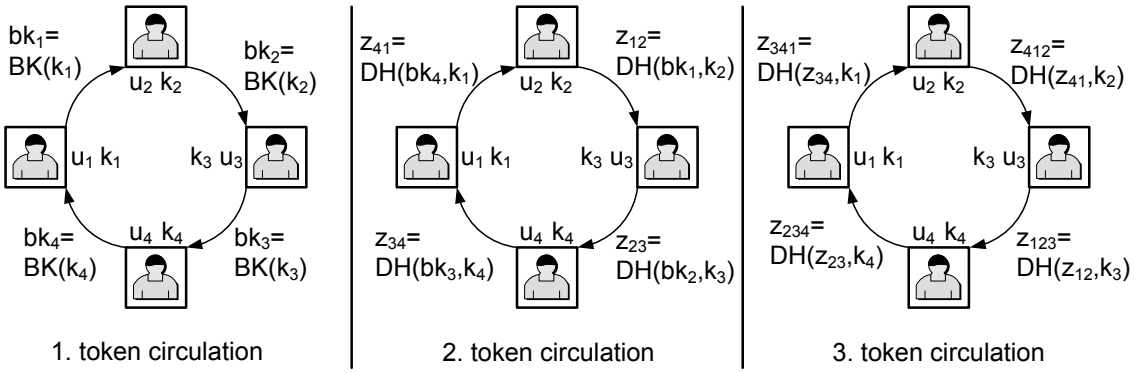

Abbildung 31: Ablauf und ausgetauschte Hilfsschlüssel beim Verfahren ITW mit vier Nutzern

Burmester-Desmedt Group Diffie-Hellman

Beim Verfahren Burmester-Desmedt Group Diffie-Hellman (BD) [Bur94] sind die Nutzer $\{u_1,\ldots,u_U\}$ ebenfalls als logischer Ring organisiert. Bei diesem Verfahren sind zur Ermittlung eines gemeinsamen Schlüssels drei Verarbeitungsschritte notwendig:

(1) Zu Beginn generiert jeder Nutzer u_i mit $i\in\{1,\ldots,U\}$ den (geheimen) Schlüssel k_i entsprechend der Vorgaben des DH-Algorithmus. Anschließend sendet der Nutzer u_i den abgeleiteten Blindschlüssel $bk_i=BK(k_i)$ an alle übrigen Nutzer $\{u_1,\ldots,u_{i-1},u_{i+1},\ldots,u_U\}$.

(2) Nach dem Empfang der Blindschlüssel $\{bk_1,\ldots,bk_{i-1},bk_{i+1},\ldots,bk_U\}$ durch die Nutzer u_i $i\in\{1,\ldots,U\}$ wird der Hilfsschlüssel $z_i = DH(\frac{bk_{i+1}}{bk_{i-1}}, k_i) \bmod p$ berechnet. Anschließend sendet der Nutzer u_i das Ergebnis an alle übrigen Nutzer $\{u_1,\ldots,u_{i-1},u_{i+1},\ldots,u_U\}$.

(3) Nach Empfang der Hilfsschlüssel kann jeder Nutzer u_i $i=1,\ldots,U$ den Gruppenschlüssel mit $k_{group} = (bk_{i-1})^{U\cdot k_i}\cdot(z_i)^{U-1}\cdot(z_{i+1})^{U-2}\cdot\ldots\cdot(z_{i-2}) \bmod p$ berechnen. Bei der Anwendung der Formel ist zu beachten, dass, wie zu Beginn des Abschnitts erwähnt, die Nutzer als Ring organisiert sind. Deshalb bezeichnet der Blindschlüssel bk_{i-1} für $i=1$ in einer Gruppe mit vier Nutzern den Blindschlüssel bk_4.

Zur Illustration des Verfahrens ist in Abbildung 32 die Schlüsselbereitstellung für die Nutzer $\{u_1,u_2,u_3,u_4\}$ dargestellt. Im ersten Protokollschritt verteilt der Nutzer u_1 den abgeleiteten Blindschlüssel bk_1, der Nutzer u_2 den abgeleiteten Blindschlüssel bk_2, der Nutzer u_3 den abgeleiteten Blindschlüssel bk_3 und der Nutzer u_4 den abgeleiteten Blindschlüssel bk_4. Im zweiten Schritt werden die Hilfsschlüssel z_1, z_2, z_3 und z_4 den übrigen Nutzern übermittelt. Im dritten Protokollschritt berechnet jeder den gemeinsamen Gruppenschlüssel $k_{group} = g^{k_1\cdot k_2+k_2\cdot k_3+k_3\cdot k_4+k_4\cdot k_1} \bmod p$ entsprechend Abbildung 32, links.

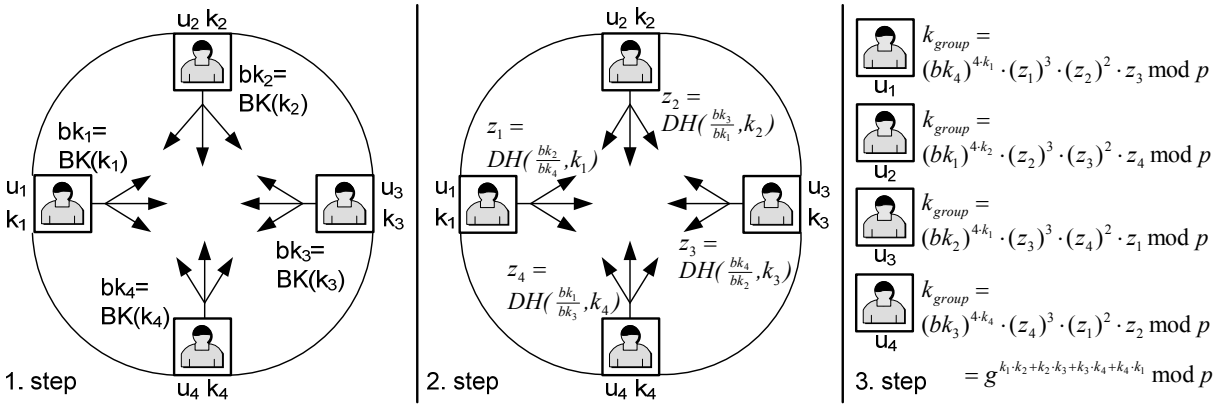

Abbildung 32: Ablauf und ausgetauschte Hilfsschlüssel beim Verfahren BD mit vier Nutzern

Steiner-Tsudik-Waidner Group Diffie-Hellman

In diesem Absatz wird das Verfahren Steiner-Tsudik-Waidner Group Diffie-Hellman (STW) [Ste96], das auch häufig als Group Diffie-Hellman (GDH) bezeichnet wird, vorgestellt. Im Gegensatz zum Burmester-Desmedt Group Diffie-Hellman sind die Nutzer in einer logischen Kette angeordnet. Das Funktionsprinzip des Verfahrens besteht darin, dass in der ersten Phase ein Datenpaket mit den Hilfsschlüsseln in der logischen Kette wandert. In der zweiten Phase werden vom letzten Nutzer der Kette die zur Gruppenschlüsselberechnung notwendigen Hilfsschlüssel an die übrigen Nutzer verteilt. Die Teilnehmeroperation JOIN läuft in einer Gruppe $\{u_1,\ldots,u_{U-1}\}$ mit U-1 Teilnehmern wie folgt ab:

(1) Der Teilnehmer u_{U-1} erzeugt entsprechend der Vorgaben des DH-Algorithmus einen zufälligen (geheimen) Schlüssel κ. Hieraus ermittelt er $\tilde{k}_{U-1}= \kappa/k_{U-1}$. Anschließend sendet der Nutzer u_i die Hilfsschlüssel $\{\tilde{z}_{1\ldots\ell-1\,\ell+1\ldots U-1} = g^{\tilde{k}_{U-1}\cdot\prod_{j=1,j\neq\ell}^{U-1}k_j} \mid \ell\in[1,U-1], \tilde{z}_{1\ldots U-1} = g^{\tilde{k}_{U-1}\cdot\prod_{j=1}^{U-1}k_j}\}$ an den Nutzer u_U.

(2) Aus den empfangenen Hilfsschlüsseln ermittelt der Nutzer u_U durch Anwendung des DH-Algorithmus $\tilde{z}_{1\ldots\ell\text{-}1\,\ell+1\ldots U}=DH(\tilde{z}_{1\ldots\ell\text{-}1\,\ell+1\ldots U},k_i)$ mit $\ell\in[1,U]$. Anschließend sendet er an die Nutzer $\{u_1,\ldots,u_{U-1}\}$ die Hilfsschlüssel $\{\tilde{z}_{1\ldots\ell-1\,\ell+1\ldots U} = g^{\tilde{k}_{U-1}\cdot\prod_{j=1,j\neq\ell}^{U}k_j} \mid \ell\in[1,U]\}$. Aus den empfangenen Informationen kann zum Beispiel der Nutzer u_i den Gruppenschlüssel durch $k_{group}=DH(\tilde{z}_{1\ldots i\text{-}1\,i+1\ldots U},k_i)$ ermitteln.

Nun wird die Schlüsselbereitstellung bei der Teilnehmeroperation LEAVE dargestellt. Tritt der Nutzer u_j aus der Gruppe von U+1 Nutzern aus, wird der neue Gruppenschlüssel wie folgt etabliert:

(1) Der Teilnehmer u_i erzeugt entsprechend der Vorgaben des DH-Algorithmus einen zufälligen (geheimen) Schlüssel κ. Hieraus ermittelt er $\tilde{k}_i= \kappa/k_i$. Anschließend sendet er die Hilfsschlüssel $\{\tilde{z}_{1\ldots\ell-1\,\ell+1\ldots U} = g^{\tilde{k}_i\cdot\prod_{j=1,j\neq\ell}^{U}k_j} \mid \ell\in[1,U]\}$ an die Nutzer $\{u_1,\ldots,u_{j-1},u_{j+1},\ldots,u_U\}$

(2) Aus den empfangenen Hilfsschlüsseln ermittelt der Nutzer u_i den Gruppenschlüssel durch $k_{group}=DH(\tilde{z}_{1\ldots i\text{-}1\,i+1\ldots U},k_i)$.

Zur Verdeutlichung des Funktionsprinzips von STW ist in Abbildung 33 der Ablauf bei der Initialisierung einer Gruppe mit vier Nutzern dargestellt. Nachdem der Nutzer u_4 zu den empfangenen Hilfsschlüsseln seine (geheimen) Schlüssel hinzugefügt hat, versendet er die Hilfsschlüssel z_{124}, z_{134} und z_{234} an die Nutzer $\{u_1,u_2,u_3\}$. Aus diesen können der Nutzer u_1

mit $k_{group}=DH(z_{234},k_1)$, der Nutzer u_2 mit $k_{group}=DH(z_{134},k_2)$ und der Nutzer u_3 mit $k_{group}=DH(z_{124},k_3)$ den Gruppenschlüssel berechnen. Der Nutzer u_4 selbst berechnet den Gruppenschlüssel mit $k_{group}=DH(z_{123},k_4)$ aus einem vom Nutzer u_3 empfangenen Hilfsschlüssel.

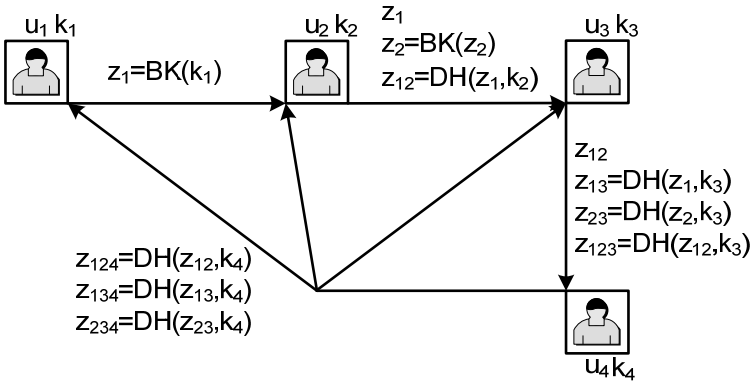

Abbildung 33: Ablauf und ausgetauschte Hilfsschlüssel beim Verfahren STW mit vier Nutzern

2^d-cube

Beim Verfahren 2^d-cube [Bec98] wird angenommen, dass $U=2^d$ Nutzer vorhanden sind. Das Protokoll benötigt in diesem Fall d Protokollschritte zur Aushandlung, wobei in jedem dieser Protokollschritte je zwei Nutzer eine Schlüsselvereinbarung mittels DH-Algorithmus durchführen. Die Nutzer werden bei diesem Verfahren mit Vektoren $\vec{u}_1,\ldots,\vec{u}_d$ des d-dimensionalen Vektorraums mit der Basis $\vec{e}_1,\ldots,\vec{e}_d$ bezeichnet.

(1) Jeder Nutzer $\vec{u}_i$ erzeugt entsprechend der Vorgaben des DH-Algorithmus einen zufälligen (geheimen) Schlüssel k_i. Anschließend vereinbart der Nutzer $\vec{u}_i$ i=1,…,U/2 mit dem Nutzer $\vec{u}_i+\vec{e}_1$ durch Anwendung des DH-Algorithmus einen gemeinsamen Schlüssel.

(2) In den nachfolgenden j Protokollschritten ($1<j\leq d$) führt der Nutzer $\vec{u}_i$ mit dem Nutzer $\vec{u}_i+\vec{e}_j$ eine Schlüsselvereinbarung gemäß dem DH-Algorithmus durch. Als (geheimer) Schlüssel wird der im Protokollschritt j-1 vereinbarte Schlüssel verwendet.

In Abbildung 34 ist zur Verdeutlichung des Verfahrens ein Beispiel mit vier Nutzern dargestellt. Diese sind mit den Vektoren $\vec{u}_1,\vec{u}_2,\vec{u}_3,\vec{u}_4$ bezeichnet und haben entsprechend der Vorgaben des DH-Algorithmus die zufälligen (geheimen) Schlüssel k_1,k_2,k_3,k_4 erzeugt. Es sind zwei Protokollschritte notwendig. Im ersten Schritt vereinbaren die Nutzer $\vec{u}_1,\vec{u}_2$ den gemeinsamen Schlüssel k_{12} und die Nutzer $\vec{u}_3,\vec{u}_4$ den gemeinsamen Schlüssel k_{34}. Nach dem Empfang des aus der Schlüsselvereinbarung zwischen $\vec{u}_1$ und $\vec{u}_2$ abgeleiteten Blindschlüssels $bk_{12}=BK(k_{12})$ kann der Nutzer $\vec{u}_3$ den Gruppenschlüssel mit $k_{group}=DH(bk_{12},k_{34})$ berechnen. Das beschriebene Protokoll funktioniert nur, wenn die Anzahl der Nutzer eine Potenz von 2 ist. Die verallgemeinerte Version des Protokolls mit einer beliebigen Zahl von Nutzern wird 2^d-octopus genannt und wird ebenfalls in [Bec98] vorgestellt.

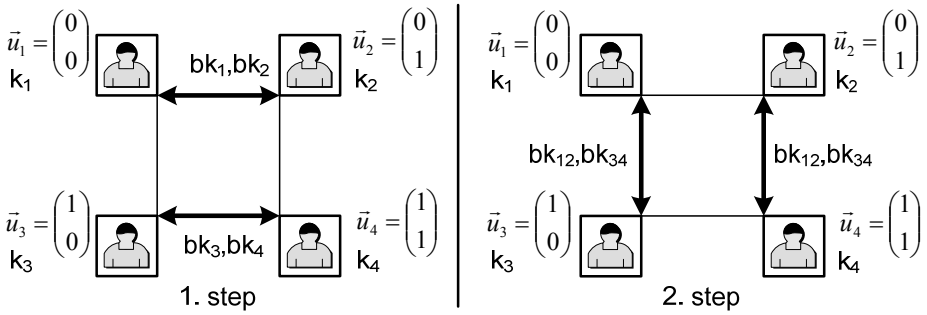

Abbildung 34: Ablauf und ausgetauschte Hilfsschlüssel beim Verfahren 2^d-cube mit vier Nutzern

Tree-based Group Diffie-Hellman

Das Verfahren Tree-based Group Diffie-Hellman (TGDH) [Kim00] berechnet den Gruppenschlüssel aus dem Schlüsselmaterial der Gruppenteilnehmer unter Verwendung eines binären Schlüsselbaums in Kombination mit einer iterativen Anwendung des DH-Algorithmus. Der Schlüsselbaum wird bei diesem Verfahren in der nachfolgend beschriebenen Art organisiert. Jedem Knoten $v_{\ell,p}$ wird ein (geheimer) Schlüssel $k_{\ell,p}$ und der aus diesem abgeleitete Blindschlüssel $bk_{l,p} = g^{k_{l,p}} \bmod p$ zugeordnet (vgl. Abschnitt 3.1). Das Prinzip des Protokolls besteht darin, dass ein Nutzer den Schlüsselbaum und alle Blindschlüssel des Baums kennt. Auf Grund dieser Kenntnisse ist er in der Lage, die Schlüssel $k_{\ell,p}$ der Knoten $v_{\ell,p}$ des Pfades zur Wurzel mittels wiederholter Anwendung des DH-Algorithmus aus den Schlüsseln bzw. Blindschlüsseln der linken Nachfolger $v_{\ell+1,2p}$ und rechten Nachfolger $v_{\ell+1,2p+1}$ zu berechnen. Der Schlüssel im Wurzelknoten ist der allen Nutzern bekannte Gruppenschlüssel. Die Blätter des Schlüsselbaumes enthalten die Beiträge der Gruppenteilnehmer zum Gruppenschlüssel. Bei Änderung von Gruppengröße und Zusammensetzung auf Grund der Teilnehmeroperationen JOIN, MERGE, LEAVE und PARTITION kann der Gruppenschlüssel gewechselt werden. Ein so genannter Sponsor übernimmt hierbei die Aufgabe, den Schlüsselbaum zu aktualisieren und die Blindschlüssel zu verteilen. Der Sponsor wird aus der Baumstruktur ermittelt. Exemplarisch sind der Beitritt und Austritt eines Nutzers dargestellt:

Zur Beschreibung des Ablaufs des Verfahrens TGDH bei der Teilnehmeroperation JOIN wird der Beitritt des Nutzers u_{U+1} zu der bereits bestehenden Gruppe $G=\{u_1,\ldots,u_U\}$ aus U Teilnehmern betrachtet.

(1) Der neue Nutzer u_{U+1} versendet die Beitrittsanfrage als Multicast-Nachricht. In der Nachricht ist der Blindschlüssel $bk_{\ell_{U+1},p_{U+1}}$ des Nutzers u_{U+1} enthalten. Dieser wurde aus dem entsprechend der Vorgaben des DH-Algorithmus zufällig erzeugten (geheimen) Schlüssel $k_{\ell_{U+1},p_{U+1}}$ abgeleitet.

(2a) Alle Nutzer fügen den neuen Nutzer u_{U+1} an die Position $v_{\ell_{U+1},p_{U+1}}$ ein. Weiterhin werden die Schlüssel $k_{\ell,p}$ und Blindschlüssel $bk_{\ell,p}$ der Knoten $\{w_{\ell_k,p_k} \mid w_{\ell_k,p_k} \in K\tilde{T} \wedge w_{\ell_k,p_k} \in path(v_{\ell_S,p_S})\}$, d.h. der Knoten auf dem Pfad vom Sponsor v_{ℓ_S,p_S} zum Wurzelknoten, gelöscht.

(2b) Der Sponsor u_S generiert den Schlüssel $\tilde{k}_{\ell_S,p_S}$ und aktualisiert den modifizierten Schlüsselbaum $K\tilde{T}$:

for each $w_{\ell_k,p_k} \in path(v_{\ell_S,p_S})$ do

    if $w_{\ell_{k-1},p_{k-1}} = = w_{\ell_k+1,2p_k}$ then do

$\tilde{k}_{\ell_k,p_{kk}}=DH(bk_{\ell_k+1,2p_k+1},\tilde{k}_{\ell_k+1,2p_k}),\ b\tilde{k}_{\ell_k,p_k}=BK(\tilde{k}_{\ell_k,p_k})$

else do

$\tilde{k}_{\ell_k,p_k}=DH(bk_{\ell_k+1,2p_k},\tilde{k}_{\ell_k+1,2p_k+1}),\ b\tilde{k}_{\ell_k,p_k}=BK(\tilde{k}_{\ell_k,p_k})$

Anschließend verteilt der Sponsor die Blindschlüssel $b\tilde{k}_{\ell,p}$ der Knoten $\{v_{\ell,p} \mid v_{\ell,p}\in K\tilde{T}\}$ an die Gruppe $\{u_1,\ldots,u_{S-1},u_{S+1},\ldots,u_{U+1}\}$.

(4) Die Nutzer $\{u_1,\ldots,u_{S-1},u_{S+1},\ldots,u_{U+1}\}$ aktualisieren den Schlüsselbaum mit den erhaltenen Blindschlüsseln. Anschließend erfolgt die Ermittlung der aktuellen Position im Schlüsselbaum $\tilde{v}_{l_r,p_r}$ mit r=1,…,S-1,S+1,…,U+1 durch die Nutzer sowie Berechnung des Gruppenschlüssels:

for each $w_{\ell_k,p_k}\in path(\tilde{v}_{\ell_r,p_r})$ do

if $w_{\ell_{k-1},p_{k-1}} = = w_{\ell_k+1,2p_k}$ then do

$\tilde{k}_{\ell_k,p_k}=DH(b\tilde{k}_{\ell_k+1,2p_k+1},\tilde{k}_{\ell_k+1,2p_k})$

else do

$\tilde{k}_{\ell_k,p_k}=DH(b\tilde{k}_{\ell_k+1,2p_k},\tilde{k}_{\ell_k+1,2p_k+1})$

Das Beispiel in Abbildung 35 verdeutlicht den Beitritt des Nutzers u_4 zu einer Gruppe mit den drei Nutzern $\{u_1,u_2,u_3\}$. Jedes bereits vorhandene Gruppenmitglied fügt den Nutzer in den Schlüsselbaum ein. Der Sponsor u_3 dieser Teilnehmeroperation generiert zusätzlich einen neuen Schlüssel, berechnet $k_{1,1}$, $bk_{1,1}$, $k_{0,0}$ und verteilt anschließend alle Blindschlüssel per Multicast. Versorgt mit dieser Nachricht können u_1 und u_2 den Schlüssel $k_{0,0}$ sowie u_4 die Schlüssel $k_{1,1}$, $k_{0,0}$ berechnen.

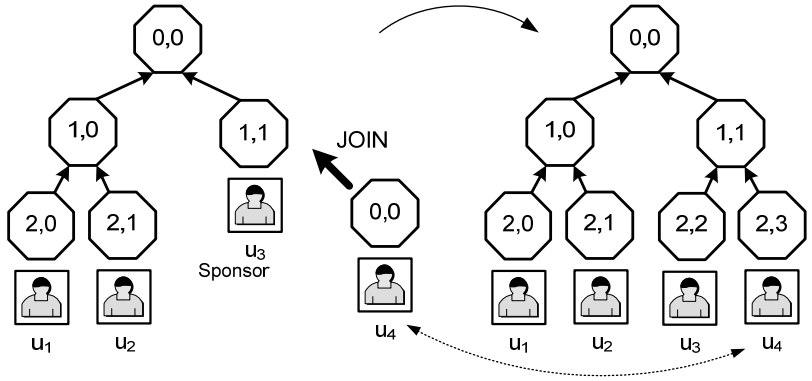

Abbildung 35: Teilnehmeroperation JOIN beim Verfahren TGDH

Zur Beschreibung der Funktionsweise des Verfahrens TGDH bei der Teilnehmeroperation LEAVE wird der Austritt des Nutzers u_i mit $i\in\{1,\ldots,U\}$ aus der bereits bestehenden Gruppe $G=\{u_1,\ldots,u_U\}$ mit U Teilnehmern betrachtet.

(1) Durch den Nutzer u_i wird die Austrittsanfrage als Multicast-Nachricht.

(2a) Alle Nutzer löschen den Knoten v_{ℓ_i,p_i} des Nutzers u_i sowie dessen Vorgänger. Weiterhin werden die Schlüssel $k_{\ell,p}$ und Blindschlüssel $bk_{\ell,p}$ der Knoten $\{w_{\ell_k,p_k} \mid w_{\ell_k,p_k}\in K\tilde{T} \wedge w_{\ell_k,p_k}\in path(v_{\ell_s,p_s})\}$, d.h. der Knoten auf dem Pfad vom Sponsor v_{ℓ_s,p_s} zum Wurzelknoten, gelöscht.

(2b) Der Sponsor u_S erzeugt zufällig entsprechend der Vorgaben des DH-Algorithmus den Schlüssel $\tilde{k}_{\ell_S,p_S}$ und aktualisiert den modifizierten Schlüsselbaum $K\tilde{T}$:
for each $w_{\ell_k,p_k} \in path(v_{\ell_S,p_S})$ do

if $w_{\ell_{k-1},p_{k-1}} = = w_{\ell_k+1,2p_k}$ then do
$\tilde{k}_{\ell_k,p_k}=DH(bk_{\ell_k+1,2p_k+1},\tilde{k}_{\ell_k+1,2p_k})$, $b\tilde{k}_{\ell_k,p_k}=BK(\tilde{k}_{\ell_k,p_k})$
else do
$\tilde{k}_{\ell_k,p_k}=DH(bk_{\ell_k+1,2p_k},\tilde{k}_{\ell_k+1,2p_k+1})$, $b\tilde{k}_{\ell_k,p_k}=BK(\tilde{k}_{\ell_k,p_k})$

Anschließend verteilt der Sponsor die Blindschlüssel $b\tilde{k}_{\ell,p}$ der Knoten $\{v_{\ell,p} \mid v_{\ell,p} \in K\tilde{T}\}$ an die Gruppe $\{u_1,\ldots,u_{i-1},u_{i+1},..,u_{S-1},u_{S+1},\ldots,u_U\}$.

(4) Die Nutzer $\{u_1,\ldots,u_{i-1},u_{i+1},..,u_{S-1},u_{S+1},\ldots,u_U\}$ aktualisieren den Schlüsselbaum mit den erhaltenen Blindschlüsseln. Anschließend wird die Position im Schlüsselbaum $\tilde{v}_{l_r,p_r}$ mit r=1,…,i-1,i+1,…,S-1,S+1,…,U durch die Nutzer ermittelt sowie der Gruppenschlüssel berechnet:
for each $w_{\ell_k,p_k} \in path(\tilde{v}_{\ell_r,p_r})$ do

if $w_{\ell_{k-1},p_{k-1}} = = w_{\ell_k+1,2p_k}$ then do
$\tilde{k}_{\ell_k,p_k}=DH(b\tilde{k}_{\ell_k+1,2p_k+1},\tilde{k}_{\ell_k+1,2p_k})$
else do
$\tilde{k}_{\ell_k,p_k}=DH(b\tilde{k}_{\ell_k+1,2p_k},\tilde{k}_{\ell_k+1,2p_k+1})$

In Abbildung 36 ist ein Beispiel dargestellt, bei dem Nutzer u_3 eine Gruppe verlässt. Die in der Gruppe verbleibenden Nutzer $\{u_1,u_2,u_4,u_5\}$ entfernen die Knoten $v_{2,2}$ und $v_{1,1}$. Der Nutzer u_4 übernimmt die Aufgaben des Sponsors. Dieser wechselt deshalb seinen Schlüssel und berechnet zusätzlich die Schlüssel $k_{1,1}$, $bk_{1,1}$ und $k_{0,0}$. Anschließend verteilt er alle im Baum befindlichen Blindschlüssel mittels Multicast. Versorgt mit dieser Nachricht können u_1 und u_2 den Schlüssel $k_{0,0}$ sowie u_5 die Schlüssel $k_{1,1}$, $k_{0,0}$ berechnen.

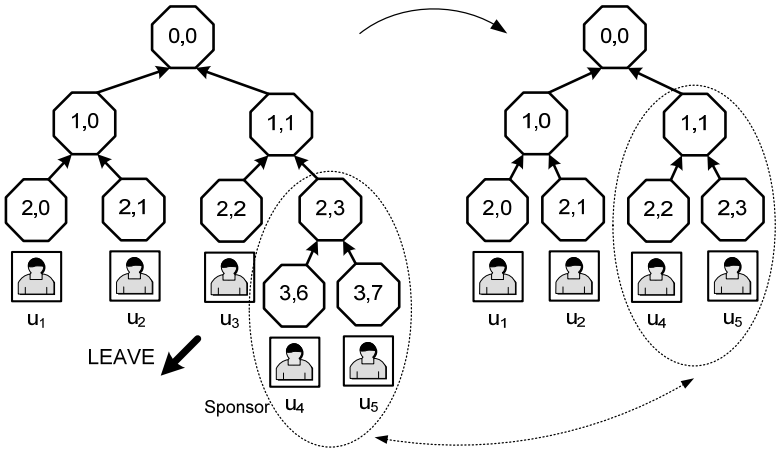

Abbildung 36: Teilnehmeroperation LEAVE beim Verfahren TGDH

Skinny Tree

Das Skinny Tree Protocol (STR) [Ste88] verwendet ebenfalls einen Schlüsselbaum bei der Bereitstellung des Gruppenschlüssels. Allerdings besitzt dieser die in Abbildung 37 dargestellte entartete Struktur. Bei diesem Protokoll wird der Gruppenschlüssel rekursiv berechnet mit $k_i=DH(bk_{i+3},k_{i+2})$ für i>0 und $k_i=DH(bk_2,k_1)$ für i=0. Zur Illustration ist in

Abbildung 37 der bei diesem Verfahren verwendete Schlüsselbaum für eine Gruppe bestehend aus vier Nutzern $\{u_1,u_2,u_3,u_4\}$ dargestellt. In diesem Beispiel berechnet der Nutzer u_1 den in der Wurzel des Baums gespeicherten Gruppenschlüssel rekursiv mit $k_{group}=k_0=DH(bk_2,DH(bk_4,DH(bk_6,k_5)))$.

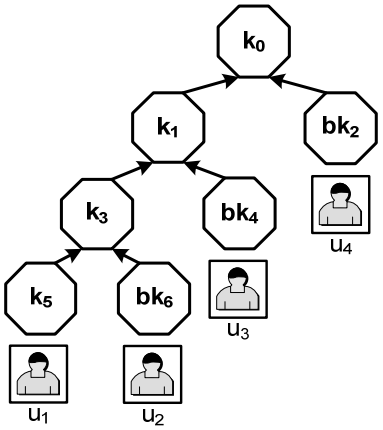

Abbildung 37: Schlüsselbaum des Nutzers u_1 beim Verfahren STR

Nachdem bisher die Verfahren der Unterkategorie Schlüsselvereinbarung vorgestellt wurden, werden nun die Verfahren der dezentralen Schlüsselverteilung dargestellt.

Distributed Logical Key Hierarchy

Das Verfahren Distributed Logical Key Hierarchy (DLKH) [Rod00] ist eine dezentrale Form des in Abschnitt 3.4.1 vorgestellten Logical Key Hierarchy. Im Gegensatz zum zentralen Verfahren kann ausschließlich ein Schlüsselbaum vom Grad d=2 als Hilfsmittel zur Schlüsselbereitstellung verwendet werden. Bei dem Verfahren DLKH existiert kein GCKS, der den vollständigen Schlüsselbaum kennt. Es wird für jeden Teilbaum ein für die Schlüsselbereitstellung verantwortlicher Nutzer, der so genannte Sub Tree Leader (STL), bestimmt. Diese Festlegung erfolgt auf der Basis der Position des Nutzers im Baum. Ein STL legt mit einem anderen STL einen gemeinsamen Schlüssel fest und ist für die Weitergabe dieser Information an die Geschwister in seinem Teilbaum verantwortlich. Durch die nachfolgenden zwei Verfahrensschritte wird von den STL u_L und u_R ein Gruppenschlüssel bereitgestellt:

(1) Der STR u_L erzeugt zufällig den Schlüssel k_{LR} und übermittelt ihn geschützt an den STL u_R. Hierzu kann u_L den Schlüssel k_{LR} zum Beispiel mit dem öffentlichen Schlüssel pk_R des Nutzers u_R verschlüsseln, d.h. $E(k_{LR},pk_R)$, übermitteln. Voraussetzung hierfür ist, dass im Vorfeld der Gruppenschlüsselbereitstellung dem Nutzer u_R das asymmetrische Schlüsselpaar (sk_R,pk_R) zugeordnet wurde.

(2) Im zweiten Schritt muss der Schlüssel k_{LR} den übrigen Nutzer mitgeteilt werden. Mit $E(k_{LR},k_L)$ übermittelt der STR u_L seinen Geschwistern im Teilbaum KT_L und mit $E(k_{LR},k_R)$ übermittelt der STR u_R seinen Geschwistern im Teilbaum KT_R den neuen Schlüssel k_{LR} des gemeinsamen Baums $K\tilde{T}=KT_L \cup KT_R$.

Bei der Initialisierung werden die beschriebenen zwei Verfahrensschritte so oft wiederholt bis bei allen Nutzern der Gruppenschüssel vorhanden ist. In Abbildung 38 ist die Festlegung eines gemeinsamen Schlüssels für die im Teilbaum KT_L organisierten Nutzer $\{u_1,u_2,u_3,u_4\}$

und die im Teilbaum KT_R organisierten Nutzer $\{u_5,u_6,u_7,u_8\}$ dargestellt. Die STL u_1 und u_8 vereinbaren den gemeinsamen Schlüssel $k_{LR}=k_{0,0}$ des Baums $K\tilde{T}=KT_L \cup KT_R$. Im Anschluss daran informiert der STL u_1 die übrigen Nutzer des Teilbaums KT_L mit $E(k_{00},k_{1,0})$ über den neuen gemeinsamen Gruppenschlüssel. Während der STL u_8 die übrigen Nutzer des Teilbaums KT_R mit $E(k_{00},k_{1,1})$ informiert.

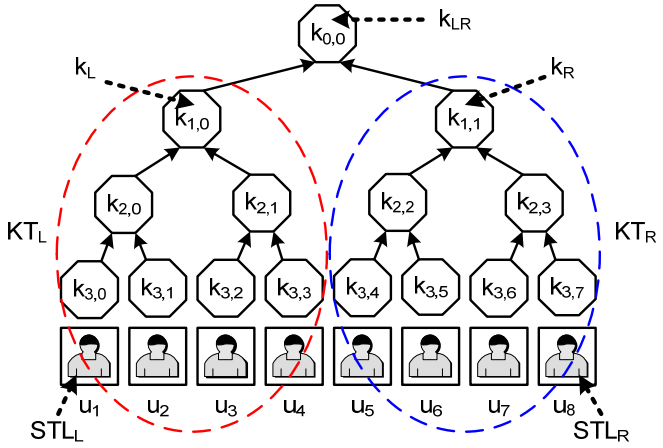

Abbildung 38: Schlüsselbaum beim Verfahren DLKH mit u_1 und u_8 als Sub Tree Leader

Distributed One-way Function Tree

Das Verfahren Distributed One-way Function Tree (DOFT) [Don99] ist eine dezentrale Form des in Abschnitt 3.4.1 vorgestellten Verfahrens One-way Function Tree. Auch bei der dezentralen Form werden die Hilfsschlüssel rekursiv aus den in den Kindern gespeicherten Schlüsseln berechnet und nicht nur verteilt. Wie beim Verfahren Distributed Logical Key Hierarchy existiert kein GCKS, der den gesamten Schlüsselbaum kennt, sondern es wird für jeden Teilbaum ein Sub Tree Leader festgelegt.

Distributed Flat Table

Zusätzlich zu den Verfahren DLKH und DOFT existiert auch das Verfahren Distributed Flat Table (DFT) [Don99], die dezentrale Form des in Abschnitt 3.4.1 vorgestellten Mechnismus Centralized Flat Table.

Virtual Token-based Key Distribution

Beim Verfahren Virtual Token-based Key Distribution (VTKD) [Liu05] existiert für jedes Nutzerpaar u_k und u_ℓ $k,\ell=\{1,\ldots,U|k\neq\ell\}$ einer Gruppe mit U Teilnehmern ein gemeinsamer mittels des DH-Algorithmus vereinbarter Schlüssel $k_{k/\ell}=D(bk_k,k_\ell)=DH(bk_\ell,k_k)$. Ein neuer Gruppenschlüssel $\tilde{k}_{group}$ wird vom so genannten Token Holder u_m an die Gruppe verteilt, in dem dieser mit allen Schlüssel $k_{m/\ell}$, verschlüsselt, d.h. $E(\tilde{k}_{group}, k_{m/\ell})$ mit $\ell=\{1,\ldots,U|\ell\neq m\}$, versendet wird. Der Token Holder wird aus der Versionsnummer des Gruppenschlüssels berechnet.

3.4.6 Bewertung verteilter Verfahren zur Bereitstellung von Gruppenschlüsseln

Für die Gruppenschlüsselbereitstellung in den Streitkräften scheinen verteilte Verfahren am Besten zu eignen, da diese eine Schlüsselgeheimhaltung in dynamischen Gruppen inklusive einer Zugriffskontrolle ermöglichen (Forderungen 1,2, Tabelle 6). Außerdem sind sie reparierbar beim Ausfall von Prozessen und eine robuste Schlüsselübermittlung kann leicht

ergänzt werden (Forderungen 4,5, Tabelle 6). Um ein geeignetes verteiltes Verfahren aus den vorgestellten zu ermitteln, wird nun die Skalierbarkeit bei steigender Gruppengröße untersucht. Konkret wird hierzu die Anzahl der ausgetauschten Nachrichten beim Schlüsselwechsel verglichen (Tabelle 11, Spalte 3,4). Hierbei wird für die auf einem Schlüsselbaum basierenden Verfahren TGDH, DLKH und DOFT angenommen, dass nach der Operation JOIN bzw. LEAVE ein vollständig besetzter Schlüsselbaum existiert, bei dem die Pfadlänge aller Blätter gleich ist.

Verfahren	Teilnehmer-operation	Anzahl der Unicast-Nachrichten	Anzahl der Multicast-Nachrichten	Anzahl der krypto. Operationen
ITW	JOIN / LEAVE	U·(U-1) / U·(U-1)	0 / 0	-
BD	JOIN / LEAVE	0 / 0	2·U+2/ 2·U-2	-
STW	JOIN / LEAVE	1 / 0	1 / 1	U+1 / U-1
2^d-cube	JOIN / LEAVE	U·d / U·d	0 / 0	-
TGDH	JOIN / LEAVE	0 / 0	2 / 1	$2\cdot\log_2 U$ / $2\cdot\log_2 U$
STR	JOIN / LEAVE	0 / 0	2 / 1	2 / 2·U-2
DLKH	JOIN / LEAVE	2 / $\log_2 U$	1 / $\log_2 U$	-
DOFT	JOIN / LEAVE	$\log_2 U$ / $\log_2 U$	$\log_2 U$ / $\log_2 U$	-
DFT	JOIN / LEAVE	$\log_2 U$ / $\log_2 U$	$\log_2 U$ / $\log_2 U$	-
VTKD	JOIN / LEAVE	4 / 1	1 / 1	-

Tabelle 11: Komplexität der verteilten Verfahren zur Bereitstellung von Gruppenschlüsseln

Aus Tabelle 11 wird die bessere Leistungsfähigkeit der Verfahren STW, TGDH und STR bei steigender Gruppengröße sichtbar. Zusätzlich zu dem Kommunikationsaufwand wird die Schlüsselbereitstellung in großen Gruppen durch die iterative Anwendung des rechenleistungsintensiven DH-Algorithmus verzögert. Zur Verdeutlichung des Zeitbedarfs des DH-Algorithmus wurde dessen Berechnungszeit gemessen und in Tabelle 12 für verschiedene Schlüssellängen zusammengefasst. Die Berechnung eines Knotens im Schlüsselbaum ergibt sich aus der Summe der in Tabelle 12 zusammengefassten Berechnungszeiten, d.h. der Summe der letzten beiden Spalten. Zur weiteren Auswahl eines geeigneten Verfahrens wird deshalb bei den Verfahren STW, TGDH und STR die maximale Anzahl der kryptographischen Operationen, d.h. Anzahl der Anwendungen des DH-Algorithmus, die ein Nutzer zur Ermittlung des Gruppenschlüssels durchführt, verglichen (Tabelle 11, Spalte 5). Der Vergleich zeigt, dass das Verfahren TGDH insbesondere bei der Teilnehmeroperation LEAVE die geringste maximale Anzahl an kryptographischen Operationen aufweist. Dieses Verfahren wird deshalb zunächst für die Schlüsselbereitstellung in einem militärischen Einsatzumfeld favorisiert.

Schlüssellänge des DH-Algorithmus	Dauer der Berechnung des gemeinsamen Schlüssels	Dauer der Berechnung des Blindschlüssels
256 Bit	0,4 ± 0,1 ms	0,5 ± 0,2 ms
512 Bit	1,1 ± 0,4 ms	2,2 ± 0,7 ms

Tabelle 12: Dauer der Schlüsselberechung eines Knotens mit dem exponentiellen (regulären) DH-Algorithmus bei Verwendung der Crypto++ Library (Intel Pentium 2 GHz Prozessor)

Bei der bisherigen Analyse wurde nicht berücksichtigt, dass die meisten dezentralen Verfahren für den Betrieb ein Gruppenkommunikationssystem (Group Communication System, GCS) benötigen [Kim00][Lia06][Rod00]. Da das Verfahren TGDH zunächst zur Schlüsselbereitstellung favorisiert wird, wird es einer ausführlichen Analyse unterzogen. Für den Betrieb von TGDH wird von den Entwicklern des Verfahrens ein Gruppenkommunikationsdienst, der die Semantik synchronisierte Gruppensicht (Virtual Synchrony, vgl. Abschnitt 2.2.4) bereitstellt, vorausgesetzt [Kim00]. Dieses wurde in Tabelle 11 nicht berücksichtigt. Nachfolgend wird eine theoretische Bewertung von TGDH in Verbindung mit dem Gruppenkommunikationssystem Totem (vgl. Abschnitt 2.2.4) vorgenommen. Bei der Bewertung wird die in Tabelle 13 zusammengefasste Notation verwendet.

U	Anzahl der Nutzer
S	Anzahl der Sender, die eine Nachricht mittels des Gruppenkommunkationssytems versenden
t_{Join}	Übertragungszeit der Join-Nachricht des Mitgliedschaftsdiensts (Totem-M-Protokoll)
s_{Join}	Größe der Join-Nachricht des Mitgliedschaftsdiensts (Totem-M-Protokoll)
$t_{C\text{-}Token}$	Übertragungszeit des Commit-Tokens (Totem-M-Protokoll)
$s_{C\text{-}Token}$	Größe des Commit-Tokens (Totem-M-Protokoll)
$t_{C\text{-}Token\text{-}Ack}$	Übertragungszeit der Quittung des Commit-Tokens (Totem-M-Protokoll)
$t_{R\text{-}Token}$	Übertragungszeit des Regular-Token des zuverlässigen Multicast-Diensts (Totem-SR-Protokoll)
$s_{R\text{-}Token}$	Größe des Regular-Tokens (Totem-SR-Protokoll)
$\Delta t_{Totem\text{-}Membership}$	Zeitbedarf des Gruppenmitgliedschaftsdiensts für die Feststellung der Teilnehmer
$\Delta t_{TGDH\text{-}Join}$	Zeitbedarf für die Bereitstellung des Gruppenschlüssels durch TGDH

Tabelle 13: Notationen für die theoretische Bewertung von TGDH in Verbindung mit dem Gruppenkommunikationssystem Totem

Hierzu wird die Zeit für den Beitritt eines Nutzers zu einer sicheren Gruppenkommunikation untersucht. Bei dieser Betrachtung wird die für die Schlüsselbereitstellung benötigte Rechenleistung vernachlässigt. Weiterhin werden für das Totem-M-Protokoll und das Totem-SR-Protokoll die folgenden vereinfachenden Annahmen getroffen:

- Zur Festlegung der Teilnehmer durch den Gruppenmitgliedschaftsdienst beim Beitritt eines neuen Prozesses in die Gruppe mit U-1 Prozessen, muss jeder Prozess nur einmal eine Join-Nachricht des Totem-M-Protokolls versenden. Diese enthält eine Liste der beteiligten Prozesse. Die Übertragungszeit der Nachricht t_{Join} ist deshalb abhängig von der Gruppengröße. Bei U Teilnehmern wird für die Join-Nachricht die Größe s_{Join}=160 Bit+U·32 Bit angenommen.
- Zwischen benachbarten Prozessen im logischen Ring soll das Commit-Token des Totem-M-Protokolls zuverlässig übertragen werden. Zu diesem Zweck wird das Token mittels UDP übertragen und quittiert. Hierfür wird die Zeit $t_{C\text{-}Token}+t_{C\text{-}Token\text{-}Ack}$ benötigt. Das Token enthält eine Liste der Gruppenteilnehmer. Aus diesem Grund ist die Übertragungszeit des Commit-Tokens $t_{C\text{-}Token}$ beim ersten Umlauf abhängig von der Gruppengröße. Bei U Teilnehmern wird für das Commit-Token die Größe $s_{C\text{-}Token}$=192 Bit+U·32 Bit angenommen.

- Im Zustand Recover des Totem-M-Protokolls müssen keine alten Nachrichten mehr übertragen werden. Deshalb wird für das Commit-Token beim zweiten Umlauf eine konstante Größe angenommen.
- Mit der Funktion $g(m_i,d)$ kann die Übertragungszeit für die Nachricht i (Nutzdaten) eines Senders in Abhängigkeit von der Datenübertragungsrate d und der Nachrichtengröße m_i abgeschätzt werden. Im Gegensatz zu den anderen Übertragungen werden die Multicast-Übertragungen der Nutzdaten als fehlerhaft angenommen, d.h. ein gewisser Prozentsatz der potentiellen Empfänger in der Gruppe wird nicht erreicht. Mit der Funktion f(x, e) kann die Anzahl der Prozesse, die einen Multicast nicht bekommen haben, in Abhängigkeit von der Zahl potentieller Empfänger x und der Fehlerrate e errechnet werden. Die Funktion wird durch die nachfolgende Gleichung bestimmt: $f(x,e) = x \cdot \frac{e}{100}$
- Bei fehlerhafter Multicast-Übertragung wird die Nachricht nachgesendet und quittiert. Die Nachsendung wird als fehlerfrei angenommen. Aus Sicht des einzelnen Totem-Prozesses muss das Regular-Token deshalb zweimal den logischen Ring durchlaufen (Abbildung 39). Im ersten Umlauf vermerkt ein Prozess, bei dem ein Nachrichtenverlust aufgetreten ist, eine Anforderung zur Übertragungswiederholung. Aus diesem Grund ist die Größe des Tokens im ersten Umlauf von der Fehlerrate abhängig. Diese Abhängigkeit wird vernachlässigt, d.h. die Übertragungszeit des Tokens $t_{R\text{-}Token}$ ist konstant. Für die Größe des Regular-Tokens wird $s_{R\text{-}Token}$=192 Bit angenommen.
- Zwischen benachbarten Totem-Prozessen im logischen Ring soll das Token des Totem-SR-Protokolls zuverlässig übertragen werden. Zu diesem Zweck wird das Token mittels UDP übertragen und quittiert. Hierfür wird die Zeitdauer $2 \cdot t_{R\text{-}Token}$ benötigt. Das Regular-Token besitzt eine sehr geringe Größe und damit Übertragungszeit. Zur Vereinfachung wird deshalb nicht zwischen der Token-Übertragung und der Quittung unterschieden.
- Alle Prozesse, die eine Multicast-Nachricht senden, sind im logischen Ring benachbart. Damit S Sender die Berechtigung zum Nachrichtenversand erhalten, wird die Zeitdauer $S \cdot 2 \cdot t_{R\text{-}Token}$ benötigt.

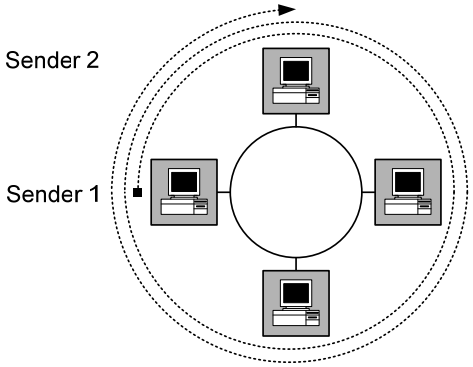

Abbildung 39: Wechsel des Regular-Tokens bei fehlerhafter Multicast-Übertragung

Die Datenübertragungszeit $\Delta t_{JoinAll}$ für den Beitritt eines Prozesses bzw. Nutzers zu einer sicheren Gruppenkommunikation setzt sich zusammen aus der Zeitdauer $\Delta t_{Totem\text{-}Membership}$ des Gruppenmitgliedschaftsdienst und der Zeitdauer $\Delta t_{TGDH\text{-}Join}$ für die Schlüsselübertragung durch das Verfahren TGDH.

$$\Delta t_{JoinAll} = \Delta t_{Totem-Membership} + \Delta t_{TGDH-Join}$$

Zur Berechnung der Übertragungszeit für die Bereitstellung eines Gruppenschlüssels durch das Verfahren TGDH wird zunächst allgemein die Zeitdauer $\Delta t_{Totem\text{-}Data}$ für die Übertragung der Datenmenge $M=\sum m_i$ mit dem Gruppenkommunikationssystem Totem abgeschätzt. Die gesamte Datenübertragungszeit besteht aus der Zeit Δt_{Data} für die eigentliche Datenübertragung und der Zeit $\Delta t_{Totem\text{-}Ack}$ für die Quittierung durch das Totem-SR-Protokoll. Die Zeitdauer des Quittungsmechanismus verursacht durch den Wechsel des Regular-Tokens bei einer Anzahl von S Sendern und U Nutzern lässt sich mit $\Delta t_{Totem\text{-}Ack}=2\cdot t_{R\text{-}Token}\cdot(2\cdot U+(S-1))$ abschätzen. Der Zeitbedarf für die Nutzdatenübertragung Δt_{Data} ergibt sich aus dem Zeitbedarf der Multicast-Übertragung sowie den f(U-1,e) Unicast-Nachsendungen. Insgesamt ergibt sich damit für die Verzögerung:

$$\Delta t_{Totem-Data} = \Delta t_{Totem-Ack} + \Delta t_{Data}$$

$$\Delta t_{Totem-Data} = 2 \cdot t_{R-Token} \cdot (2 \cdot U + (S-1)) + \sum_{i=1}^{s} g(m_i,d) + f(U-1,e) \cdot g(m_i,d)$$

Nun wird die oben genannte Formel für das Verfahren TGDH konkretisiert und die Zeitdauer $\Delta t_{TGDH\text{-}Join}$ für die Gruppenschlüssel berechnet. Die Berechnung wird für Schlüssel mit einer Länge von 512 Bit durchgeführt. Für den Beitritt des Nutzers u_U zur sicheren Gruppenkommunikation mit U-1 Nutzern sind zwei Sender notwendig. Diese versenden eine Beitrittsanfrage der Größe m_{jr} und eine Nachricht der Größe m_{ku} mit dem aktualisierten Schlüsselbaum. Die oben genannte Formel kann umgeformt werden zu:

$$\Delta t_{TGDH-Join} = 2 \cdot t_{R-Token}(2 \cdot U + 1) + g(m_{jr},d) + f(U\text{-}1,e) \cdot g(m_{jr},d) + g(m_{ku},d) + f(U\text{-}1,e) \cdot g(m_{ku},d)$$

Die Zeitdauer $\Delta t_{Totem\text{-}Membership}$ für die Aufnahme des Nutzers u_U durch den Mitgliedschaftsdienst des Gruppenkommunikationssystems in den logischen Ring mit U-1 Nutzern lässt sich mit der nachfolgenden Formel abschätzen:

$$\Delta t_{Totem-Membership} = \underbrace{2 \cdot U \cdot t_{Join}}_{\text{Term 1}} + \underbrace{2 \cdot U \cdot (t_{C-Token} + t_{C-Token-Ack})}_{\text{Term 2}}$$

Mit dieser Formel wird der Zeitbedarf des Totem-M-Protokolls für den günstigen Fall abgeschätzt. Im ungünstigen Fall werden im Rahmen des Totem-M-Protokolls $1+U\cdot(U-1)$ Join-Nachrichten anstatt der in Term 1 angegebenen Anzahl auszutauschen.

Unter Verwendung der erstellten Formeln für die Zeitdauer $\Delta t_{Totem\text{-}Membership}$ und $\Delta t_{TGDH\text{-}Join}$ wird nun die Verzögerungszeit auf Grund des Kommunikationsaufwands für den Beitritt berechnet. Diese ergibt sich aus der Addition der beiden Zeiten. Das Ergebnis der Berechnung ist in Abbildung 40 dargestellt. Wie bereits erwähnt, sind bei den dargestellten Zeiten die Berechnungskosten, die bei der Bereitstellung eines Gruppenschlüssels anfallen, nicht berücksichtigt. Die Abbildung zeigt, dass in Netzwerken mit hoher Übertragungsrate erst bei großen Teilnehmerzahlen die Aufnahme eines neuen Nutzers in den logischen Ring von Totem eine erhebliche Verzögerung bei der Schlüsselbereitstellung bewirkt. In Netzwerken mit einer geringen Übertragungsrate verlängert sowohl die Aufnahme eines neuen Nutzers in den logischen Ring als auch die Quittierung aller Managementnachrichten die Schlüsselbereitstellung.

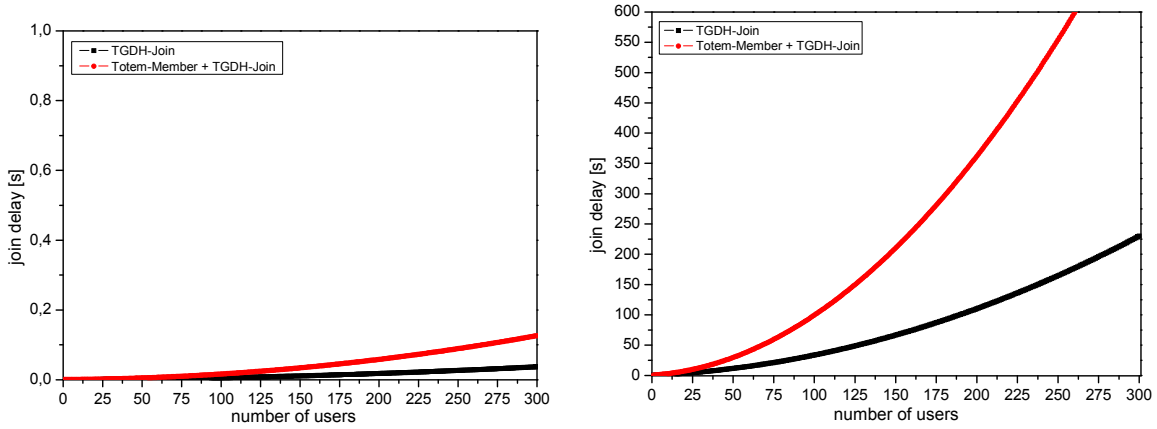

Abbildung 40: Berechnung der Verzögerungszeit auf Grund des Kommunikationsaufwands für den Beitritt eines Nutzers für die Kommunikationsinfrastruktur Ethernet (links) und VHF (rechts)

Zusammenfassung der Bewertung: Auf den ersten Blick scheinen sich verteilte Verfahren für die Gruppenschlüsselbereitstellung in den Streitkräften zu eignen, da diese die Forderungen Schlüsselgeheimhaltung, Zugriffskontrolle, robuster Schlüsselwechsel und Reparierbarkeit erfüllen (Forderungen 1,2,4,5, Tabelle 6). Bei der Untersuchung der Skalierbarkeit zeigt sich, dass die verteilten Verfahren auch in kleinen Gruppen gut nutzbar sind. Bei einer theoretischen Untersuchung wurde festgestellt, dass mit den Verfahren Skinny Tree Protocol (STR) und Tree-based Group Diffie-Hellman (TGDH), auch bei steigender Gruppengröße ein effizienter Schlüsselwechsel durchgeführt werden kann. Wird allerdings berücksichtigt, dass für den Betrieb ein Gruppenkommunikationssystem vorhanden sein muss, so tritt bei großen Teilnehmerzahlen bzw. in Netzwerken mit einer geringen Übertragungsrate wegen des Quittungsmechanismus des Gruppenkommunikationssystems eine erhebliche Verzögerung der Schlüsselbereitstellung auf. Eine weitere Auswirkung des Gruppenkommunikationssystems besteht darin, dass bei temporären Verbindungsverlusten jedes Mal der Gruppenmitgliedschaftsdienst ausgeführt wird und dann keine Gruppenkommunikation möglich ist. Die Forderung 3, Tabelle 6 ist deshalb nur bedingt erfüllt. Außerdem benötigt ein Gruppenkommunikationssystem für die Kommunikation Duplexverbindungen, da Datenübertragungen durch einen Quittungsmechanismus bestätigt werden. Aus diesem Grund ist ein Betrieb mit Nutzern im Zustand EMCON nicht möglich (Forderung 7, Tabelle 6). Der Einsatz der verteilten Verfahren zur Bereitstellung von Gruppenschlüsseln in Kombination mit IPSec (Forderung 9, Tabelle 6) ist bei keinem Verfahren Bestandteil des Konzepts, aber eine Adaption wäre möglich. Diese würde dann die Gruppenschlüsselbereitstellung zu einer GSA-Bereitstellung für IPSec erweitern. Die Bewertung der verteilten Verfahren zur Bereitstellung von Gruppenschlüsseln ist in Tabelle 14 zusammengefasst.

Verfahren/ Anforderung	1 Schlüssel-geheim-haltung	2 Zugriffs-kontrolle	3 Skalier-barkeit	4 Robuster Schlüssel-wechsel	5 Reparier-barkeit	6 Mehrfach-anfragen	7 EMCON	8 IPSec-Adaption	9 Realzeit
ITW	+	0	-	+	+	0	-	0	+
BD	+	0	-	+	+	-	-	0	+
STW	+	0	-	+	+	-	-	0	+
2^d-cube	+	0	-	+	+	-	-	0	+
TGDH	+	0	-	+	+	-	-	0	+
STR	+	0	-	+	+	-	-	0	+
DLKH	+	+	-	+	+	0	-	0	+
DOFT	+	+	-	+	+	0	-	0	+
DFT	+	+	-	+	+	0	-	0	+
VTKD	+	+	-	+	+	0	-	0	+

+ erfüllt 0 prinzipiell erfüllbar - nicht erfüllt ? unklar

Tabelle 14: Bewertung verteilter Verfahren zur Bereitstellung von Gruppenschlüsseln

3.4.7 Resümee der Bewertung existierender Schlüsselbereitstellungsverfahren

In diesem Teilkapitel wird die Bewertung der existierenden Verfahren zur Bereitstellung von Gruppenschlüsseln aus den Abschnitten 3.4.2, 3.4.4 und 3.4.6 zusammengefasst. Einen Überblick über die vorgestellten Mechanismen zur Schlüsselbereitstellung vermittelt die Abbildung 41.

Mit den meisten zentralen Verfahren zur Schlüsselbereitstellung ist es möglich, Gruppen, die einer Veränderung in Gruppengröße und Zusammensetzung unterliegen, mit Schlüsselmaterial zu versorgen und Schlüsselgeheimhaltung zu gewährleisten. Die meisten zentralen Verfahren weisen außerdem eine gute Skalierbarkeit auf, d.h. sie sind in kleinen Gruppen einsetzbar und ermöglichen bei steigender Gruppengröße einen effizienten Schlüsselwechsel. Es existieren zum Teil auch Konzepte zur Verarbeitung von Mehrfachanfragen und eine Schlüsselbereitstellung an Nutzer im Zustand EMCON ist möglich bzw. kann einfach hinzugefügt werden. Ein schwerwiegender Nachteil aller dieser Verfahren dieser Kategorie besteht darin, dass sie durch den GCKS einen Single Point of Failure aufweisen und so die geforderte Reparierbarkeit des Systems nicht erfüllt ist. Außerdem kann der GCKS die Gruppe nicht verlassen.

Mittels hierarchischer Verfahren kann ebenfalls eine dynamische Schlüsselbereitstellung unter Einhaltung der Vorschriften für den Schlüsselwechsel erfolgen. Durch die Aufteilung der Gruppe in Teilgruppen wird insbesondere die Effizienz des Schlüsselwechsels bei einer großen Teilnehmerzahl gewährleistet. Weiterhin wird die Problematik mit Schlüsselbereitstellung an einem Ort vermindert und damit ein gewisses Maß an Reparierbarkeit gewährleistet. Nachteil dieser Aufteilung besteht darin, dass die hierarchischen Verfahren nicht in kleinen Gruppen einsetzbar sind. Durch Übersetzung der verschlüsselten Multicast-Nutzdaten bzw. der Sitzungsschlüssel besitzen einige hierarchische Verfahren außerdem den Nachteil, dass sie für eine Gruppenschlüsselbereitstellung zum Schutz von Realzeit-Datentransfer nur bedingt geeignet sind.

Auf den ersten Blick scheinen verteilte Verfahren für die Gruppenschlüsselbereitstellung in den Streitkräften am Besten geeignet, da diese eine Gruppenschlüsselgeheimhaltung in dynamischen Gruppen ermöglichen und reparierbar sind beim Ausfall von Prozessen. Der Kommunikationsaufwand des Gruppenkommunikationssystems, das verteilte Verfahren benötigen, bewirkt allerdings, dass insbesondere in Netzwerken mit beschränkter Datenübertragungskapazität bei steigender Gruppengröße diese Verfahren an ihre Grenzen stoßen. Zusätzlich sind Verfahren nicht einsetzbar, wenn sich ein Teil der Nutzer im Zustand EMCON befindet.

Die Kernaussage des Resümees besteht darin, dass existierende Verfahren zur Bereitstellung von Gruppenschlüsseln nicht gleichzeitig eine ausreichende Reparierbarkeit im Fehlerfall, eine Skalierbarkeit hinsichtlich der unterstützten Gruppengröße und eine Schlüsselbereitstellung für Nutzer im Zustand EMCON bieten.

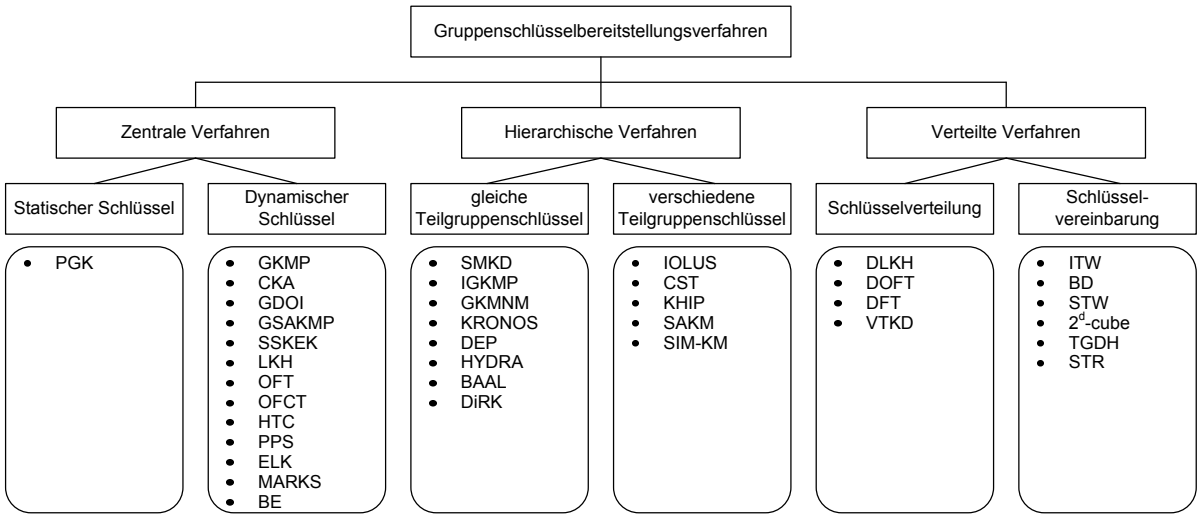

Abbildung 41: Graphischer Überblick über die vorgestellten Schlüsselbereitstellungsverfahren

3.5 Kapitelzusammenfassung

Nachdem im vorherigen Kapitel die Notwendigkeit von Sicherheitsdiensten für Multicast motiviert und das Schlüsselmanagement als fundamentale Komponente identifiziert wurde, erfolgt in diesem Kapitel die Herausarbeitung der Anforderungen an ein Schlüsselmanagementsystem. In der Literatur definierte grundlegende Anforderungen wurden um Anforderungen, die aus der Nutzung des Schlüsselmanagements in den Streitkräften resultieren, erweitert. Im Anschluss daran wurden bereits existierende Forschungsarbeiten präsentiert und die Erfüllung der gestellten Anforderungen bewertet. Eine nicht ausreichende Reparierbarkeit im Fehlerfall in Kombination mit einer Skalierbarkeit hinsichtlich der unterstützten Gruppengröße und eine Schlüsselbereitstellung für Nutzer im Zustand EMCON bei existierenden Schlüsselverwaltungskonzepten sind die Motivation für die Entwicklung eines neuartigen Konzepts im Rahmen der vorliegenden Arbeit. Dieses wird in dem nächsten Kapitel dargestellt.

4 Schlüsselmanagementkonzept MIKE

In diesem Kapitel wird das Konzept MIKE zur Bereitstellung eines Gruppenschlüssels vorgestellt. Dieses umfasst eine Beschreibung der grundlegenden Architektur und deren Teilkomponenten. Weiterhin wird die Kommunikation der MIKE-Prozesse untereinander erläutert. Hauptaufgabe von MIKE ist die Gruppenschlüsselbereitstellung für IPSec. Allerdings wurde das Konzept so flexibel entworfen, dass auch andere Sicherheitsdienste mit Schlüsselmaterial versorgt werden können. Voraussetzung für den Betrieb von MIKE ist eine Kommunikationstechnologie, die den Unicast- und den Multicast-Dienst bereitstellt.

4.1 Idee: Basiskonzept Schlüsselbaum

Zur dynamischen Gruppenschlüsselbereitstellung wird ein auf einem DH-Algorithmus basierendes verteiltes Schlüsselbereitstellungsverfahren eingesetzt und als Betriebsmodus Key Agreement bezeichnet. Der DH-Algorithmus wird hierbei als ein Beispiel für ein Verfahren, bei dem aus geheimem Schlüssel und Blindschlüssel einen gemeinsamen Schlüssel ermitteln, interpretiert. Dadurch, dass jeder Teilnehmer die Aufgabe des Gruppenschlüsselverwalters übernehmen kann, ist das System im Fehlerfall reparabel. Somit ist die Forderung nach einem reparierbaren Schlüsselmanagement (Forderung 5, Tabelle 6) mit einem teilnehmersensitiven Schlüsselwechsel (Forderung 1, Tabelle 6) aus Abschnitt 3.3 erfüllt. Durch die Einführung des zweiten Betriebsmodus Key Distribution wird eine alternative Schlüsselbereitstellung ermöglicht, falls der Betriebsmodus Key Agreement für die Durchführung eines Schlüsselwechsels eine zu große Zeitspanne benötigt. Dieses kann auftreten, wenn die Gruppengröße wächst und die Teilnehmer nur über beschränkte Ressourcen, d.h. Rechenleistung und Datenübertragungskapazität, verfügen. Somit wird mit dem vorgestellten Konzept MIKE durch die Einführung der zwei Betriebsmodi Key Agreement und Key Distribution die Bereitstellung eines Gruppenschlüssels bei größeren Gruppen skalierbar (Forderung 3, Tabelle 6). Die Idee des Konzeptes besteht darin, dass beide Betriebsmodi auf einem Schlüsselbaum basieren [Aur04]. Eine formale Definition für den in beiden Schlüsselbereitstellungsverfahren eingesetzten Schlüsselbaum wurde hierzu bereits in Abschnitt 3.1 eingeführt. Auf dem zum Schlüsselmanagement verwendeten Schlüsselbaum gibt es drei Grundoperationen, das Suchen eines Knotens (`Find`), das den Baum unverändert lässt, das Einfügen (`Insert`) sowie das Löschen eines Knotens (`Delete`). Außerdem sind zwei Betriebsmodus-spezifische Operationen auf dem Baum notwendig. Diese Operationen sind das Wechseln der Schlüssel (`GetUpdateKeys`) und die Aktualisierung der Schlüssel (`SetUpdateKeys`). Durch die Verwendung der Methodik Schlüsselbaum wird die zentrale Forderung einer ressourcenschonenden und schnellen Verwaltung des Gruppenschlüssels erfüllt. Außerdem wird ein Betriebsmoduswechsel ohne erneuten Gruppenaufbau möglich, falls die Gruppe wächst und der Modus Key Agreement zu viele Ressourcen benötigt.

Eine Konsequenz des Konzepts besteht in einer alternativen Klassifizierung von Schlüsselmanagementverfahren auf der Grundlage der verwendeten Datenstruktur. Hierdurch ergibt sich eine Einteilung in die Kategorien listen-basierte, tabellen-basierte und baum-basierte Systeme (Abbildung 42). Das System MIKE kann der Kategorie der baum-basierten Schlüsselmanagementsysteme zugeordnet werden.

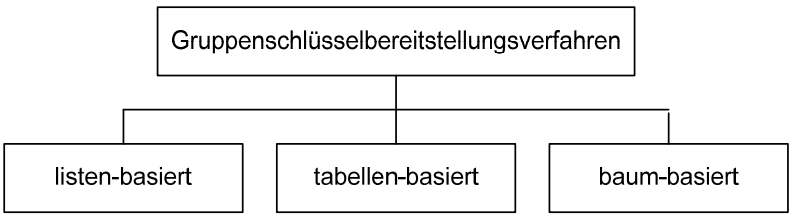

Abbildung 42: Alternative Klassifizierung von Schlüsselmanagementverfahren

4.2 Vorteile des Konzeptes

Die bisherigen Forschungsanstrengungen im Themenfeld Gruppenschlüsselmanagement konzentrieren sich auf die verteilte Schlüsselbereitstellung in Dynamic Peer Groups. In Abschnitt 3.4.5 wurden derartige Verfahren vorgestellt. Wie am Beispiel von Tree-based Group-Diffie-Hellman (TGDH) gezeigt wurde, stoßen solche Verfahren insbesondere in Netzwerken mit beschränkter Datenübertragungskapazität bei steigender Gruppengröße an ihre Grenzen. Alternativ existieren zentrale Schlüsselbereitstellungsverfahren für große Gruppen. In Abschnitt 3.4.1 wurden derartige Verfahren präsentiert. Bisher noch nicht betrachtet wurde die Schlüsselbereitstellung in beiden Gruppenarten. Insbesondere fehlt ein Schlüsselmanagementkonzept für Dynamic Peer Groups, die sich während ihrer Existenz derart verändern, dass der Einsatz eines verteilten Verfahrens zur Schlüsselbereitstellung nicht mehr sinnvoll ist. Diese Veränderung kann verursacht werden, durch den Beitritt einer Vielzahl von Teilnehmern oder durch den Beitritt von Teilnehmern mit geringer Datenübertragungskapazität. Bedingt dadurch, dass derartige Gruppenveränderungen noch nicht betrachtet wurden, wurden bisher zentrale und verteilte Verfahren zur Bereitstellung von Gruppenschlüsseln getrennt untersucht. Mit dem vorgestellten Konzept wird gezeigt, dass eine getrennte Betrachtung von Schlüsselmanagementsystemen nach zentralem und verteiltem Verfahren nicht sinnvoll ist und sich beide Schlüsselbereitstellungsverfahren kombinieren lassen. Wird die Datenstruktur Schlüsselbaum eingesetzt, unterscheiden sich die beiden Verfahren nur in dem kryptographischen Mechanismus, wie die Teilnehmer aus den übertragenen Hilfsschlüsseln den Gruppenschlüssel erhalten. Synergieeffekt der gemeinsamen Betrachtung beider Schlüsselbereitstellungsarten ist eine einfache Realisierung beider Ansätze und die Möglichkeit, zwischen diesen umzuschalten. Somit werden die Vorteile beider Verfahren, d.h. die ressourcenschonende Schlüsselbereitstellung von Key Distribution und die Reparierbarkeit von Key Agreement, genutzt. Eine konsequente Verfolgung des Ansatzes ermöglicht die gleichzeitige Optimierung beider Betriebsmodi, indem die Verarbeitung des Schlüsselbaums verbessert wird. Durch derartige in Kapitel 7 vorgestellte Optimierungen werden die Forderungen nach Mechanismen aus Abschnitt 3.3 zur Schlüsselbereitstellung in Gruppen mit Zeiträumen, in denen eine starke Fluktuation auftritt und mit Nutzern, die nur über eine Simplexverbindung verfügen, erfüllt (Forderung 6,7, Tabelle 6).

4.3 Module des Konzepts

Dieser Abschnitt beschreibt die Module des Schlüsselmanagements MIKE. In Abbildung 43 ist zusätzlich zu den Modulen eine Einordnung von MIKE in die IPSec-Systemarchitektur

dargestellt. Das Schlüsselmanagement MIKE besteht aus vier Modulen mit der nachfolgend beschriebenen Funktionalität [Aur03]:

- Modul `MessageDispatcher`
  Das Modul `MessageDispatcher` ist für die Abwicklung der Kommunikation verantwortlich. Zur Interaktion mit anderen MIKE-Prozessen besitzt das Modul eine Schnittstelle zum Netzwerk. Weiterhin sind Schnittstellen zum Auslösen von Teilnehmeroperationen notwendig. Das MIKE-Konzept stellt hierzu drei verschiedene Möglichkeiten bereit:
  (1) Über eine Kommandozeile kann von einem Administrator der Gruppenbeitritt bzw. Gruppenaustritt initiiert werden.
  (2) Zeitgesteuerte Teilnehmeroperationen werden durch die Verarbeitung einer vor Beginn der sicheren Kommunikation erstellten Ereigniswarteschlange ermöglicht. Inhalt dieser Warteschlange ist ein Tripel bestehend aus einem Zeitstempel, der IP-Adresse des Ausführenden und dem Typ der durchzuführenden Teilnehmeroperation. Die Ereigniswarteschlange zur Initiierung von Gruppenbeitritt bzw. Gruppenaustritt kann gut für die Durchführung von Messungen genutzt werden.
  (3) Um einen automatisierten Gruppenbeitritt bzw. -austritt zu ermöglichen, ist in MIKE eine Schnittstelle zum IPSec Discovery Protocol (IDP) [Sei06] vorgesehen. Dieses durchsucht ein Netzwerk nach Prozessen, die einen Schutz der Multicast-Nutzdaten mit IPSec ermöglichen. Wird MIKE in Kombination mit IDP verwendet, wird automatisch ein Gruppenbeitritt initiiert, wenn IDP entsprechende Prozesse detektiert. Wird im umgekehrten Fall mittels IDP das Ende einer Multicast-IPSec-Kommunikation bekannt gegeben, so wird automatisch die Gruppe verlassen.
  Neben den oben beschriebenen Schnittstellen offeriert das Modul zwei Schnittstellen für die Übergabe des verwalteten Gruppenschlüssels bereit:
  (1) Über die PF_KEY-Schnittstelle [McD98] können Umfang und Eigenschaften des Nutzdatenschutzes in die Datenbanken Security Policy Database (SPD) und Security Association Database (SAD) des Betriebssystemkerns gespeichert werden (Abbildung 43). Basierend auf den in SAD und SPD definierten Regeln fügt das IPSec-Kernel-Modul in die Pakete der Anwendungsprotokolle die IPSec-Sicherheitsprotokolle AH und ESP in die Nutzdatenpakete ein und schützt so die übertragenen Daten.
  (2) Außerdem kann über eine Dateischnittstelle auch Anwendungsprogrammen der Gruppenschlüssel zur Verfügung gestellt werden. Als Beispiel wurde die paketbasierte Sprachkommunikation PC-Phone [Lan02] gewählt, bei der mit Hilfe des Gruppenschlüssels die Sprachkommunikation geschützt wird.

- Modul `GroupManagementFramework`
  Die Aufgabe des Moduls `GroupManagementFramework` besteht darin, dem Modul `MessageDispatcher` eine einheitliche Schnittstelle zum Modul `KeyManagement` zur Verfügung zu stellen. Das Modul `GroupManagementFramework` startet bei eintreffenden Ereignissen oder Nachrichten das Modul `KeyManagement` im entsprechenden Betriebsmodus.

- Modul `KeyManagement`
  Das Modul `KeyManagement` ist die zentrale Komponente von MIKE. Es enthält die

Mechanismen für die Gruppenschlüsselbereitstellung beider Betriebsmodi. Für deren Realisierung werden Zustandsautomaten eingesetzt. Dies ist in verteilten Systemen eine übliche Methode zur Realisierung von Diensten. Die Zustandsautomaten erzeugen bzw. verarbeiten die zur Bereitstellung des Gruppenschlüssels notwendigen Nachrichten und Ereignisse. Nach einer Nachrichten- oder Ereignisbearbeitung ändern sie ihren Zustand. Das Modul `KeyManagement` nutzt das verbindungslose User Datagram Protocol (UDP) der Internet-Technologie zum Nachrichtenaustausch mit anderen Mike-Prozessen. Der zum effizienten Schlüsselmanagement notwendige Schlüsselbaum wird ebenfalls in diesem Modul verwaltet.

- Modul `GroupPolicyDatabase`
  Die Module `KeyManagement` und `GroupManagementFramework` haben Zugriff auf das Modul `GroupPolicyDatabase`. Dieses wertet die Gruppensicherheitsvorschrift aus und sorgt für deren Einhaltung. Die Gruppensicherheitsvorschrift enthält einen Satz von Regeln zur Festlegung des sicherheitsrelevanten Verhaltens und die Berechtigungen zum Gruppenbeitritt. Die Gruppensicherheitsvorschrift muss vor dem Betrieb von MIKE in einem administrativen Vorgang erstellt werden. Um zu gewährleisten, dass nur autorisierte Teilnehmer der Gruppe beitreten, werden Authentisierungsmethoden eingesetzt, die auf digitalen Signaturen und den Mechanismen einer Public Key Infrastructure basieren.

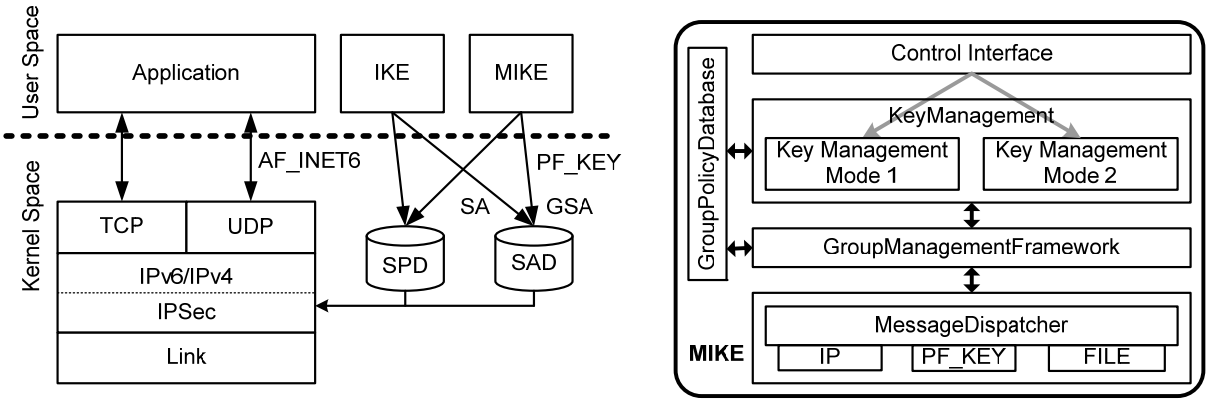

Abbildung 43: MIKE als Teil der IPSec-Architektur (links) und die Module von MIKE (rechts)

4.4 Modul KeyManagement

Nachdem im vorherigen Abschnitt ein Überblick über alle Module von MIKE gegeben wurde, wird in diesem Abschnitt die wichtigste Komponente eines Schlüsselmanagements, das Modul `KeyManagement`, detaillierter erläutert. Hierbei wird die Gruppenschlüsselbereitstellung mittels eines dezentralen und eines zentralen Verfahrens dargestellt. Beide Ansätze wurden bereits im Multicast-Sicherheit-Referenzframework, das die Basiselemente für Multicast-Sicherheit identifiziert, als Möglichkeiten der Gruppenschlüsselbereitstellung beschrieben (vgl. Abschnitt 2.5). Das dezentrale Verfahren wird durch den Betriebsmodus Key Agreement realisiert, während das zentrale Verfahren als Betriebsmodus Key Distribution bezeichnet wird. Die Eigenschaften der beiden Verfahren sind in Tabelle 15 zusammengefasst.

Betriebsmodusbezeichnung	Key Agreement	Key Distribution
Schlüsselerzeugung	verteilt	zentral
Beitrag zum Gruppenschlüssel	jeder Nutzer	der Schlüsselverwalter
Schlüsselverwalter	dynamisch	statisch

Tabelle 15: Eigenschaften der Betriebsmodi

Die folgende Darstellung der beiden Betriebsmodi umfasst eine Beschreibung der zum Schlüsselwechsel eingesetzten kryptographischen Verfahren sowie der verwendeten Managementnachrichten. Nachdem die beiden Betriebsmodi vorgestellt wurden, wird der Mechanismus zum Umschalten von Key Agreement auf Key Distribution beschrieben.

Vor einer detaillierten Erläuterung der Betriebsmodi wird das Rahmenkonzept des verwendeten Schlüsselmanagements vorgestellt. Zur Schlüsselverwaltung wird ISAKMP für die Verwendung in Gruppen konkretisiert. Dieses ist ein universelles Schlüssel- bzw. Sicherheitsmanagementprotokoll und wird bereits beim Schutz einer Punkt-zu-Punkt-Kommunikation mittels IPSec eingesetzt. ISAKMP verwaltet abstrakte Sicherheitsassoziationen (Security Association, SA) für alle Schichten des OSI-Modells. Eine SA legt Umfang, Eigenschaften und den Schlüssel sicherer Verbindungen fest. Wie bereits erwähnt, wird meistens sprachlich nicht zwischen einer SA und dem enthaltenen Schlüssel differenziert. In MIKE werden, wie in Abschnitt 2.5.2 beschrieben, drei Arten von Sicherheitsassoziationen bzw. Schlüsseln, verwendet:

- Kategorie-1-SA bzw. Kategorie-1-Schlüssel
  Im Modus Key Distribution legt die Kategorie-1-SA den Umfang und die Eigenschaften der sicheren Verbindung zwischen GC und Nutzer fest. Der Kategorie-1-Schlüssel wird auch als Individualschlüssel bezeichnet. Im Modus Key Agreement existiert zu jedem Teilnehmer der Gruppenkommunikation eine Kategorie-1-SA.
- Kategorie-2-SA bzw. Kategorie-2-Schlüssel
  Die Kategorie-2-SA legt Umfang, Eigenschaft und Schlüssel, die zum Gruppenschlüsselwechsel verwendet werden, fest.
- Kategorie-3-SA bzw. Kategorie-3-Schlüssel
  Die Kategorie-3-SA legt Umfang und Eigenschaften der sicheren Verbindungen fest, die zum Austausch der Nutzdaten verwendet werden. Mit dem Kategorie-3-Schlüssel wird der Nutzdatenverkehr geschützt. In dem vorgestellten Konzept wird der Kategorie-3-Schlüssel aus dem Kategorie-2-Schlüssel abgeleitet.

Genau genommen stellt ISAKMP nur ein Protokollformat bereit, in dem verschiedene Nachrichtentypen definiert sind. Mit diesen Nachrichtentypen kann dann das eigentliche Schlüsselmanagementprotokoll aufgebaut werden. ISAKMP-Nachrichten werden mit dem verbindungslosen UDP transportiert. Vor jeder Nachricht wird ein ISAKMP-Nachrichtenkopf platziert. Teile dieses Nachrichtenkopfs, z.B. Cookies und Message-ID, ermöglichen trotz des verbindungslosen Transportprotokolls das Halten eines minimalen Verbindungskontextes. Der Rest ist zur Bearbeitung der dem Nachrichtenkopf folgenden Nutzlastfelder notwendig. Die dem ISAKMP-Nachrichtenkopf folgenden Nutzlastfelder, die so genannten Payloads,

werden durch das Protokoll bestimmt. Die nachfolgenden Nutzlastfelder aus RFC 2408 [Mau98] und RFC 3547 [Har03] werden eingesetzt (Abbildung 44):

- Nutzlastfeld `NONCE`
  Der Zufallswert, den dieses Nutzlastfeld überträgt, wird zum Schutz gegen Angriffe durch Übertragungswiederholung eingesetzt.
- Nutzlastfeld `SIG`
  Die digitale Signatur des Nutzlastfelds `SIG` wird zur Gewährleistung der Authentizität und Integrität der ISAKMP-Nachrichten genutzt.
- Nutzlastfeld `SA`
  Das Nutzlastfeld `SA` setzt sich aus den Bestandteilen SA Key Encryption Key (SAK) und SA Traffic Encryption Key (SAT) zusammen. Das Nutzlastfeld SA KEK wird zum Transport der Kategorie-2-SA verwendet. Mit Hilfe der SA TEK werden Kategorie-3-SA transportiert.
- Nutzlastfeld `KD`
  Das Nutzlastfeld `KD` (Key Download, KD) wird zum Übertragen der in den SAs eingesetzten Schlüssel verwendet. Wird ein Schlüsselbaum als Hilfsmittel zur Verwaltung dieser Schlüssel eingesetzt, so enthält das Nutzlastfeld die Hilfsschlüssel des Schlüsselbaums mit den zugehörigen Verwaltungsinformationen. Hilfsschlüssel und Verwaltungsinformationen werden zu so genannten Schlüsselpaketen zusammengefasst. Das Nutzlastfeld KD ist somit eine aus Schlüsselpaketen bestehende ungeordnete Liste.
- Nutzlastfeld `SEQ`
  Die mit dem Nutzlastfeld `SEQ` (Sequence Number, SEQ) transportierte Sequenznummer wird dazu eingesetzt, eine Reihenfolge für die Nachrichten des Schlüsselwechsels zu gewährleisten.
- Nutzlastfeld `TMD`
  Das im Rahmen der Arbeit zusätzlich zu den bestehenden Nutzlastfeldern definierte Feld `TMD` (TM Download, TMD) wird ausschließlich im Modus Key Agreement verwendet und dient zur Übermittlung der IP-Adresse des Transaction Managers.

Das in MIKE verwendete Protokoll besitzt drei Phasen (Abbildung 44). Die ausgetauschten Nachrichten sind zur Gewährleistung der Authentizität mit einer digitalen Signatur versehen. Diese wird im Nutzlastfeld `SIG` transportiert. Die Mechanismen der Authentizitätsprüfung werden von dem Modul `GroupPolicyDatabase` bereitgestellt und sind in Abschnitt 0 beschrieben. In der Phase I werden die Nachrichten `p1JoinRequest`, `p1JoinDistribute` und `p1JoinConfirm` zur Anmeldung verwendet. Die im Nutzlastfeld `NONCE` enthaltenen Zufallszahlen bieten Schutz vor Angriffen durch wiederholtes Senden. Im Rahmen des Anmeldevorgangs wird durch das Modul `GroupPolicyDatabase` die Berechtigung zum Gruppenbeitritt geprüft. Die Phase entspricht dem bereits existierenden Protokoll IKE Aggressive Mode. Nachdem ein neuer Teilnehmer die Phase I erfolgreich absolviert hat, wird diesem in der Phase II zur Initialisierung des Schlüsselwechsels die Nachricht `p2Distribute` zugesandt. Hauptziel dieser Initialisierung ist die Etablierung der in der Gruppe verwendeten Sequenznummer. Der

Schlüsselwechsel selbst wird in der Phase III mit den Nachrichten `p3TMDistribute` und `p3UpdateDistribute` vollzogen. Diese werden per Multicast an alle Teilnehmer der Gruppe versandt. Die Nachrichten enthalten eine Sequenznummer, die bei jedem Versand der Nachrichten `p3TMDistribute` bzw. `p3UpdateDistribute` erhöht wird. Die in den Phase-III-Nachrichten enthaltene Sequenznummer wird dazu eingesetzt, eine Reihenfolge für die Nachrichten des Schlüsselwechsels zu gewährleisten. Weiterhin dient die Sequenznummer in der Phase III zum Schutz gegen wiederholtes Senden. Die Nachricht `p3TMDistribute` wird nur im Betriebsmodus Key Agreement eingesetzt. Zum Gruppenaustritt werden die Nachrichten `p1LeaveRequest` und `p1LeaveConfirm` verwendet. Der Versand einer Bestätigung des Gruppenaustritts sichert die Anfrage zum Gruppenaustritt gegen Verlust ab.

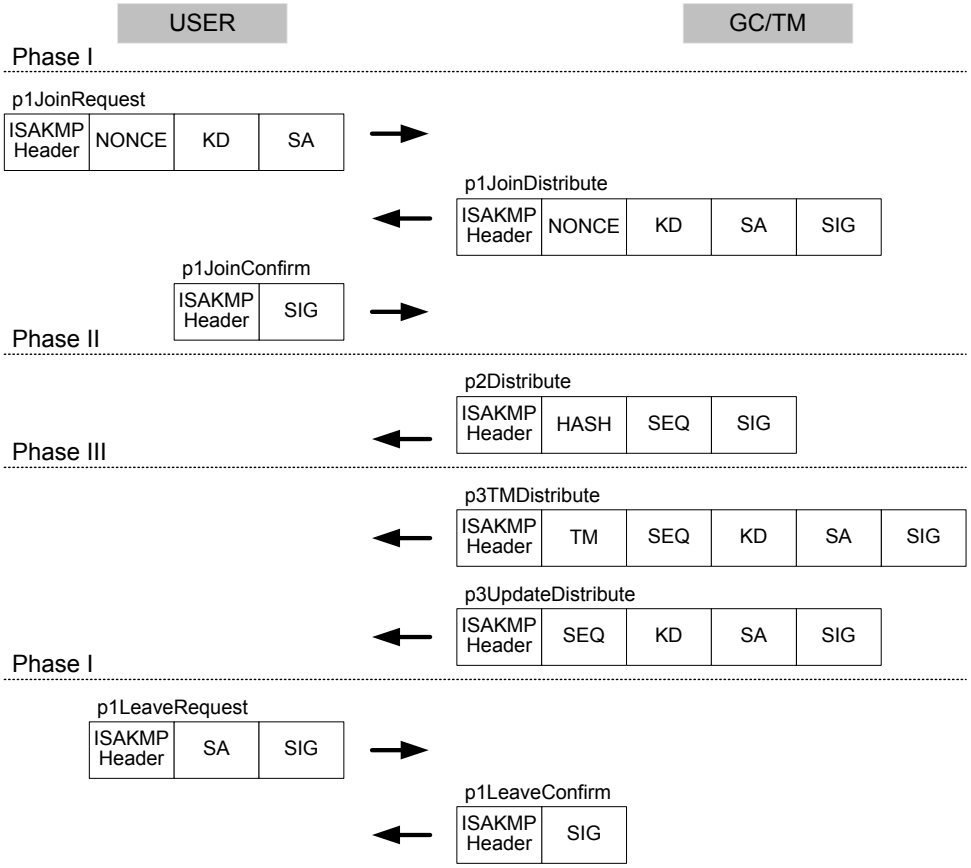

Abbildung 44: Aufbau der ISAKMP-Nachrichten in MIKE

Nachdem das Rahmenkonzept des Schlüsselbereitstellungsprotokolls von MIKE vorgestellt wurde, werden im nächsten Abschnitt die Verfahren des Schlüsselwechsels erläutert. Der Fokus hierbei liegt auf den in den drei Sicherheitsassoziationskategorien verwendeten Schlüsseln. Deshalb sind im weiteren Verlauf der Arbeit die für die Durchführung des Schlüsselwechsels zu übertragenden Hilfsschlüssel und die Ermittlung des Gruppenschlüssels Hauptbestandteil der Erläuterung. Hierbei werden die in Tabelle 2 und Tabelle 3 zusammengefassten Notationen verwendet.

4.4.1 Betriebsmodus 1 (Key Agreement)

Prinzipiell kann der Betriebsmodus Key Agreement des Schlüsselmanagements MIKE mittels TGDH realisiert werden. Allerdings wurde in Abschnitt 3.4.5 nachgewiesen, dass das Verfahren TGDH sehr viele Netzwerkressourcen benötigt. Deshalb war der Entwurf einer Variante des Verfahrens notwendig. Diese neu entwickelte TGDH-Variante benötigt einen Transaction Manager (TM). Die Aufgaben des TM können von jedem Teilnehmer der Gruppe übernommen werden. Der TM stellt nur sicher, dass auftretende Nutzeranfragen in einer eindeutigen Reihenfolge bearbeitet werden und ermöglicht eine verteilte Schlüsselbereitstellung, ohne dass eine Gruppensemantik durch ein Kommunikationssystem garantiert wird. Außerdem werden durch das neu entworfene Verfahren zusätzliche Informationen bereitgestellt, die den Einsatz von IPSec zum Schutz von Multicast-Datenverkehr ermöglichen.

Funktionsprinzip:

Im Modus Key Agreement berechnet jeder Teilnehmer den Gruppenschlüssel eigenständig mittels eines Schlüsselbaums vom Grad zwei in Kombination mit einer wiederholten Anwendung des DH-Algorithmus. Auf diese Weise wird das in Abschnitt 3.1 beschriebene Schlüsselmanagement der Punkt-zu-Punkt-Kommunikation für Gruppen erweitert. Der Schlüsselbaum wird im Modus Key Agreement in der nachfolgend beschriebenen Art aufgebaut. Jedem Knoten $v_{\ell,p}$ wird ein Schlüssel $k_{\ell,p}$ und der aus diesem abgeleitete Blindschlüssel $bk_{\ell,p}$ zugeordnet (vgl. Abschnitt 3.1). Funktionsprinzip des Verfahrens ist, dass ein Nutzer bei Kenntnis aller Blindschlüssel des Baums die Schlüssel des Pfades zur Wurzel mittels des DH-Algorithmus berechnen kann. Der Schlüssel $k_{\ell,p}$ des Knotens $v_{\ell,p}$, eines Baums mit der Höhe h, dem linken Nachfolger $v_{\ell+1,2p}$ und dem rechten Nachfolger $v_{\ell+1,2p+1}$ kann rekursiv wie folgt berechnet werden:

$$k_{\ell,p} = DH(bk_{\ell+1,2p},k_{\ell+1,2p+1}) = DH(bk_{\ell+1,2p+1},k_{\ell+1,2p}) \quad \text{mit } \ell=0,\ldots,h\text{-}1 \text{ und } p=0,\ldots,p^*_{max}$$

Insbesondere kann der Schlüssel im Wurzelknoten, welcher der Gruppenschlüssel ist, auf diese Art ermittelt werden. Es ist anzumerken, dass der Modus Key Agreement auf einem kryptographischen Algorithmus beruht, bei dem aus geheimem Schlüssel und Blindschlüssel einen gemeinsamer Schlüssel ermitteln wird und nicht konkret z.B. den exponentiellen DH-Algorithmus fordert.

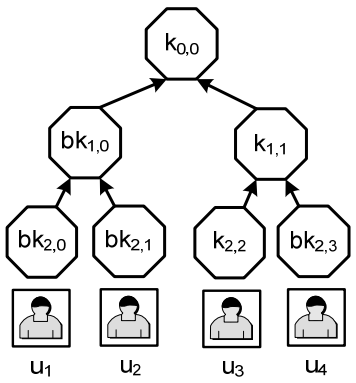

Abbildung 45: Schlüsselbaum beim Modus Key Agreement aus Sicht des Nutzers u_3

Zum Beispiel berechnet der Nutzer u_3 den Gruppenschlüssel, indem er zweimal hintereinander den DH-Algorithmus anwendet (Abbildung 45). Beim ersten Mal erfolgt dies unter Verwendung des Blindschlüssels $bk_{2,3}$ des Nutzers u_4 und des eigenen privaten Schlüssels $k_{2,2}$, d.h. $k_{1,1}=DH(bk_{2,3},k_{2,2})$. Die zweite Berechnung mit dem DH-Algorithmus erfolgt mit dem Ergebnis der ersten Berechnung, d.h. dem Schlüssel $k_{1,1}$ und dem Blindschlüssel $bk_{1,0}$ der Teilgruppe u_1, u_2. Als Resultat erhält man den Gruppenschlüssel $k_{0,0}=DH(bk_{1,0},DH(bk_{2,3},k_{2,2}))$.

Der Rest dieses Abschnitts gibt einen Überblick über die im Modus Key Agreement unterstützten Teilnehmeroperationen JOIN, MERGE, LEAVE/EJECT und PARTITION. Die Teilnehmeroperationen werden nachfolgend zuerst formal beschrieben und dann durch Beispiele veranschaulicht. Hierbei wird für aktualisierte Schlüssel des Knotens $v_{\ell,p}$ die Bezeichnung $\tilde{k}_{\ell_k,p_k}$ verwendet. Die Zustandsautomaten zur Realisierung des Verfahrens sind im Anhang dargestellt.

Beitritt eines Nutzers (JOIN):

Zur Beschreibung des Ablaufs der Teilnehmeroperation JOIN wird der Beitritt des Nutzers u_{U+1} zu der bereits bestehenden Gruppe $G=\{u_1,\ldots,u_U\}$ aus U Teilnehmern betrachtet. Zur Verwaltung der Gruppe wird der Schlüsselbaum KT mit Grad zwei und Höhe h eingesetzt.

(1) Zur Gruppenanmeldung wird ein 3-Wege-Anmeldevorgang zwischen dem TM und dem neuen Nutzer u_{U+1} durchgeführt, d.h. die Nachrichten `p1JoinRequest`, `p1JoinDistribute` und `p1JoinConfirm` ausgetauscht. Die Nachricht `p1JoinRequest` enthält den Blindschlüssel $bk_{\ell_{U+1},p_{U+1}}$ des Nutzers u_{U+1} und wird an die Multicast-Gruppe, die zum Schlüsselmanagement verwendet wird, versandt. Der Blindschlüssel wurde aus dem entsprechend der Vorgaben des DH-Algorithmus zufällig erzeugten (geheimen) Schlüssel $k_{\ell_{U+1},p_{U+1}}$ abgeleitet. Nach dem Anmeldevorgang wird beim Nutzer u_{U+1} der Schlüsselwechsel durch die Nachricht `p2Distribute` initialisiert.

(2a) Der neue Nutzer wird als Knoten $v_{\ell_{U+1},p_{U+1}}$ in den Schlüsselbaum eingefügt. Anschließend werden die Schlüssel und Blindschlüssel der Knoten $\{w_{\ell_k,p_k} \mid w_{\ell_k,p_k}\in K\tilde{T} \wedge w_{\ell_k,p_k}\in path(v_{\ell_{U+1},p_{U+1}})\}$ des neuen Baums $K\tilde{T}=KT\cup bk_{\ell_{U+1},p_{U+1}}$ gelöscht.

(2b) Aus der Struktur des neuen Schlüsselbaums $K\tilde{T}$ ermittelt der TM den Nutzer u_{U+1} als neuen TM.
(2c) Mit der Multicast-Nachricht `p3TMDistribute` wird der Nutzer u_{U+1} zum neuen TM erklärt. Außerdem werden die Blindschlüssel $bk_{\ell,p}$ der Knoten $\{v_{\ell_k,p_k} \mid v_{\ell_k,p_k}\in K\tilde{T} \setminus v_{\ell_k,p_k}\in path(v_{\ell_{U+1},p_{U+1}})\}$ verteilt.

(3a) Der neue TM aktualisiert den Schlüsselbaum $K\tilde{T}$ (Operation `GetUpdateKeys`):
for each $w_{\ell_k,p_k}\in path(v_{\ell_{U+1},p_{U+1}})$ do

if $w_{\ell_{k-1},p_{k-1}} = = w_{\ell_k+1,2p_k}$ then do
$\tilde{k}_{\ell_k,p_k}=DH(bk_{\ell_k+1,2p_k+1},\tilde{k}_{\ell_{k-1}+1,2p_k})$, $b\tilde{k}_{\ell_k,p_k}=BK(\tilde{k}_{\ell_k,p_k})$
else do
$\tilde{k}_{\ell_k,p_k}=DH(bk_{\ell_k+1,2p_k},\tilde{k}_{\ell_k+1,2p_k+1})$, $b\tilde{k}_{\ell_k,p_k}=BK(\tilde{k}_{\ell_k,p_k})$,

(3b) Anschließend verteilt der neue TM die Blindschlüssel $bk_{\ell,p}$ der Knoten $\{v_{\ell_k,p_k} \mid v_{\ell_k,p_k} \in K\tilde{T} \wedge v_{\ell_k,p_k} \in path(v_{\ell_{U+1},p_{U+1}}) \setminus v_{0,0}\}$ an die Nutzer $\{u_1,\ldots,u_U\}$ mit der Multicast-Nachricht `p3UpdateDistribute`.

(4) Die Nutzer $\{u_1,\ldots,u_U\}$ aktualisieren den Schlüsselbaum mit den empfangenen Blindschlüsseln. Anschließend wird die aktuelle Position im Schlüsselbaum $\tilde{v}_{\ell_r,p_r}$ durch die Nutzer u_r mit r=1,…,U ermittelt und eine Berechnung des Gruppenschlüssels durchgeführt (Operation `SetUpdateKeys`):

for each $w_{\ell_k,p_k} \in path(\tilde{v}_{\ell_r,p_r})$ do

    if $w_{\ell_{k-1},p_{k-1}} = = w_{\ell_k+1,2p_k}$ then do
        $\tilde{k}_{\ell_k,p_k} = DH(b\tilde{k}_{\ell_k+1,2p_k+1}, \tilde{k}_{\ell_k+1,2p_k})$
    else do
        $\tilde{k}_{\ell_k,p_k} = DH(b\tilde{k}_{\ell_k+1,2p_k}, \tilde{k}_{\ell_k+1,2p_k+1})$

Da einem neuer Nutzer der aktuelle TM einer Gruppe nicht bekannt ist, sendet er seine Beitrittsanfrage an die Multicast-Gruppen, die zur effizienten Übermittlung der Schlüsselwechselinformationen verwendet wird. Dies bedeutet, es wird zur Schlüsselverwaltung eine Multicast-Gruppe verwendet. Als Transaction Manager können sowohl der neue Nutzer als auch sein Nachbar fungieren. Der TM-Status wird an den neuen Nutzer übergeben, weil dann bei der Teilnehmeroperation MERGE eine Nachricht gespart werden kann. In Abbildung 46 ist ein Beispiel dargestellt, bei der Nutzer u_8 einer Gruppe mit sieben Nutzern beitritt. Die Abbildung enthält Protokoll, Schlüsselbaum und Inhalt der Nachricht, die den Schlüsselwechsel übermittelt. In der Darstellung des Protokolls werden mit u(j), u(i) und u(k) Nutzer bezeichnet. Nach erfolgreichem 3-Wege-Anmeldevorgang, bei dem der Nutzer u_8 den Blindschlüssel $bk_{3,7}$ übermittelt, wird der neue Nutzer in den Baum eingefügt. Anschließend wird der Nutzer u_8 durch die Nachricht `p3TMDistribute` zum neuen TM der Gruppe erklärt. Der neue TM berechnet durch iterative Anwendung des DH-Algorithmus die Schlüssel $\tilde{k}_{2,3}$, $\tilde{k}_{1,1}$ und den Gruppenschlüssel $\tilde{k}_{0,0}$. Anschließend verteilt er die Blindschlüssel $b\tilde{k}_{2,3}$ und $b\tilde{k}_{1,1}$ mit der Nachricht `p3UpdateDistribute`. Nach dem Empfang dieser Nachricht kann jeder Nutzer den neuen Gruppenschlüssel $\tilde{k}_{0,0}$ berechnen. Der Nutzer u_1 berechnet den Gruppenschlüssel mit $\tilde{k}_{0,0} = DH(b\tilde{k}_{1,1}, k_{1,0})$.

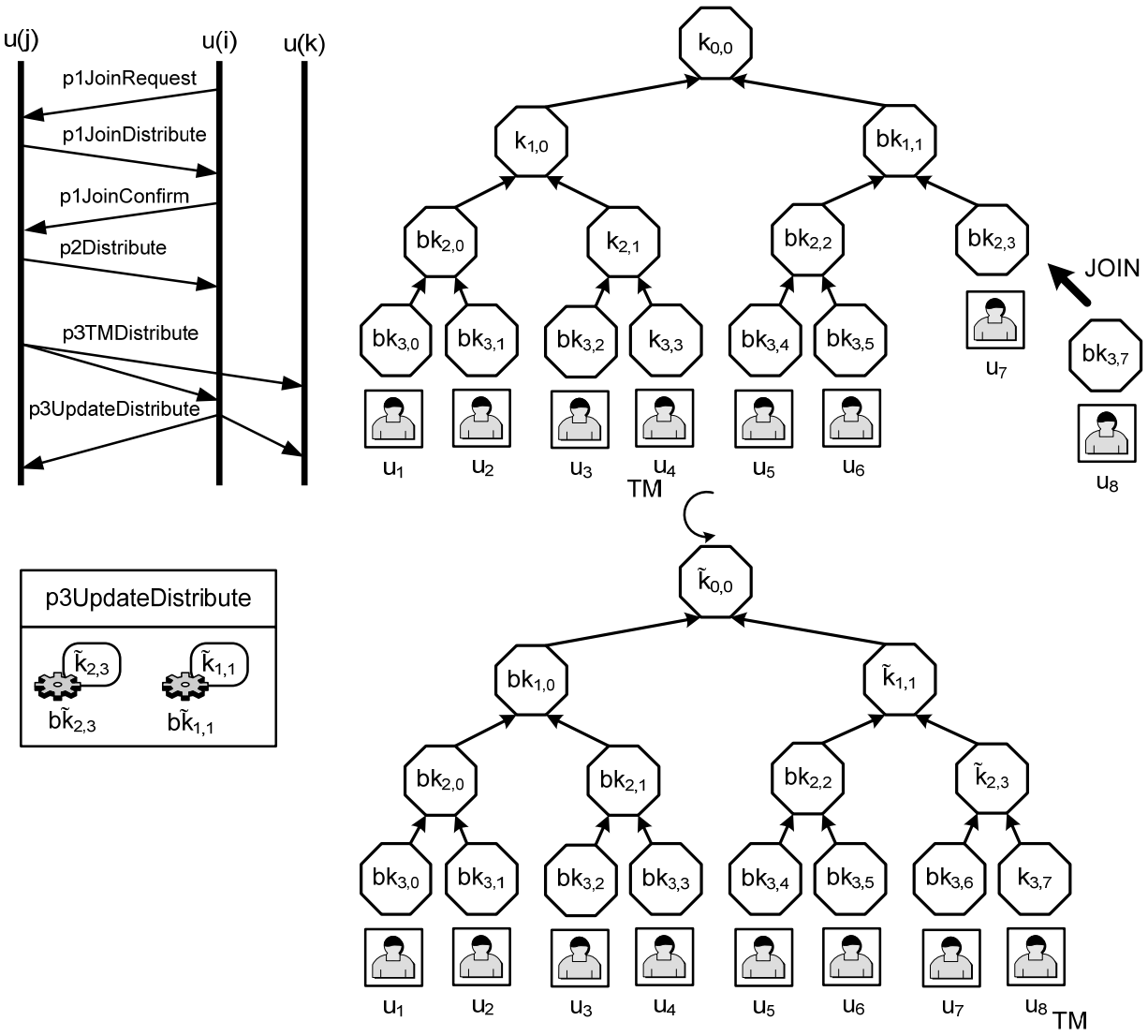

Abbildung 46: Beitritt des Nutzers u_8 beim Betriebsmodus Key Agreement

Beitritt einer Gruppe von Nutzern (MERGE):

Zur Beschreibung des Ablaufs der Teilnehmeroperation MERGE wird der Beitritt der Teilgruppe $G'=\{u_{U+1},\ldots,u_i,\ldots,u_{U'}\}$ mit U′ Teilnehmern, dem TM u_i und dem Schlüsselbaum KT′ betrachtet. Der Beitritt erfolgt zu der bereits bestehenden Gruppe $\{u_1,\ldots,u_U\}$ aus U Teilnehmern. Diese setzt zur Verwaltung den Schlüsselbaum KT mit Grad zwei und Höhe h ein.

(1) Zur Gruppenanmeldung wird ein 3-Wege-Anmeldevorgang zwischen dem TM und u_i, dem TM der Teilgruppe durchgeführt, d.h. die Nachrichten `p1JoinRequest`, `p1JoinDistribute` und `p1JoinConfirm` ausgetauscht. Bei dem Anmeldevorgang werden alle Blindschlüssel des Baumes KT′ übermittelt. Nach dem Anmeldevorgang wird beim Nutzer u_i der gemeinsame Schlüsselwechsel für beide Gruppen durch die Nachricht `p2Distribute` initialisiert.

(2a) Die neue Teilgruppe wird in den Schlüsselbaum eingefügt. In dem gemeinsamen Baum $K\tilde{T}=KT \cup KT'$ wird dem Nutzer u_i der Knoten v_{ℓ_i,p_i} zugeordnet. Löschen der Schlüssel und Blindschlüssel der Knoten $\{w_{\ell_k,p_k} \mid w_{\ell_k,p_k} \in K\tilde{T} \wedge w_{\ell_k,p_k} \in path(v_{\ell_i,p_i})\}$.

(2b) Aus der Struktur des neuen Schlüsselbaums $K\tilde{T}$ ermittelt der TM den Nutzer u_i als neuen TM.

(2c) Mit der Multicast-Nachricht `p3TMDistribute` wird der Nutzer u_i zum neuen TM erklärt. Außerdem werden die Blindschlüssel bk_{ℓ_k,p_k} der Knoten $\{v_{\ell_k,p_k} \mid v_{\ell_k,p_k} \in K\tilde{T} \setminus v_{\ell_k,p_k} \in path(v_{\ell_i,p_i})\}$ verteilt.

(3a) Der neue TM erzeugt entsprechend der Vorgaben des DH-Algorithmus den zufälligen Schlüssel $\tilde{k}_{\ell_i,p_i}$ und aktualisiert den Schlüsselbaum (Operation `GetUpdateKeys`):
for each $w_{\ell_k,p_k} \in path(v_{\ell_i,p_i})$ do

if $w_{\ell_{k-1},p_{k-1}} = = w_{\ell_k+1,2p_k}$ then do
$\tilde{k}_{\ell_k,p_k}=DH(bk_{\ell_k+1,2p_k+1},\tilde{k}_{\ell_k+1,2p_k})$, $b\tilde{k}_{\ell_k,p_k}=BK(\tilde{k}_{\ell_k,p_k})$
else do
$\tilde{k}_{\ell_k,p_k}=DH(bk_{\ell_k+1,2p_k},\tilde{k}_{\ell_k+1,2p_k+1})$, $b\tilde{k}_{\ell_k,p_k}=BK(\tilde{k}_{\ell_k,p_k})$

(3b) Anschließend verteilt der neue TM die Blindschlüssel $b\tilde{k}_{\ell,p}$ der Knoten $\{v_{\ell_k,p_k} \mid v_{\ell_k,p_k} \in K\tilde{T} \setminus v_{0,0}\}$ an die Nutzer $\{u_1,\ldots,u_{i-1},u_{i+1},\ldots,u_{U+U'}\}$ mit der Multicast-Nachricht `p3UpdateDistribute.`

(4) Die Nutzer $\{u_1,\ldots,u_{i-1},u_{i+1},\ldots,u_{U+U'}\}$ aktualisieren den Schlüsselbaum mit den empfangenen Blindschlüsseln. Anschließend wird die Position im Schlüsselbaum $\tilde{v}_{\ell_r,p_r}$ durch die Nutzer u_r mit $r=1,\ldots,i-1,i+1,\ldots,U+U'$ ermittelt, sowie eine Berechnung des Gruppenschlüssels durchgeführt (Operation `SetUpdateKeys`):
for each $w_{\ell_k,p_k} \in path(\tilde{v}_{\ell_r,p_r})$ do

if $w_{\ell_{k-1},p_{k-1}} = = w_{\ell_k+1,2p_k}$ then do
$\tilde{k}_{\ell_k,p_k}=DH(b\tilde{k}_{\ell_k+1,2p_k+1},\tilde{k}_{\ell_k+1,2p_k})$
else do
$\tilde{k}_{\ell_k,p_k}=DH(b\tilde{k}_{\ell_k+1,2p_k},\tilde{k}_{\ell_k+1,2p_k+1})$

In Abbildung 47 ist ein Beispiel dargestellt, wie eine Teilgruppe, bestehend aus den Nutzern u_8, u_9, einer Gruppe mit sieben Nutzern $\{u_1,u_2,u_3,u_4,u_5,u_6,u_7\}$ beitritt. Die Abbildung enthält Protokoll, Schlüsselbaum und Inhalt der Nachricht, die den Schlüsselwechsel übermittelt. In der Darstellung des Protokolls werden mit u(j), u(i) und u(k) Nutzer bezeichnet. Der 3-Wege-Anmeldevorgang wird durch den Nutzer u_8, den TM der Teilgruppe, abgewickelt. Dieser übermittelt beim Anmeldevorgang die Blindschlüssel $bk_{3,7}$, $bk_{4,14}$, $bk_{4,15}$. Durch den Nutzer u_4, TM der Gruppe mit sieben Nutzern, wird der Baum der Teilgruppe in den Schlüsselbaum der Gruppe $\{u_1,u_2,u_3,u_4,u_5,u_6,u_7\}$ eingefügt. Anschließend wird der Nutzer u_8 durch die Nachricht `p3TMDistribute` zum neuen TM der Gruppe erklärt. Der neue TM berechnet durch iterative Anwendung des DH-Algorithmus die Schlüssel $\tilde{k}_{3,7}$, $\tilde{k}_{2,3}$, $\tilde{k}_{1,1}$ und den Gruppenschlüssel $\tilde{k}_{0,0}$. Anschließend verteilt er alle Blindschlüssel des Schlüsselbaums mit der Nachricht `p3UpdateDistribute`. Nach dem Empfang dieser Nachricht kann jeder Nutzer den neuen Gruppenschlüssel $\tilde{k}_{0,0}$ berechnen. Der Nutzer u_1 berechnet den Gruppenschlüssel mit $\tilde{k}_{0,0}=DH(b\tilde{k}_{1,1},k_{1,0})$.

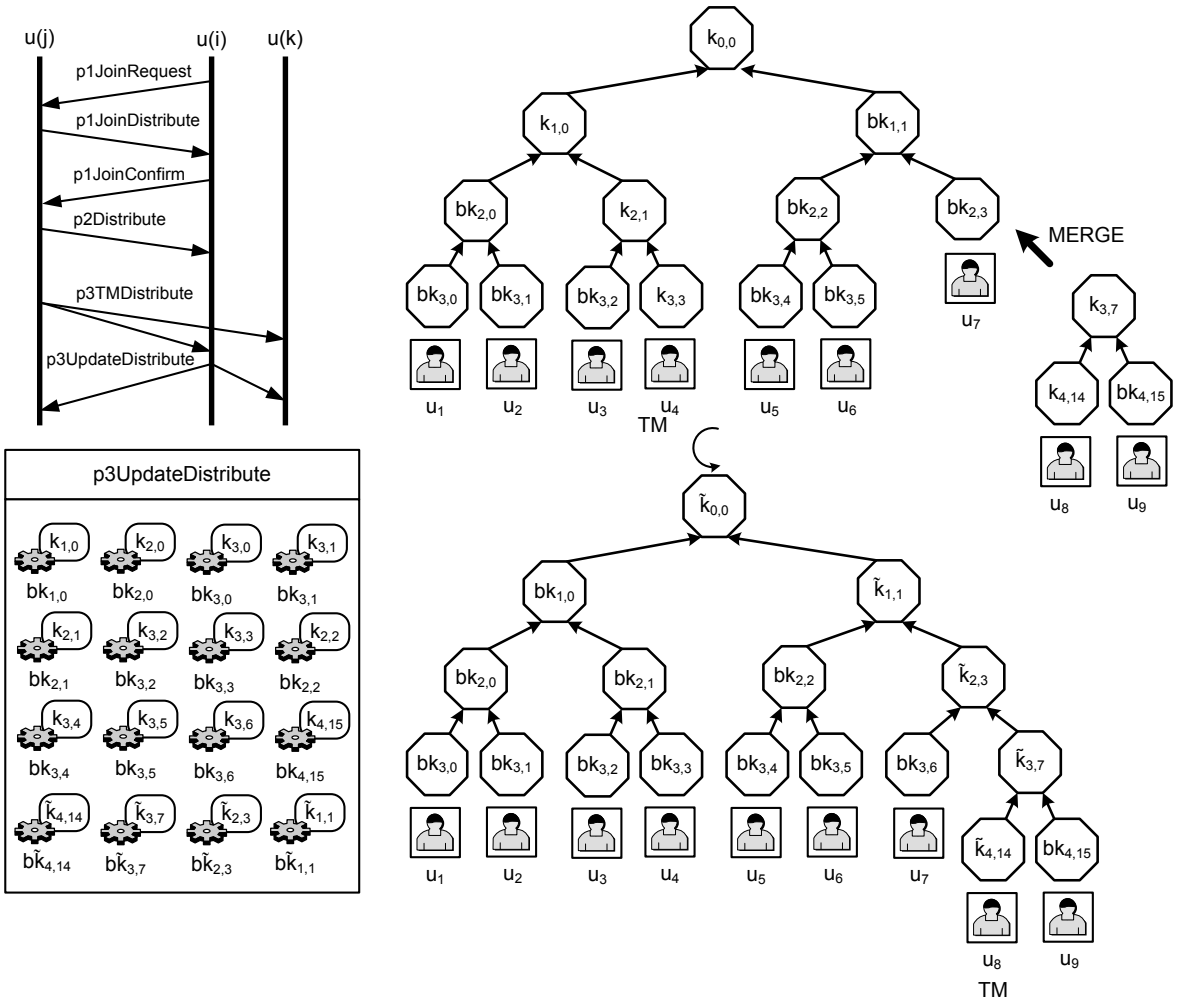

Abbildung 47: Beitritt der Teilgruppe u_8, u_9 beim Betriebsmodus Key Agreement

Austritt/Ausschluss eines Nutzers (LEAVE/EJECT):

Zur Beschreibung des Ablaufs der Teilnehmeroperation LEAVE wird der Austritt des Nutzers u_i mit $i \in \{1,\ldots,U\}$ aus der bereits bestehenden Gruppe $\{u_1,\ldots,u_U\}$ mit U Teilnehmern betrachtet. Zur Verwaltung der Gruppe wird der Schlüsselbaum KT mit Grad zwei und Höhe h eingesetzt.

(1) Zur Gruppenabmeldung wird ein 2-Wege-Abmeldevorgang zwischen dem TM und dem Nutzer u_i durchgeführt, d.h. die Nachrichten `p1LeaveRequest` und `p1LeaveConfirm` ausgetauscht.

(2a) Der dem Nutzer u_i zugeordnete Knoten v_{ℓ_i,p_i} sowie der Schlüssel und Blindschlüssel der Knoten $\{w_{\ell_k,p_k} \mid w_{\ell_k,p_k} \in K\tilde{T} \wedge w_{\ell_k,p_k} \in \text{path}(v_{\ell_i,p_i})\}$ wird gelöscht.

(2b) Aus der Struktur des neuen Schlüsselbaums $K\tilde{T}$ ermittelt der TM den Nutzer u_h als neuen TM. Für den Fall, dass der Knoten v_{ℓ_h,p_h} linker Geschwisterknoten ist, gilt h=i+1 bzw. im umgekehrten Fall h=i-1.

(2c) Mit der Multicast-Nachricht `p3TMDistribute` wird der Nutzer u_h zum neuen TM erklärt. Der Knoten v_{ℓ_h,p_h} des Nutzers u_h wird nach $\tilde{v}_{\ell_h,p_h}$ verschoben. Außerdem werden die

Blindschlüssel $bk_{\ell,p}$ der Knoten $\{v_{\ell_k,p_k} \mid v_{\ell_k,p_k} \in K\tilde{T} \setminus v_{\ell_k,p_k} \in path(\tilde{v}_{\ell_h,p_h})\}$ an die Nutzer $\{u_1,\ldots,u_{i-1},u_{i+1},\ldots,u_U\}$ verteilt.

(3a) Der neue TM generiert entsprechend der Vorgaben des DH-Algorithmus den Schlüssel $\tilde{k}_{\ell_h,p_h}$ und aktualisiert den Schlüsselbaum (Operation `GetUpdateKeys`):
for each $w_{\ell_k,p_k} \in path(\tilde{v}_{\ell_h,p_h})$ do

    if $w_{\ell_{k-1},p_{k-1}} = = w_{\ell_k+1,2p_k}$ then do
        $\tilde{k}_{\ell_k,p_k}=DH(bk_{\ell_k+1,2p_k+1},\tilde{k}_{\ell_k+1,2p_k})$, $b\tilde{k}_{\ell_k,p_k}=BK(\tilde{k}_{\ell_k,p_k})$
    else do
        $\tilde{k}_{\ell_k,p_k}=DH(bk_{\ell_k+1,2p_k},\tilde{k}_{\ell_k+1,2p_k+1})$, $b\tilde{k}_{\ell_k,p_k}=BK(\tilde{k}_{\ell_k,p_k})$

(3b) Anschließend verteilt der neue TM die Blindschlüssel $b\tilde{k}_{\ell_k,p_k}$ der Knoten $\{v_{\ell_k,p_k} \mid v_{\ell_k,p_k} \in K\tilde{T} \wedge v_{\ell_k,p_k} \in path(\tilde{v}_{\ell_h,p_h}) \setminus v_{0,0}\}$ an die Nutzer $\{u_1,\ldots,u_{i-1},u_{i+1},..,u_{h-1},u_{h+1},\ldots,u_U\}$ mit der Multicast-Nachricht `p3UpdateDistribute`.

(4) Die Nutzer $\{u_1,\ldots,u_{i-1},u_{i+1},..,u_{h-1},u_{h+1},\ldots,u_U\}$ aktualisieren den Schlüsselbaum mit den empfangenen Blindschlüsseln. Anschließend wird die Position im Schlüsselbaum $\tilde{v}_{\ell_r,p_r}$ durch die Nutzer u_r mit $r=1,\ldots,i-1,i+1,\ldots,h-1,h+1,\ldots,U$ ermittelt sowie eine Berechnung des Gruppenschlüssels durchgeführt (Operation `SetUpdateKeys`):
for each $w_{\ell_k,p_k} \in path(\tilde{v}_{\ell_r,p_r})$ do

    if $w_{\ell_{k-1},p_{k-1}} = = w_{\ell_k+1,2p_k}$ then do
        $\tilde{k}_{\ell_k,p_k}=DH(b\tilde{k}_{\ell_k+1,2p_k+1},\tilde{k}_{\ell_k+1,2p_k})$
    else do
        $\tilde{k}_{\ell_k,p_k}=DH(b\tilde{k}_{\ell_k+1,2p_k},\tilde{k}_{\ell_k+1,2p_k+1})$

Beabsichtigt der TM, die Gruppe zu verlassen, muss er zuerst den Status TM an einen anderen Nutzer übergeben. Anschließend kann dann die Teilnehmeroperation LEAVE wie oben beschrieben durchgeführt werden. Soll der TM aus der Gruppe ausgeschlossen werden, muss vorher ein Wechsel des TM auf Anfrage eines Nutzers erfolgen. Der Ablauf eines solchen Wechsels ist im übernächsten Abschnitt beschrieben.

In Abbildung 48 ist ein Beispiel dargestellt, bei dem Nutzer u_8 eine Gruppe verlässt. Die Abbildung enthält Protokoll, Schlüsselbaum und Inhalt der Nachricht, die den Schlüsselwechsel übermittelt. In der Darstellung des Protokolls werden mit u(j), u(i) und u(k) Nutzer bezeichnet. Nach erfolgreichem Abmeldevorgang wird der Nutzer aus dem Baum gelöscht. Anschließend wird der Nutzer u_7 durch die Nachricht `p3TMDistribute` zum TM der Gruppe erklärt. Dieser berechnet durch iterative Anwendung des DH-Algorithmus den Schlüssel $\tilde{k}_{1,1}$ und den Gruppenschlüssel $\tilde{k}_{0,0}$. Anschließend verteilt er die Blindschlüssel $b\tilde{k}_{2,3}$ und $b\tilde{k}_{1,1}$ mit der Nachricht `p3UpdateDistribute`. Nach dem Empfang dieser Nachricht kann jeder Nutzer den neuen Gruppenschlüssel $\tilde{k}_{0,0}$ berechnen. Der Nutzer u_1 berechnet den Gruppenschlüssel mit $\tilde{k}_{0,0}=DH(b\tilde{k}_{1,1},k_{1,0})$.

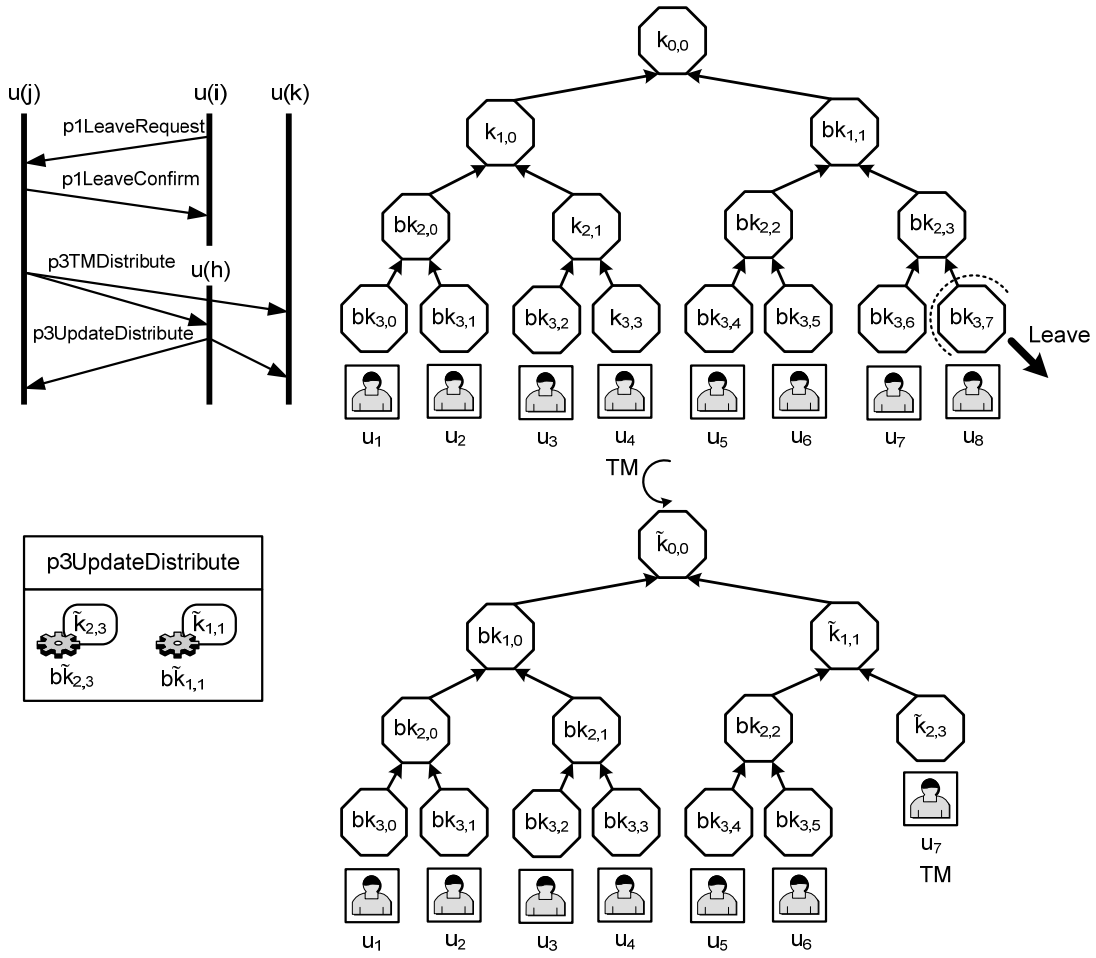

Abbildung 48: Austritt des Nutzers u_8 beim Betriebsmodus Key Agreement

Austritt einer Gruppe von Nutzern (PARTITION):

Unter der Teilnehmeroperation PARTITION wird der Austritt einer Teilgruppe von Nutzern verstanden, die einen gemeinsamen Wurzelknoten besitzen, der verschieden vom Wurzelknoten des Schlüsselbaums ist. Tritt eine Teilgruppe von Nutzern aus, die keine gemeinsame Wurzel besitzen, so wird von jedem Nutzer getrennt die Teilnehmeroperation LEAVE durchgeführt. Ein Nutzer der Teilgruppe initiiert hierzu den Gruppenaustritt. Der Protokollablauf ist identisch mit dem Ablauf der Teilnehmeroperation LEAVE.

In Abbildung 49 ist ein Beispiel dargestellt, bei dem eine Teilgruppe bestehend aus den Nutzern u_7, u_8 die Gruppe verlässt. Die Abbildung enthält Protokoll, Schlüsselbaum und Inhalt der Nachricht, die den Schlüsselwechsel übermittelt. In der Darstellung des Protokolls werden mit u(j), u(i) und u(k) Nutzer bezeichnet. Nach erfolgreichem Anmeldevorgang durch den Nutzer u_7 wird der Teilbaum mit den beiden Nutzern aus dem Baum gelöscht. Anschließend wird der Nutzer u_5 durch die Nachricht `p3TMDistribute` zum TM der Gruppe erklärt. Dieser berechnet durch iterative Anwendung des DH-Algorithmus den Schlüssel $\tilde{k}_{1,1}$ und den Gruppenschlüssel $\tilde{k}_{0,0}$. Anschließend verteilt er den Blindschlüssel $b\tilde{k}_{1,1}$ mit der Nachricht `p3UpdateDistribute`. Nach dem Empfang dieser Nachricht kann jeder Nutzer den neuen Gruppenschlüssel $\tilde{k}_{0,0}$ berechnen. Der Nutzer u_1 berechnet den Gruppenschlüssel mit $\tilde{k}_{0,0}=DH(b\tilde{k}_{1,1},k_{1,0})$.

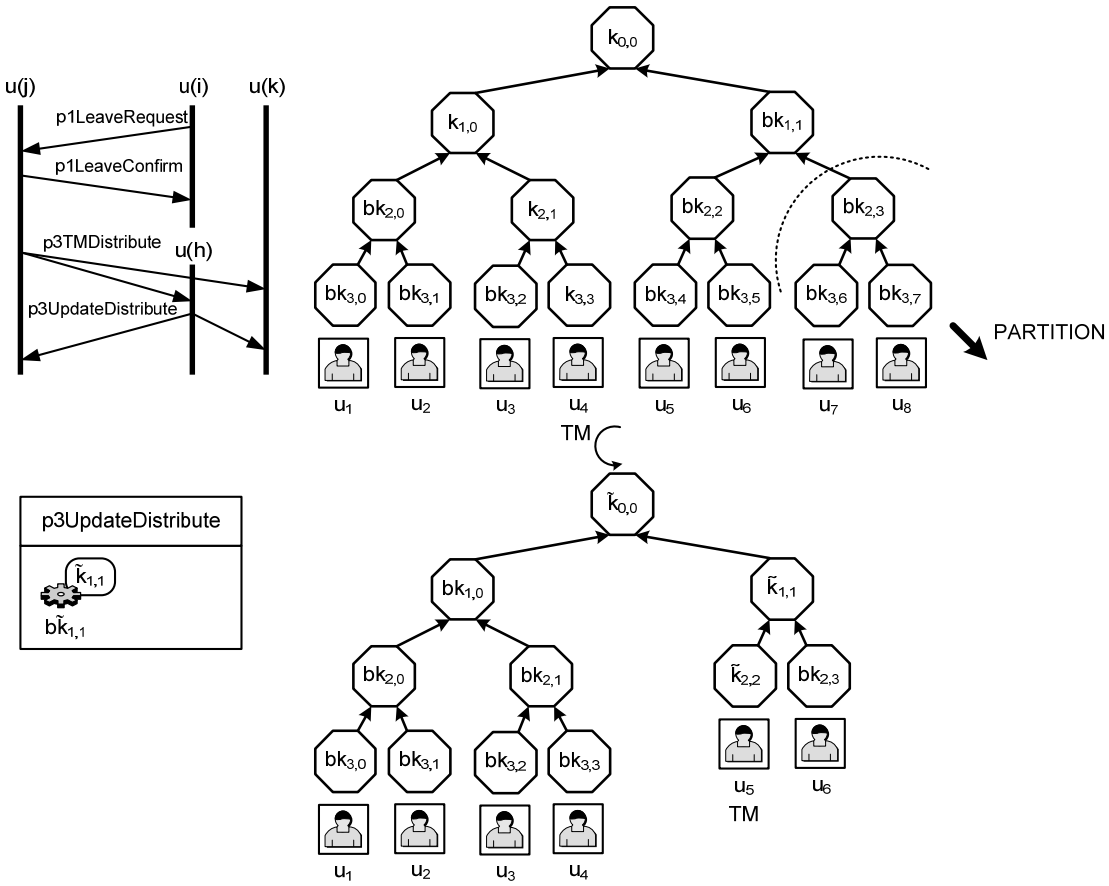

Abbildung 49: Austritt der Teilgruppe u_7, u_8 beim Betriebsmodus Key Agreement

Wechsel des TM auf Anfrage eines Nutzers:

Zur Bearbeitung von Nutzeranfragen wechselt der TM innerhalb der Gruppe. Es ist aber auch möglich, dass ein Nutzer auf Anfrage, d.h. ohne vorherige Teilnehmeroperation, TM wird. Ein derartiger Wechsel des TM kann mit dem in Abbildung 50 dargestellten Protokollablauf durchgeführt werden.

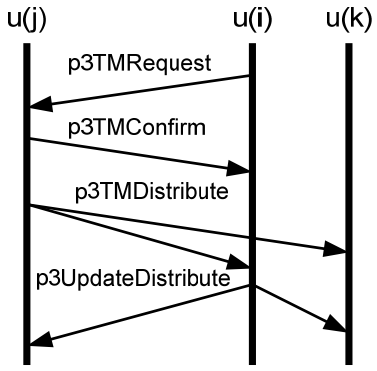

Abbildung 50: Wechsel des TM auf Anfrage eines Nutzers beim Betriebsmodus Key Agreement

Bei den Operationen JOIN und MERGE wählt der TM den Einfügepunkt für neue Nutzer bzw. neue Teilgruppen mit dem Ziel, den Schlüsselbaum zu balancieren. Die Auswirkung der Auswahl des Einfügepunktes wird in Abschnitt 7.4 untersucht. Als Aufwandsabschätzung des

Betriebsmodus Key Agreement wird in Tabelle 16 die Komplexität des Verfahrens in Abhängigkeit von der TM-Position $v_{\ell,p}$ mit der Pfadlänge $k=\|path(v_{\ell,p})\|$ beim Schlüsselwechsel angegeben. Hierbei werden die Anzahl der ausgetauschten Nachrichten, die Anzahl der kryptographischen Operationen des TM und die Anzahl der übertragenen Hilfsschlüssel betrachtet. Im Betriebsmodus Key Agreement sind die kryptographischen Operationen, die zur Ermittlung des Gruppenschlüssels notwendig sind, Exponentiationen. Bei der Ermittlung der Exponentiationen des TM ist zu beachten, dass zur Berechnung eines Knotens des Baums zwei Exponentiationen notwendig sind. Eine wird benötigt, um mit dem DH-Algorithmus den Schlüssel aus den Schlüsseln der Kinder zu berechnen, die zweite Exponentiation dient zur Erzeugung des Blindschlüssels. Bei der Komplexitätsanalyse wird angenommen, dass nach der Teilnehmeroperation ein vollständig besetzter Schlüsselbaum mit einer Nutzerzahl U existiert, bei dem die Pfadlänge aller Blätter gleich ist. In einem derartigen Baum mit dem Grad zwei sind V=2U-1 Knoten enthalten. Für die Teilnehmeroperation MERGE wird angenommen, dass zwei Teilgruppen mit der Teilnehmerzahl U′ und U″ sich zu einer gemeinsamen Gruppe zusammenschließen. Die von den Teilgruppen zur Verwaltung verwendeten Schlüsselbäume besitzen 2U′-1 bzw. 2U″-1 Knoten.

Verfahren	Teilnehmeroperation	Anzahl der Nachrichten	Anzahl der Exponentiationen des TM	Anzahl der Hilfsschlüssel
MIKE, Betriebsmodus Key Agreement	JOIN	2	2·k-1	2·U-1
	MERGE	2	2·k-1	2·(2·U′-1)+2·U″-1
	LEAVE/EJECT	2	2·k-1	k-1
	PARTITION	2	2·k-1	k-1

Tabelle 16: Komplexität des Betriebsmodus Key Agreement

In Tabelle 16 erkennt man, dass bei dem vorgestellten Verfahren die Anzahl der kryptographischen Operationen logarithmisch mit der Teilnehmeranzahl steigt. Allerdings wächst die Anzahl der übertragenen Hilfsschlüssel linear. Dieser bestehende Nachteil ist die Motivation für die Einführung eines zweiten Betriebsmodus.

4.4.2 Betriebsmodus 2 (Key Distribution)

Der Modus Key Distribution ist ein zentrales Verfahren, bei dem ein so genannter Group Controller (GC) die Schlüsselbereitstellung übernimmt. Die Bezeichnung GC ist äquivalent zur Bezeichnung GCKS, die im Rahmen des Multicast-Sicherheit-Referenzframework eingeführt worden ist. Dieser Modus kombiniert zur Bereitstellung des Gruppenschlüssels eine modifizierte Form des Konzepts Group Domain of Interpretation mit Logical Key Hierarchy (v.g.l Abschnitt 3.4.1). Das Verfahren Logical Key Hierarchy wurde ausgewählt, da es ebenfalls auf einem Schlüsselbaum basiert. Es gibt eine schlüssel-, benutzer- und gruppenorientierte Variante des Verteilungsverfahrens. Im Konzept MIKE wird nur die gruppenorientierte Variante des Verfahrens eingesetzt, weil diese nur eine Nachricht für den Schlüsselwechsel benötigt. Eine Modifikation des Konzepts Group Domain of Interpretation ist notwendig, da auf Grund des 2-Phasen-Ansatzes bei Verwendung von IKE Main Mode in der Phase I des Gruppenbeitritts neun Nachrichten notwendig sind. Um eine schnelle Anmeldung zu ermöglichen, wird IKE Aggressive Mode für die Phase I verwendet sowie die

Phase II verkürzt. Werden zum Gruppenaustritt die in IKE definierten Mechanismen eingesetzt, ist die Anfrage für den Gruppenaustritt nicht gegen Nachrichtenverlust abgesichert. Möglicherweise nicht mehr vorhandene Teilnehmer sind dann bei Nachrichtenverlust aus Sicht des GC noch Teilnehmer der Gruppe. In diesem Fall ist Forward Secrecy nicht mehr gewährleistet. Zusätzlich wird deshalb die Austrittsanfrage von GC quittiert.

Funktionsprinzip:

Soll der Gruppenschlüssel $k_{0,0}$ dynamisch durch einen GC verteilt werden, so muss jeder Teilnehmer über einen Individualschlüssel, der nur ihm und dem GC bekannt ist, verfügen. Ein Gruppenschlüssel könnte an die Teilnehmer sicher verteilt werden, indem der GC den Gruppenschlüssel $k_{0,0}$ mit dem Individualschlüssel k_{ℓ_i,p_i} jedes Nutzers u_i verschlüsselt. Dieses Verfahren wäre jedoch nicht besonders effizient. Der Gruppenschlüssel müsste hierbei mit zu vielen Individualschlüsseln verschlüsselt werden, um sicher an die Gruppe verteilt zu werden.

Das Verfahren wird effizienter, wenn die Nutzer zu Teilgruppen zusammengefasst und diesen Teilgruppen vom GC erzeugte Hilfsschlüssel zugewiesen werden. Eine Zusammenfassung von Nutzern u_f, u_h und u_i zu einer Teilgruppe und die Zuordnung des Hilfsschlüssels $k_{\ell_{fhi},p_{fhi}}$ zu diesen Gruppen kann einfach mittels eines Schlüsselbaums durchgeführt werden (Abbildung 51). Bei diesem Verfahren müssen einem Nutzer nur die jeweiligen Schlüssel des Pfades zur Wurzel bekannt sein. Durch $E(k_{\ell,p},k_{\ell_{fhi},p_{fhi}})$ ist jetzt eine Übermittlung des Schlüssels $k_{\ell,p}$ an die Teilgruppe (u_f,u_h,u_i) möglich. Betrachtet man zum Beispiel den in Abbildung 51 dargestellten Schlüsselbaum, kann durch Verschlüsselung des in der Wurzel des Schlüsselbaums befindlichen Gruppenschlüssels $k_{0,0}$ mit dem Hilfsschlüssel $k_{1,0}$, d.h. durch den Versand von $E(k_{0,0},k_{1,0})$, dieser sicher und effizient an die Teilnehmer u_1, u_2, u_3 übermittelt werden.

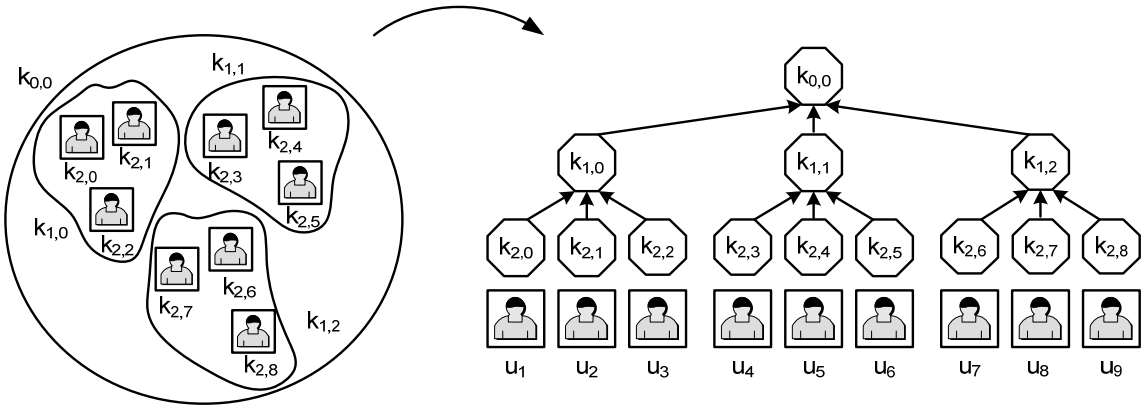

Abbildung 51: Teilgruppenbildung (links) und Schlüsselbaum (rechts) beim Betriebsmodus Key Distribution

Im Modus Key Distribution werden die Teilnehmeroperationen JOIN und LEAVE/EJECT unterstützt. Da in diesem Modus keine Interaktionen zwischen Group Controller vorgesehen sind, können die Teilnehmeroperationen MERGE und PARTITION nicht durchgeführt werden. Die Teilnehmeroperationen werden nachfolgend zuerst formal beschrieben und dann durch ein Beispiel veranschaulicht. Hierbei wird für aktualisierte Schlüssel des Knotens $v_{\ell,p}$

die Bezeichnung $\tilde{k}_{\ell_k,p_k}$ verwendet. Die Zustandsautomaten zur Realisierung des Verfahrens sind im Anhang dargestellt.

Beitritt eines Nutzers (JOIN):

Zur Beschreibung des Ablaufs bei der Teilnehmeroperation JOIN wird der Beitritt des Nutzers u_{U+1} zu der bereits bestehenden Gruppe $\{u_1,\ldots,u_U\}$ aus U Teilnehmern betrachtet. Zur Verwaltung der Gruppe wird der Schlüsselbaum KT mit Grad d und Höhe h eingesetzt.

(1) Zur Gruppenmeldung wird ein 3-Wege-Anmeldevorgang zwischen dem GC und dem Nutzer u_{U+1} durchgeführt, d.h. die Nachrichten `p1JoinRequest`, `p1JoinDistribute` und `p1JoinConfirm` ausgetauscht. Hierbei wird der Schlüssels $k_{\ell_{U+1},p_{U+1}}$ vereinbart. Anschließend wird der Schlüsselwechsel beim Nutzer u_{U+1} durch die Nachricht `p2Distribute` initialisiert.

(2a) Der GC fügt den neuen Nutzer als Knoten $v_{\ell_{U+1},p_{U+1}}$ in den Schlüsselbaum ein.

(2b) Der aktualisierte Schlüsselbaum $K\tilde{T}$ wird durch den GC mit den zufällig generierten Schlüsseln $\tilde{k}_{\ell,p}$ für die Knoten $\{w_{\ell_k,p_k} \mid w_{\ell_k,p_k} \in K\tilde{T} \wedge w_{\ell_k,p_k} \in path(v_{\ell_{U+1},p_{U+1}})\}$ erstellt.

(2c) Der GC erzeugt die Schlüsselwechselinformationen für die Nutzer (Operation `GetUpdateKeys`):
for each $w_{\ell_k,p_k} \in path(v_{\ell_{U+1},p_{U+1}})$ do

$E(\tilde{k}_{\ell_k,p_k}, k_{\ell_k,p_k})$, $E(\tilde{k}_{\ell_k,p_k}, k_{\ell_{U+1},p_{U+1}})$

(2d) Die Informationen werden an die bisherigen Nutzer $\{u_1,\ldots,u_U\}$ (Fall I) sowie den neuen Nutzer u_{U+1} (Fall II) mit einer Multicast-Nachricht `p3UpdateDistribute` verteilt.

(3, Fall I) Die Nutzer $\{u_1,\ldots,u_U\}$ ermitteln aus den empfangenen Schlüsselwechselinformationen die aktuelle Schlüsselbaumposition $\tilde{v}_{\ell_r,p_r}$ mit r=1,…,U ermittelt und berechnen den Gruppenschlüssel (Operation `SetUpdateKeys`):
for each $w_{\ell_k,p_k} \in path(\tilde{v}_{\ell_r,p_r})$ do

$\tilde{k}_{\ell_k,p_k} = D(\tilde{k}_{\ell_k,p_k}, k_{\ell_k,p_k})$

(3, Fall II) Der Nutzer u_{U+1} ermittelt aus den empfangenen Schlüsselwechselinformationen die aktuelle Schlüsselbaumposition $v_{\ell_{U+1},p_{U+1}}$ und berechnet den Gruppenschlüssel (Operation `SetUpdateKeys`):
for each $w_{\ell_k,p_k} \in path(v_{\ell_{U+1},p_{U+1}})$ do

$\tilde{k}_{\ell_k,p_k} = D(\tilde{k}_{\ell_k,p_k}, k_{\ell_{U+1},p_{U+1}})$

In Abbildung 52 ist ein Beispiel dargestellt, bei dem der Nutzer u_9 einer Gruppe mit acht Nutzern beitritt. Die Abbildung enthält Protokoll, Schlüsselbaum und Inhalt der Nachricht, die den Schlüsselwechsel übermittelt. In der Darstellung des Protokolls werden mit u(i) und u(k) Nutzer bezeichnet. Im Rahmen des 3-Wege-Anmeldevorgangs vereinbaren der Nutzer u_9 und der GC den Individualschlüssel $k_{2,8}$. Anschließend wird der neue Nutzer in den Baum einfügt und die Schlüssel des Pfads vom Einfügepunkt erneuert. Hierzu generiert der GC die zufälligen Schlüssel $\tilde{k}_{1,2}$ und $\tilde{k}_{0,0}$. Den Nutzern wird der Schlüsselwechsel mit der Nachricht `p3UpdateDistribute` mitgeteilt. Diese enthält die verschlüsselten Schlüssel $E(\tilde{k}_{0,0}, k_{0,0})$, $E(\tilde{k}_{1,2}, k_{1,2})$, $E(\tilde{k}_{0,0}, k_{2,8})$, $E(\tilde{k}_{1,2}, k_{2,8})$. Nach dem Empfang dieser Nachricht können nur die in der

Gruppe befindlichen Nutzer den neuen Gruppenschlüssel $\tilde{k}_{0,0}$ berechnen. Der Nutzer u_1 berechnet zum Beispiel den Gruppenschlüssel mit $\tilde{k}_{0,0}=D(E(\tilde{k}_{0,0},k_{0,0}),k_{0,0})$.

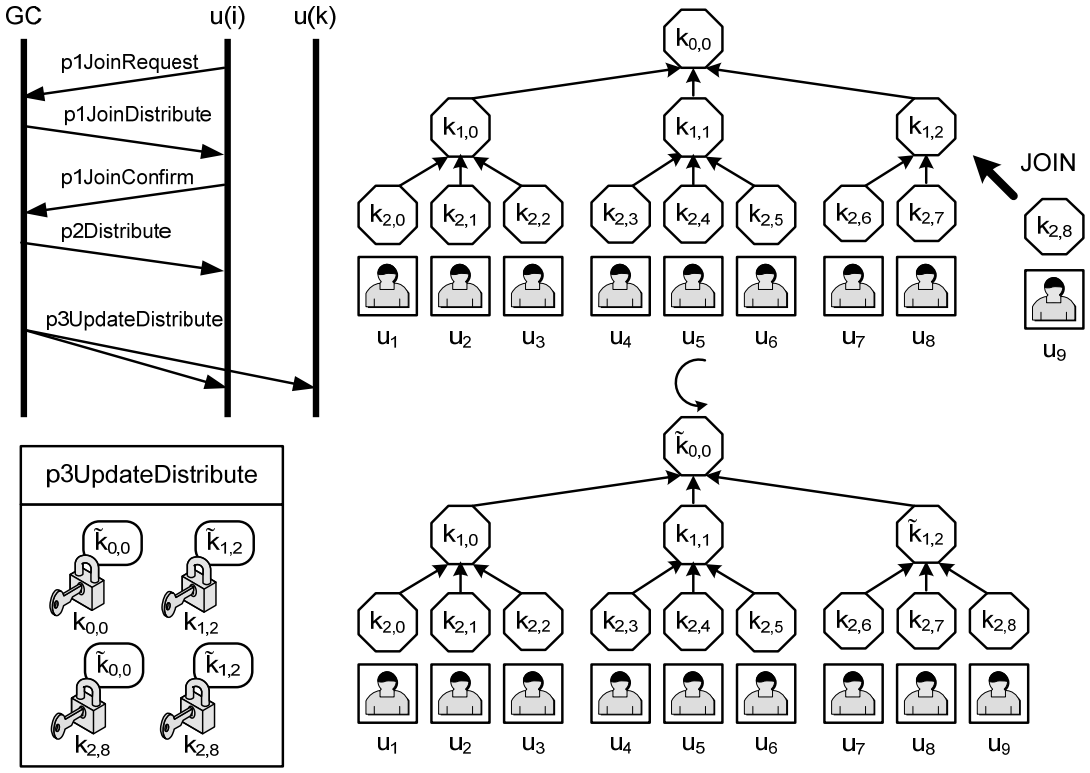

Abbildung 52: Beitritt des Nutzers u_9 beim Betriebsmodus Key Distribution

Austritt/Ausschluss eines Nutzers (LEAVE/EJECT):

Zur Beschreibung des Ablaufs bei der Teilnehmeroperation LEAVE wird der Austritt des Nutzers u_i mit $i \in \{1,\ldots,U\}$ aus der bereits bestehenden Gruppe $\{u_1,\ldots,u_U\}$ mit U Teilnehmern betrachtet. Zur Verwaltung der Gruppe wird der Schlüsselbaum KT mit Grad d und Höhe h eingesetzt.

(1) Zur Gruppenabmeldung wird ein 2-Wege-Abmeldevorgang zwischen dem GC und dem Nutzer u_i. durchgeführt, d.h. die Nachrichten `p1LeaveRequest` und `p1LeaveConfirm` ausgetauscht.

(2a) Der GC löscht den dem Nutzer zugeordneten Knotens v_{ℓ_i,p_i} im Schlüsselbaum.

(2b) Der aktualisierte Schlüsselbaum $K\tilde{T}$ wird durch den GC mit den zufällig generierten Schlüsseln $\tilde{k}_{\ell,p}$ für die Knoten $\{w_{\ell_k,p_k} \mid w_{\ell_k,p_k} \in K\tilde{T} \wedge w_{\ell_k,p_k} \in path(v_{\ell_i,p_i})\}$ erstellt.

(2c) Der GC erzeugt die Schlüsselwechselinformationen für die Nutzer (Operation `GetUpdateKeys`):

for each $w_{\ell_k,p_k} \in path(v_{\ell_i,p_i})$ do

  for each $s_{\ell_j,p_j} \in child(w_{\ell_k,p_k})$

    $E(\tilde{k}_{\ell_k,p_k},k_{\ell_j,p_j})$

(2d) Die Informationen werden an die Nutzer $\{u_1,\ldots,u_{i-1},u_{i+1},..,u_U\}$ mit der Multicast-Nachricht `p3UpdateDistribute` verteilt.

(3) Die Nutzer $\{u_1,\ldots,u_{i-1},u_{i+1},..,u_U\}$ ermitteln aus den empfangenen Schlüsselwechselinformationen die aktuelle Schlüsselbaumposition $\tilde{v}_{\ell_r,p_r}$ mit i=1,…,i-1,i+1,…,U und berechnen den Gruppenschlüssel (Operation `SetUpdateKeys`): for each $w_{\ell_k,p_k} \in path(\tilde{v}_{\ell_r,p_r})$ do

$$\tilde{k}_{\ell_k,p_k}=D(\tilde{k}_{\ell_k,p_k},k_{\ell_{k-1},p_{k-1}})$$

In Abbildung 53 ist dargestellt, wie Nutzer u_9 eine Gruppe mit neun Teilnehmern verlässt. Die Abbildung enthält Protokoll, Schlüsselbaum und Inhalt der Nachricht, die den Schlüsselwechsel übermittelt. In der Darstellung des Protokolls werden mit u(i) und u(k) Nutzer bezeichnet. Nach erfolgreichem Abmeldevorgang wird der Nutzer u_9 in dem Baum gelöscht und die Schlüssel des Pfades vom Einfügepunkt erneuert. Hierzu generiert der GC die zufälligen Schlüssel $\tilde{k}_{1,2}$ und $\tilde{k}_{0,0}$. Den Nutzern wird der Schlüsselwechsel mit der Nachricht `p3UpdateDistribute` mitgeteilt. Diese enthält die verschlüsselten Schlüssel $E(\tilde{k}_{0,0},k_{1,0})$, $E(\tilde{k}_{0,0},k_{1,1})$, $E(\tilde{k}_{0,0},\tilde{k}_{1,2})$, $E(\tilde{k}_{1,2},k_{2,6})$, $E(\tilde{k}_{1,2},k_{2,7})$. Nach dem Empfang dieser Nachricht können nur die in der Gruppe befindlichen Nutzer den neuen Gruppenschlüssel $\tilde{k}_{0,0}$ berechnen. Der Nutzer u_1 berechnet den Gruppenschlüssel mit $\tilde{k}_{0,0}=D(E(\tilde{k}_{0,0},k_{1,0}),k_{1,0})$.

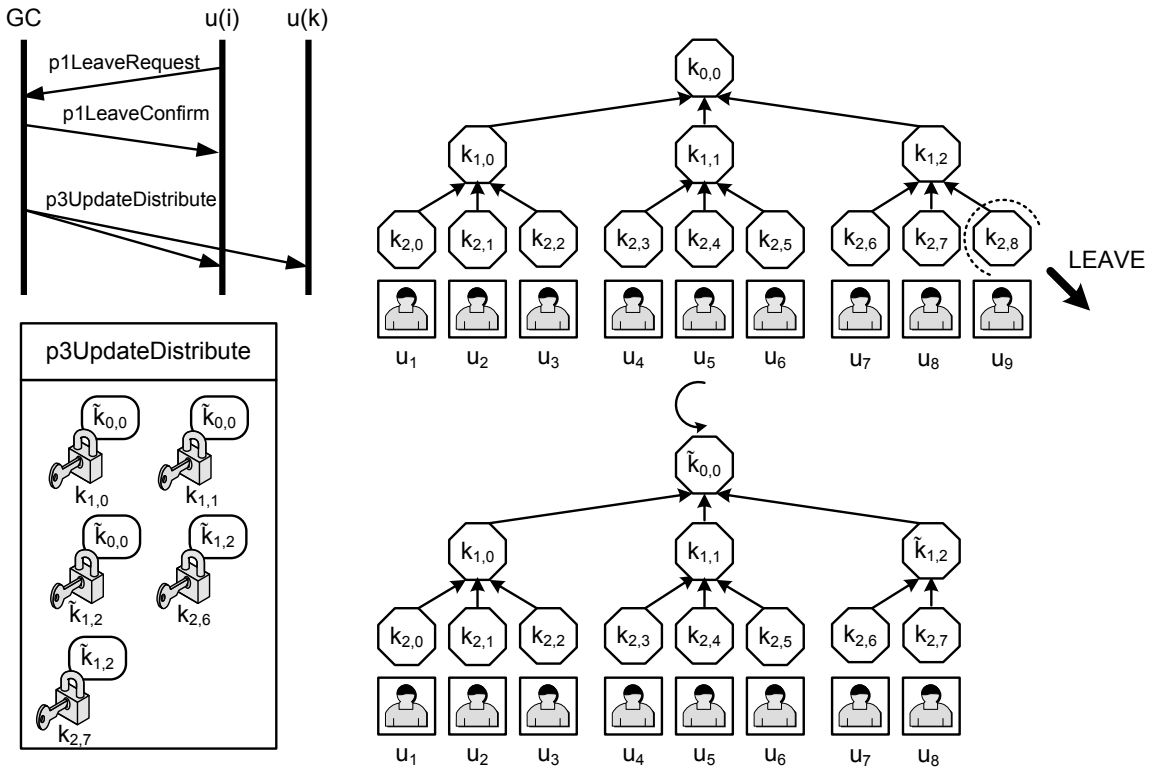

Abbildung 53: Austritt des Nutzers u_9 beim Betriebsmodus Key Distribution

Bei der Operation JOIN wählt der GC den Einfügepunkt für neue Nutzer mit dem Ziel, den Schlüsselbaum zu balancieren. Die Auswirkung der Auswahl des Einfügepunktes wird in Abschnitt 7.4 untersucht. Als Aufwandsabschätzung des Betriebsmodus Key Distribution wird in Tabelle 17 die Komplexität des Verfahrens in Abhängigkeit von Einfüge-/Löschposition $v_{\ell,p}$ mit der Pfadlänge $k=\|path(v_{\ell,p})\|$ beim Schlüsselwechsel angegeben. Hierbei werden die Anzahl der ausgetauschten Nachrichten, die Anzahl der kryptographischen Operationen und die Anzahl der übertragenen Hilfsschlüssel betrachtet. Im

Betriebsmodus Key Distribution sind die kryptographischen Operationen Verschlüsselungen von Hilfsschlüsseln. Bei der Komplexitätsanalyse wird angenommen, dass nach der Teilnehmeroperation ein vollständiger Schlüsselbaum besteht. Für einen derartigen Baum mit einer Nutzeranzahl U und dem Grad d gilt $k=Log_dU$.

Verfahren	Teilnehmeroperation	Anzahl der Nachrichten	Anzahl der Verschlüsselungen	Anzahl der Hilfsschlüssel
MIKE, Betriebsmodus Key Distribution	JOIN	1	2·k	2·k
	LEAVE/EJECT	1	d·k	d·k

Tabelle 17: Komplexität des Betriebsmodus Key Distribution

4.4.3 Betriebsmoduswechsel

Falls der Betriebsmodus Key Agreement nicht mehr effizient genug ist, kann auf den Modus Key Distribution umgeschaltet werden, ohne dass die Gruppe neu initialisiert werden muss. Hierzu löscht der Nutzer, der die Aufgaben des GC übernimmt, den ihm zugeordneten Knoten (vgl. Abbildung 54). Dieses ist notwendig, da im Modus Key Distribution der GC kein Bestandteil des Baumes ist. Weiterhin berechnet der GC unter Verwendung des DH-Algorithmus einen neuen Baum. Der im Modus Key Distribution verwendete Baum besitzt den Knotengrad d=2. Die Nutzer müssen in diesem Betriebsmodus nur den Pfad zur Wurzel kennen. Den berechnen sie unter Verwendung des DH-Algorithmus und des Blindschlüssels des GC. Ein Umschalten in die umgekehrte Richtung ist nur sinnvoll, wenn im Modus Key Distribution ein Schlüsselbaum vom Grad d=2 verwendet wird. In diesem Fall muss der GC einen Knoten, der ihm selbst zugeordnet wird, dem Schlüsselbaum hinzufügen (vgl. Abbildung 54). Beim Umschalten des Betriebsmodus Key Distribution in den Betriebsmodus Key Agrement übernimmt der GC die Aufgabe des ersten TM. Da im Modus Key Distribution den Nutzern die Baumstruktur nicht bekannt ist, müssen sie darüber informiert werden. Dies entspricht im Prinzip einer neuen Initialisierung der Gruppe.

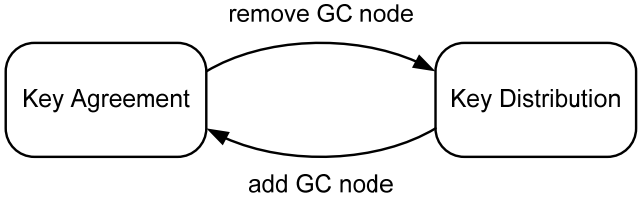

Abbildung 54: Betriebsmoduswechsel beim Schlüsselmanagements MIKE

Der nachfolgende ist die Funktionsweise des Betriebsmoduswechsels von Key Agreement nach Key Distribution etwas formaler beschrieben. Es wird angenommen, dass der Nutzer $u_{i_{GC}}$ GC wird und dessen Schlüssel $k_{\ell_{GC},p_{GC}}$, $bk_{\ell_{GC},p_{GC}}$ in dem Knoten $v_{\ell_{GC},p_{GC}}$ gespeichert sind:

(1) Der Nutzer $u_{i_{GC}}$, löscht im Schlüsselbaum den Knoten $v_{\ell_{GC},p_{GC}}$.

(2a) Der Nutzer $u_{i_{GC}}$ ermittelt anschließend den im Betriebsmodus Key Distribution verwendeten Schlüsselbaum mit:

```
for ℓ=0 to h do
   for p=0 to p_max(ℓ) do
      if ℓ = = ℓ_GC & p = = p_GC then do
```

delete $v_{\ell_{GC},p_{GC}}$

else do

$\tilde{k}_{\ell,p}=DH(bk_{\ell,p,},k_{\ell_{GC},p_{GC}})$

(2b) Die übrigen Nutzer u_j mit $j \neq i_{GC}$ ermitteln den im Betriebsmodus Key Distribution verwendeten Schlüsselbaum mit:

for each $w_{\ell_k,p_k} \in path(v_{\ell_j,p_j})$ do

$\tilde{k}_{\ell_k,p_k}=DH(bk_{\ell_{GC},p_{GC}},k_{\ell_k,p_k})$

In Abbildung 55 ist ein Beispiel für die Berechnung des Schlüsselbaums durch den GC in einer Gruppe mit vier Nutzern bei einem Betriebsmoduswechsel dargestellt. Der Wechsel wird derart durchgeführt, dass der Nutzer u_1 im Modus Key Distribution GC wird. Da bei dem Verfahren Key Distribution der GC nicht Bestandteil des Baumes ist, löscht der Nutzer u_1 seinen Eintrag im Baum. Anschließend berechnet er die Schlüssel des Baums unter Verwendung des DH-Algorithmus und seines Schlüssels $k_{2,0}=k_{\ell_{GC},p_{GC}}$. Die im Betriebsmodus Key Distribution verwendeten Schlüssel für eine Gruppe mit vier Teilnehmern sind $\tilde{k}_{0,0}=DH(bk_{0,0},k_{2,0})$, $\tilde{k}_{1,0}=DH(bk_{1,0},k_{2,0})$, $\tilde{k}_{1,1}=DH(bk_{1,1},k_{2,0})$, $\tilde{k}_{2,2}=DH(bk_{2,2},k_{2,0})$ und $\tilde{k}_{2,3}=DH(bk_{2,3},k_{2,0})$

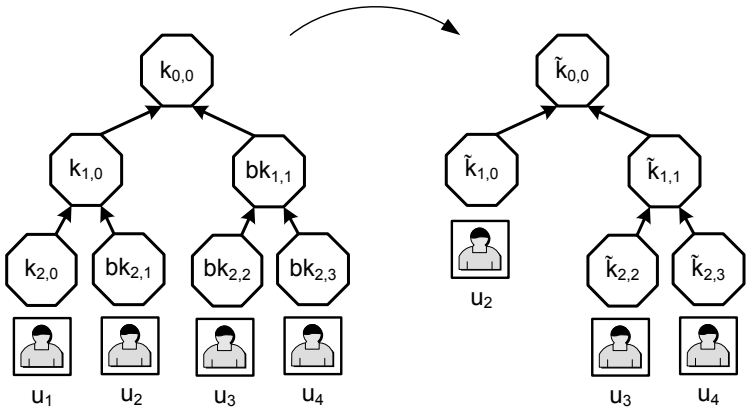

Abbildung 55: Schlüsselbaumumwandlung des Nutzers u_1 beim Betriebsmoduswechsel

Zum Umschalten Betriebsmodus Key Distribution in den Betriebsmodus Key Agreement sind die folgenden Arbeitsschritte notwendig:

(1) Der GC bzw. der erste TM fügt ein ihm zugeordnetes Blatt $v_{\ell_{GC},p_{GC}}$ in den Baum ein.

(2) Anschließend verteilt er die Blindschlüssel aller Nutzer des Baums mit der Multicast-Nachricht `p3TMDistribute`. Um wieder einen Gruppenschlüssel bereitzustellen, wird anschließend der in Abschnitt 7.3.1 für die Sammelverarbeitung von Nutzeranfragen entworfene Algorithmus angewandt.

4.5 Modul GroupPolicyDatabase

Zur Schlüsselverwaltung wird eine Sicherheitsvorschrift benötigt. Eine solche Gruppensicherheitsvorschrift ist definiert als ein Satz von Regeln zur Festlegung des Verhaltens bei sicherheitsrelevanten Aufgaben und der Berechtigung zum Gruppenbeitritt [Har01]. Das Modul `GroupPolicyDatabase` sorgt für die Einhaltung der festgelegten

Gruppensicherheitsvorschrift. Diese basiert auf dem Prinzip der rollen-basierten Sicherheit [Fer92]. Das bedeutet, dass Verantwortlichkeiten für sicherheitsrelevante Aufgaben unterschiedlichen Rollen zugeordnet werden. Anschließend werden diese abstrakten Rollen dann den Teilnehmern zugewiesen. Nachfolgende drei Rollen sind definiert:

- Gruppenbesitzer (Group Owner, GO)
  Die Rolle GO spezifiziert die Gruppensicherheitsvorschrift. Er wird deshalb auch als Herausgeber der Sicherheitsvorschrift bezeichnet.
- Gruppenverwalter (Group Manager, GM)
  Je nach Betriebsart wird die Rolle GM vom GC bzw. TM übernommen. Aufgaben des Gruppenverwalters sind die Schlüsselbereitstellung und die Durchsetzung der Zugriffskontrolle. Der GC/TM gewährleistet die Zugriffskontrolle, indem er das Schlüsselmaterial nur an autorisierte Nutzer verteilt und einen Schlüsselwechsel initiiert, wenn sich Gruppengröße und Zusammensetzung ändern. Aus der Sicht der Gruppensicherheitsvorschrift wird die Rolle GM im Modus Key Distribution von einem Nutzer, der selbst nicht in der Gruppe ist, durchgeführt. Im Gegensatz dazu ist im Modus Key Agreement der TM ein Nutzer, der die Rollen GM und die im nächsten Absatz beschriebene Rolle M übernimmt. Die Rolle GM wird in diesem Modus dynamisch zugewiesen.
- Mitglied (Member, M)
  Den Nutzern wird die Rolle M zugewiesen. Sie sind die Anwender des Gruppenschlüssels. Die Nutzer akzeptieren den Empfang von Schlüsselmaterial nur von einem Teilnehmer, dem die Rolle GM zugewiesen wurde.

Die Gruppensicherheitsvorschrift wird vor Beginn der Gruppenkommunikation in einem administrativen Vorgang von der Rolle GO erstellt und im Datenspeicher des Moduls `GroupPolicyDatabase` von allen Beteiligten abgelegt. Die Erstellung der Vorschrift durch die Rolle GO wird in [Pra99] vorgeschlagen. Auf Grund der komplexen Struktur der Gruppenkommunikation ist die Gruppensicherheitsvorschrift ebenfalls sehr komplex. Die Einteilung der Sicherheitsvorschrift in fünf Kategorien dient zur Strukturierung der Vorschrift und erleichtert eine verständliche Darstellung [Har01]. Nachfolgend wird ein Überblick über die Kategorien und deren Inhalt gegeben:

- Kategorie 1: Kennzeichnung (Policy identification)
  Die Gruppensicherheitsvorschrift muss eine eindeutige Kennzeichnung besitzen. Hierzu wird die Multicast-Adresse des zu schützenden Nutzdatenverkehrs verwendet.
- Kategorie 2: Autorisierung von sicherheitsrelevanten Aufgaben (Authorisation for group actions)
  Die Zuweisung der Berechtigung für die Durchführung sicherheitsrelevanter Aufgaben im Schlüsselmanagement erfolgt hauptsächlich implizit. Einen Überblick über die in einem Schlüsselmanagement vorhandenen sicherheitsrelevanten Aufgaben und deren Zuordnung zu den oben definierten Rollen vermittelt Tabelle 18. Im Modus Key Distribution wird explizit einem Nutzer die Rolle GM zugewiesen. Im Modus Key Agreement erfolgt dieses dynamisch während des Betriebs.

- Kategorie 3: Zugriffskontrolle (Access control)
  Alle zur Teilnahme an der Gruppenkommunikation berechtigten Nutzer besitzen einen Eintrag in der Gruppensicherheitsvorschrift. Ein solcher Eintrag enthält unter anderem einen Verweis auf ein Zertifikat, das die Gültigkeit seines Signaturschlüssels bescheinigt. Dieser wird verwendet, um Managementnachrichten zu authentisieren. Die hierzu verwendeten, asymmetrischen Schlüssel werden mit Mechanismen einer Public Key Infrastructure, d.h. mittels eines Zertifikats einem Nutzer zugeordnet. Die benötigten Zertifikate und Authentisierungsschlüssel sind vor Beginn des Betriebes zu erstellen.
- Kategorie 4: Mechanismen für die Gruppensicherheitsdienste (Mechanisms for group security services)
  Diese Kategorie enthält einen Teilbereich, in dem die Gruppenverwaltungsmethoden festgelegt werden. In einem zweiten Teilbereich werden die Mechanismen zum Schutz der Nutzdatenübertragung festgesetzt. Der Teilbereich Gruppenverwaltungsmethode spezifiziert Parameter zur Gruppenschlüsselverwaltung, z.B. den Grad des eingesetzten Schlüsselbaums. Im Rahmen der Festlegung der Nutzdatenschutzmechanismen wird unter anderem das eingesetzte Sicherheitsprotokoll, z.B. IPSec ESP, festgelegt.
- Kategorie 5: Verifikation der Gruppensicherheitsvorschrift (Group policy verification)
  Um die Integrität und Authentizität der Vorschrift zu gewährleisten, wird diese vom GO mit einer Signatur versehen.

Sicherheitsrelevante Aufgabe	Beschreibung	Rolle M	Rolle GM	Rolle GO
Sicherheitsvorschrifterzeugung	Erzeugung und Verteilung der Gruppensicherheitsvorschrift	-	-	X
Sicherheitsvorschriftsänderung	Veränderung der Gruppensicherheitsvorschrift	-	-	X
Zuweisungen von Berechtigungen	Berechtigungszuweisung für sicherheitsrelevante Aufgaben	-	-	X
Schlüsselerzeugung	Erzeugung von Schlüsselmaterial	-	X	-
Gruppenauflösung	Beendigung einer sicheren Gruppenkommunikation	-	X	-
Schlüsselbereitstellung	Bereitstellung von Schlüsselmaterial	-	X	-
Schlüsselwechselinitiierung	Initiierung eines Schlüsselwechsels	-	X	-
Aufgabenzuweisung	Zuweisung von Aufgaben an Teilnehmer	-	X	-
Teilnehmeraufnahme	Aufnahme eines Nutzers in die Gruppe	-	X	-
Teilnehmerausschluss	Ausschluss eines Nutzers aus der Gruppe	-	X	-
Gruppenüberwachung	Überwachung der Gruppenanmeldung	-	X	-
Gruppenschlüsselzugriff	Zugriff auf den Gruppenschlüssel	X	X	-

Tabelle 18: Zuweisung der sicherheitsrelevanten Aufgaben zu den definierten Rollen

Eine wichtige Aufgabe des Moduls `GroupPolicyDatabase` ist die Realisierung der Zugriffskontrolle. Zu deren Gewährleistung werden nur Anfragen zum Gruppenbeitritt bearbeitet, die von berechtigten Nutzern gestellt werden. Hierzu wird überprüft, ob die Anfrage zur Gruppenteilnahme von einem berechtigten Absender signiert ist. Nach der Überprüfung des Eintrags in der Gruppensicherheitsvorschrift und der Nachrichtensignatur wird mit den nachfolgenden drei Schritten die Gültigkeit des Signaturschlüssels verifiziert:

- Der Gültigkeitszeitraum des Zertifikats, das die Gültigkeit des Signaturschlüssels bescheinigt, wird überprüft. Zusätzlich wird verifiziert, ob das Zertifikat von einer bekannten und akzeptierten Zertifizierungsinstanz ausgestellt, d.h. signiert, wurde.
- Die Gültigkeit des Zertifikats der Zertifizierungsinstanz, welche das Nutzerzertifikat herausgegeben hat, wird überprüft.
- Um sicherzustellen, dass das Zertifikat nicht widerrufen wurde, wird geprüft, ob das Zertifikat in der letzten von der Zertifizierungsinstanz veröffentlichten Zertifikatswiderrufsliste enthalten ist.

Mit den beschriebenen Verfahren wird auch die Berechtigung des TM/GC zum Versand der Nachrichten des Schlüsselwechsels verifiziert.

Ein vollständiges Schlüsselmanagement verfügt auch über die Möglichkeit, einen Teilnehmer auszuschließen. Dieser Ausschluss erfolgt durch einen vom GC bzw. TM initiierten Schlüsselwechsel. Die Aufforderung zur Teilnehmeroperation EJECT erhält dieser über ein Protokoll, das nicht Bestandteil des Schlüsselmanagements ist. Die Berechtigung zur Gruppenmitgliedschaft wird einem ausgeschlossenen Nutzer durch Widerruf seines Zertifikats, d.h. durch Widerruf der Gültigkeit seines Signaturschlüssels, entzogen. Hierzu wird die Zertifikatswiderrufliste aktualisiert und an den GC bzw. im Modus Key Agreement an alle Nutzer verteilt. Für den Fall das im Modus Key Agreement der TM ausgeschlossen werden soll, muss von einem Nutzer der in Abschnitt 4.6.1 beschriebene Mechanismus Neuwahl des TM initiiert werden.

4.6 Verhalten in Fehlersituationen

Bei Verwendung eines Schlüsselmanagements in einem kritischen Einsatzbereich muss eine hohe Verlässlichkeit gewährleistet sein. Die Entwicklung eines verlässlichen, verteilten Schlüsselmanagementsystems stellt eine Forschungsherausforderung dar. Diese ist insbesondere dann erheblich, wenn kein Gruppenkommunikationssystem verwendet wird, das den Gruppenmitgliedschaftsdienst und den zuverlässigen, geordneten Multicast-Dienst bereitstellt. Die Verlässlichkeit eines Systems kann durch Integration von Mechanismen zur Fehlertoleranz verbessert werden. Bei deren Entwicklung wird das Schlüsselmanagement als partiell synchrones System modelliert. Hierbei besteht das System aus MIKE-Prozessen $\{p_1,\ldots,p_U\}$, deren einzige Kommunikationsmöglichkeit im Austausch von Nachrichten besteht. Als eindeutige Bezeichnung dieses Prozesses kann die IP-Adresse verwendet werden. Partielle Synchronisation bedeutet, es existiert eine Schranke für die Nachrichtenübertragungszeit, die aber nicht bekannt ist bzw. die Schranke für die Nachrichtenübertragungszeiten ist bekannt, aber sie gilt erst nach einer unbekannten endlichen Zeit [Gär01]. Weiterhin besitzt jeder MIKE-Prozess eine lokale nicht synchronisierte Uhr. Für die Entwicklung der Mechanismen zur Fehlertoleranz wird weiterhin angenommen, dass das System Prozessfehlern der Fehlerklasse Zusammenbruchsfehler (Crash) unterliegt. Bei Zusammenbruchsfehlern arbeitet ein Prozess von MIKE entweder korrekt oder hält vollständig an [Gär01]. Übertragungsfehler des Kommunikationsnetwerks werden behandelt wie Zusammenbruchsfehler eines Mike-Prozesses, da es nicht möglich ist,

zu unterscheiden, ob ein Prozess nicht in der Lage ist zu kommunizieren oder ob er zusammengebrochen ist.

Um einen Einsatz von MIKE in Kommunikationsnetzwerken mit funkbasierten Verbindungen zu ermöglichen, werden zusätzlich Mechanismen, die die Fehlertoleranz eines MIKE-Prozesses bei temporären Kommunikationsstörungen verbessern, integriert. Bei deren Entwurf wird angenommen, dass das zur Verfügung stehende Kommunikationsnetzwerk Fehlern der Fehlerklasse Zusammenbruchsfehler-Wiederherstellung (Crash-Recovery) unterliegt.

4.6.1 Verhalten bei Zusammenbruchsfehlern

Das Verhalten des Systems bei Zusammenbruchsfehlern wird für jeden Betriebsmodus getrennt beschrieben. Diese Beschreibung wird mit der Darstellung des Verhaltens im Modus Key Agreement begonnen. Bei Entdeckung eines Zusammenbruchsfehlers eines Nutzers wird durch Rekonfiguration die fehlerhafte Komponente ausgeschlossen. Die von einem Nutzer im Modus Key Agreement durchgeführte Gruppenanmeldung besitzt nur eine begrenzte Gültigkeitsdauer $t_{UserTimeout}$. Zur Fehlererkennung wird diese Zeitschranke überwacht. Erfolgt innerhalb dieses Zeitraums keine erneute Gruppenanmeldung, wird vom TM angenommen, dass bei dem Nutzer ein Zusammenbruchsfehler vorliegt und dieser wird aus der Gruppe entfernt. Mit dem Mechanismus wird verhindert, dass fehlerhafte MIKE-Prozesse Teilnehmer der Gruppe sind. Wird ein Nutzer auf Grund der Annahme eines Zusammenbruchsfehlers aus der Gruppe entfernt, kann dieser sich im Gegensatz zum Ausschluss durch die Teilnehmeroperation EJECT erneut anmelden.

Darüber hinaus kann durch einen Verbindungsverlust (Zusammenbruch des Kommunikationsnetzwerks des TM) oder durch einen Zusammenbruchsfehler des Mike-Prozesses ein Verlust des TM auftreten (Abbildung 56). Die Nutzer des Schlüsselmanagements können einen Verlust des TM anhand der nachfolgenden Kriterien vermuten:

- Kein Empfang der Nachricht `p3UpdateDistribute` nach dem erfolgreichen Empfang der Nachricht `p3TMDistribute`.
- Kein Empfang der Nachricht `p3UpdateDistribute`, obwohl die Gültigkeit des Gruppenschlüssels $t_{KeyTimeout}$ abgelaufen ist.

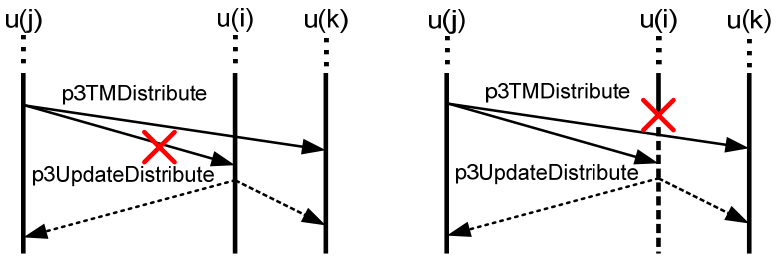

Abbildung 56: Verlust des TM durch Zusammenbruchsfehler (rechts) und Verbindungsverlust (links)

Da diese Kriterien nicht eindeutig den Verlust des TM identifizieren, nimmt der Nutzer zunächst an, dass ein TM existiert und er versucht eine erneute Gruppenanmeldung. Ist diese

erfolglos, wird der Mechanismus zur Neuwahl des TM eingesetzt. Nutzer, die sich ebenfalls bei einer erneuten Gruppenanmeldung befinden, brechen diese ab und führen den Mechanismus Neuwahl eines TM durch (Abbildung 57). Das Ziel des Mechanismus besteht darin, einen Konsens über den Teilnehmerzustand zu erhalten und einen neuen TM festzulegen. Hierzu wurde das SecureRing-Membership-Protokoll für die Verwendung im Schlüsselmanagement angepasst [Kih98]. Das Protokoll wurde ausgewählt, da es auch in Netzwerken mit Übertragungsfehlern terminiert, d.h. Konsens über den Teilnehmerzustand herstellt. Bei diesem Mechanismus verwaltet jeder Prozess vier Statuslisten:

- Die Liste `my_memb` enthält die Nutzer, die vor dem Verlust des TM Teilnehmer der Gruppe waren.
- Die Liste `my_proc_set` enthält die Nutzer, die als funktionierend eingestuft und Teilnehmer in dem wiederhergestellten Schlüsselmanagement sind.
- Die Liste `my_fault_set` enthält Nutzer, die als fehlerhaft eingestuft und nicht Teilnehmer des wiederhergestellten Schlüsselmanagements sind.
- Die Liste `my_agreement` zeigt an, mit welchen Nutzern Übereinstimmung besteht.

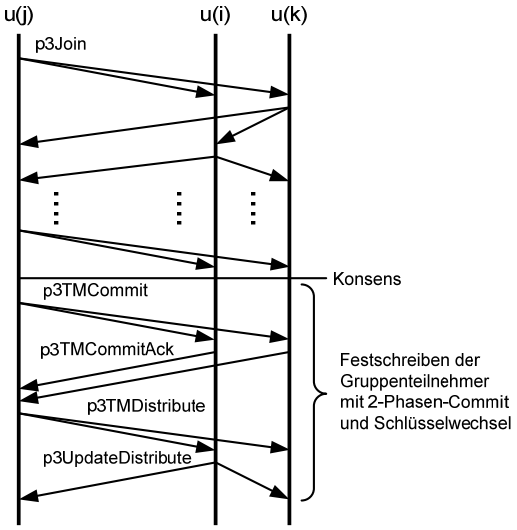

Abbildung 57: Ablauf des Mechanismus Neuwahl des TM

Um einen Konsens zu erzielen, werden von dem Mechanismus die nachfolgenden zwei Nachrichten verwendet:

- Nachricht `p3Join`
  Bestandteil der Nachricht ist das Feld `proc_set`, welches Nutzer enthält, die vom Sender der Nachricht als funktionierend eingestuft werden. Weiterhin werden mit dem Feld `fault_set` die als fehlerhaft eingestuften Teilnehmer übermittelt. Eine Sequenznummer dient als Ordnungskriterium für die Nachricht.
- Nachricht `p3TMCommit`
  Diese Nachricht dient als Bestätigung über die Teilnehmer im wiederhergestellten Schlüsselmanagement. Das Feld `memb_list` enthält hierzu die Nutzer, die am wiederhergestellten Schlüsselmanagement teilnehmen. Eine Sequenznummer dient als Ordnungskriterium für die Nachrichten.

- Nachricht `p3TMCommitAck`
  Mittels dieser Nachricht wird die Nachricht `p3TMCommit` quittiert.

Nach erfolglosem Versuch einer erneuten Gruppenanmeldung wird in den Zustand `AgreeStateGather` gewechselt. Hierbei wird die Nachricht `p3Join` mit dem Inhalt der Listen `my_proc_set` und `my_fault_set` versandt. Erhält ein Teilnehmer die Nachricht `p3Join`, vergleicht er nach Überprüfung der Sequenznummer die Felder `proc_set` und `fault_set` mit den Listen `my_proc_set` und `my_fault_set`. Sind diese identisch, wird die Übereinstimmung in der Liste `my_agreement` vermerkt. Ist dieses nicht der Fall und ist der Absender der Nachricht nicht in der Liste `my_fault_set` enthalten, werden die noch nicht in den Listen `my_proc_set` und `my_fault_set` enthaltenen Nutzer dort jeweils eingetragen. Ein Empfänger der Nachricht `p3Join` versendet, nachdem er seine Statuslisten aktualisiert hat, seinerseits die Nachricht `p3Join`. Ein Nutzer erreicht Konsens über den Teilnehmerstatus, wenn alle Nutzer der Liste `my_proc_set` auch in der Liste `my_agreement` vermerkt sind. Erreicht der Nutzer mit der kleinsten IP-Adresse Konsens über den Teilnehmerstatus, wechselt er in den Zustand `AgreeStateCommit` und generiert die Nachricht `p3TMCommit`. Diese wird jedem Nutzer, der seinerseits als funktionierend eingestuft wurde, zugesandt. Stimmen die mit der Nachricht `p3TMCommit` übermittelten Nutzer mit denen vom Empfänger als funktionierend eingestuften Teilnehmer überein, bestätigt der Teilnehmer diese Übereinstimmung. Hat der Nutzer mit der kleinsten IP-Adresse von allen Teilnehmern eine Bestätigung, d.h. die Nachricht `p3TMCommitAck` erhalten, wird dieser TM der wiederhergestellten Gruppe und versendet die Nachricht `p3TMDistribute`. Ab diesem Zeitpunkt arbeitet das Schlüsselmanagementprotokoll wie in Abschnitt 4.4.1 beschrieben. Wird die Nachricht `p3TMCommit` nicht von allen Teilnehmern bestätigt, d.h. tritt ein Fehler auf, wechselt der potentielle TM in den Zustand `AgreeStateGather` und versendet die Nachricht `p3Join`. Der oben beschriebene Mechanismus beginnt erneut. Zur Verdeutlichung des Mechanismus ist in Abbildung 58 ein vereinfachter Zustandsautomat dargestellt.

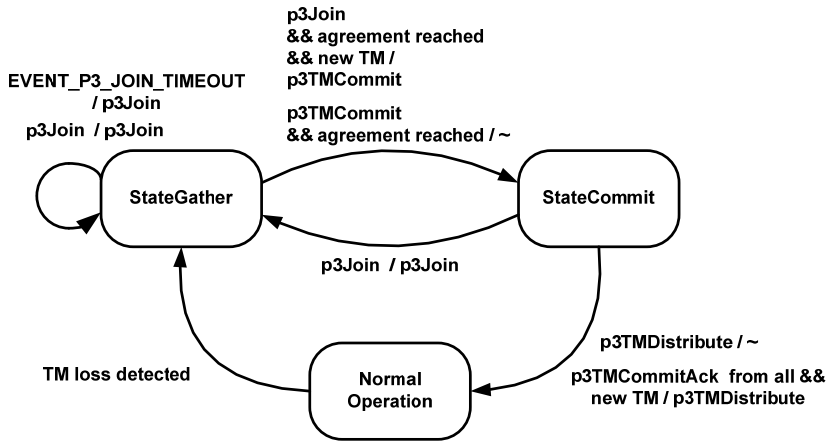

Abbildung 58: Zustandsdiagramm des Mechanismus Neuwahl des TM

Im folgenden Abschnitt wird auf die Fehlertoleranzmechanismen des Modus Key Distribution eingegangen. In diesem Modus wird ein Zusammenbruchsfehler eines Nutzers durch Rekonfiguration behoben, d.h. fehlerhafte Komponenten werden ausgetauscht. Als Mittel zur Fehlererkennung wird die Zeitschrankenüberwachung eingesetzt. Nachfolgend wird der

Mechanismus genauer erläutert. Im Modus Key Distribution hat die durchgeführte Gruppenanmeldung eines Nutzers beim GC ebenfalls nur eine begrenzte Gültigkeitsdauer $t_{UserTimeout}$. Eine erneute Gruppenanmeldung ist erforderlich, damit der GC den Nutzer nicht auf Grund der Annahme eines Zusammenbruchsfehlers aus der Gruppe entfernt. Hierdurch wird verhindert, dass fehlerhafte MIKE-Prozesse Teilnehmer der Gruppe sind. Wird ein Nutzer auf Grund der Annahme eines Zusammenbruchsfehlers aus der Gruppe entfernt, kann dieser sich im Gegensatz zum Ausschluss durch die Teilnehmeroperation EJECT erneut beim GC anmelden. Für Zusammenbruchsfehler des GC sind keine Fehlertoleranzmechanismen enthalten.

4.6.2 Verhalten bei temporären Kommunikationsfehlern

Um ein Schlüsselmanagement in Kommunikationsnetzwerken mit funkbasierten Verbindungen zu ermöglichen, müssen insbesondere temporäre Kommunikationsfehler berücksichtigt werden. Als Mittel zur Fehlertoleranz werden Zeit- und Informationsredundanz eingesetzt. Nachfolgend werden die in MIKE hierzu eingesetzten Mechanismen genauer beschrieben. Begonnen wird die Darstellung mit den Kompensationsmechanismen beim Gruppenbeitritt und Gruppenaustritt. Nachrichten des Typs `p1JoinRequest`, `p1JoinDistribute`, `p1JoinConfirm`, `p1LeaveRequest`, `p1LeaveConfirm` und `p3TMRequest` werden mittels des Mechanismus Zeitredundanz, d.h. genauer Übertragungswiederholung, gegen temporäre Kommunikationsfehler geschützt. Zu der Erkennung wird beim Versand einer Nachricht des genannten Typs eine Zeitschrankenüberwachung gestartet, die den Empfang einer Antwort auf die Nachricht überwacht. Ein Überschreiten der Zeitschranke generiert das Ereignis `EVENT_CONNECT_TIMEOUT`. Wird dieses dem aktuellen Zustand übergeben, wird eine Aufforderung zur erneuten Übertragung der zuletzt gesendeten Nachricht versandt. Dieses wird so lange wiederholt, bis eine vorgegebene maximale Anzahl der Übertragungswiederholungen erreicht ist. Der Mechanismus Zeitredundanz ist zur Kompensation von Kommunikationsstörungen beim Schlüsselwechsel, d.h. bei den Multicast-Nachrichten `p3TMDistribute` und `p3UpdateDistribute` bzw. der Nachricht `p2Distribute` nicht geeignet. Gründe hierfür sind die bei Paketverlust auftretende große Anzahl an Anfragen zur Übertragungswiederholung sowie die Anzahl der Übertragungswiederholungen selbst. Außerdem führt der Mechanismus Zeitredundanz zu großen Nachrichtenverzögerungen. Stattdessen werden diese Nachrichten mit Informationsredundanz, d.h. mit einer Vorwärtsfehlerkorrektur (Forward Error Correction, FEC), versehen. Vorwärtsfehlerkorrekturverfahren werden meistens von allgemeinen Fehlerkorrekturverfahren, die auch Bitfehler korrigieren können, abgeleitet. Im Fall IP-basierter Kommunikation werden Bitfehler durch die darunter liegenden Schichten schon behandelt, so dass nur noch Paketverluste kompensiert werden müssen. Hier kommen so genannte Block Erasure Codes zum Einsatz. Das in [Riz97] beschriebene Verfahren wird eingesetzt, weil es hierfür geeignet ist. Bei Fehlerkorrekturverfahren, die Paketverluste ausgleichen, wird ein Datenpaket in n kodiert und anschließend versendet. Der Empfang von k dieser kodierten Pakete (n>k) ist ausreichend, um die Daten wieder herzustellen. Das Verfahren lässt sich der Klasse der linearen Fehlerkorrekturcodes zuordnen. Dabei wird der

Paketvektor $\vec{x} = (x_0,...,x_{k-1})$ mit den k Datenpaketen mit einer n×k Kodierungsmatrix G multipliziert, um den zu sendenden Paketvektor $\vec{y} = (y_0,...,y_{n-1})$ der Größe $s_{Data}^{y_i}$ zu erhalten:

$$\vec{y} = G \cdot \vec{x}$$

Wenn k Pakete aus $\vec{y}$ empfangen werden, können die k Quellpakete wiederhergestellt werden. Hierzu werden die empfangenen k Datenpakete zum Paketvektor $\vec{y}'$ zusammengefasst. Es gilt für den empfangenen Paketvektor $\vec{y}'$ die Gleichung

$$\vec{y}' = G' \cdot \vec{x}$$

Hierbei bezeichnet G´ eine k×k Kodierungsmatrix. Die Daten können mit der Matrix G'^{-1} durch die Berechnung der nachfolgenden Gleichung wieder hergestellt werden:

$$\vec{x} = G'^{-1} \cdot \vec{y}'$$

Zusätzlich zu den eigentlichen kodierten Daten $\vec{y}$ müssen weitere Informationen übermittelt werden, wie z.B. die Sequenznummer des Pakets innerhalb des versendeten Paketvektors sowie die Sequenznummer des Paketvektors, um bei vertauschten Paketen und bei Paketverlusten ankommende Pakete in der richtigen Reihenfolge zu verarbeiten. Hierzu muss zu jedem Element von $\vec{y}$ ein FEC-Paketkopf hinzugefügt werden. Die Vorwärtsfehlerkorrekturverfahren können dazu eingesetzt werden, kurzzeitige Kommunikationsstörungen bzw. Paketverluste zu kompensieren. Zur Erkennung länger andauernder temporärer Übertragungsfehler enthalten Nachrichten des Typs `p3TMDistribute` und `p3UpdateDistribute` eine fortlaufende Sequenznummer. Diese Nummer wird beim Empfang mit einer bereits gespeicherten Sequenznummer verglichen. Nachrichten, deren Sequenznummer kleiner als die gespeicherte Sequenznummer ist, werden, da diese veraltet sind oder von einem Angriff durch wiederholtes Senden stammen, ignoriert. Nachrichten, die eine um eins erhöhte Sequenznummer enthalten, werden verarbeitet. Der Empfang von Nachrichten des Typs `p3TMDistribute` und `p3UpdateDistribute` mit einer um mehr als eins erhöhten Sequenznummer weist daraufhin, dass der Teilnehmer einige Schlüsselwechsel nicht empfangen hat. Wird ein derartiger Fall festgestellt, so führt ein Nutzer einen erneuten Gruppenbeitritt durch. Eine weitere Möglichkeit, einen temporären Übertragungsfehler zu erkennen, nutzt die begrenzte Gültigkeit des Gruppenschlüssels aus. Nach dem Empfang eines Gruppenschlüssels wird eine Zeitschrankenüberwachung gestartet, die dessen Gültigkeitszeitraum überwacht. Ist dieser abgelaufen und der Nutzer hat noch keinen neuen Gruppenschlüssel erhalten, schließt der Nutzer auf einen temporären Verbindungsverlust und führt einen erneuten Gruppenbeitritt durch.

4.7 Kapitelzusammenfassung

In diesem Kapitel wurde das Konzept MIKE zur Verwaltung eines dynamischen Gruppenschlüssels vorgestellt. Die Verfügbarkeit von Multicast ist die einzige Anforderung, die von der Kommunikationsinfrastruktur erfüllt werden muss, um einen Einsatz zu ermöglichen. Kernpunkte des Konzepts sind zwei Betriebsmodi zur Schlüsselbereitstellung und die Mechanismen zur Gewährleistung von Reparierbarkeit. Durch den Einsatz des Hilfsmittels Schlüsselbaum kann die zentrale Forderung nach einem effizienten

Schlüsselwechsel erfüllt werden. Synergieeffekt der Schlüsselbaumverwendung ist die einfache Realisierung beider Betriebsmodi. Eine Reparierbarkeit wird im Modus Key Agreement durch den Verzicht auf einen zentralen Prozess zur Schlüsselbereitstellung erreicht. Sollte der mit diesem Verzicht verbundene erhöhte Ressourcenbedarf die Kapazität der Kommunikationsinfrastruktur übersteigen, kann auf den zweiten Betriebsmodus umgeschaltet werden. Nachteil dieses Modus ist die Existenz eines SPoF. Beide Betriebsmodi setzen zur Kompensation temporärer Übertragungsfehler die Mechanismen Übertragungswiederholung, vorwärtsgerichtete Fehlerkorrektur, sowie Sequenznummernüberwachung in Kombination mit einem erneuten Anmeldeverfahren ein. In wissenschaftlicher Hinsicht wird mit dem vorgestellten Konzept gezeigt, dass eine getrennte Betrachtung der Schlüsselverwaltung nach zentralem und verteiltem Verfahren nicht immer sinnvoll ist und sich beide Schlüsselbereitstellungsarten kombinieren lassen.

5 Implementierung des Konzepts MIKE

Zur Entwicklung der Software für eine Realisierung des Konzepts MIKE wird die Methode des objektorientierten Designs verwendet. Diese Methode stützt sich auf die Annahme, dass ein komplexes System nicht sofort bis in alle Einzelheiten geplant werden kann und während der Realisierung noch experimentiert werden muss [Lev99]. Aus diesem Grund erscheint diese Methode für die Erstellung einer experimentellen Implementierung von MIKE am besten geeignet. Zusätzlich wird durch Verwendung modularer Programmbausteine Verständnis, Übersicht und Erweiterbarkeit wesentlich verbessert. Ein objektorientiertes Design ist unabhängig von der Programmiersprache, jedoch ist es vorteilhaft, auch eine objektorientierte Programmiersprache zu verwenden. Bei der Umsetzung wird deshalb die Programmiersprache C++ eingesetzt.

Im Folgenden wird die Softwarearchitektur des Moduls `KeyManagement` erläutert. Dieses ist der Kern des Schlüsselmanagements MIKE. Außerdem wird der verwendete Datenspeicher des Moduls `GroupPolicyDatabase` vorgestellt.

5.1 Realisierung des Moduls KeyManagement

Schwerpunkte der Erläuterung des Moduls `KeyManagement` sind die Softwarearchitektur der Zustandsautomaten der Schlüsselbereitstellungsmechanismen sowie die verwalteten Schlüsselbäume. Die Zustandsautomaten erzeugen bzw. verarbeiten mittels eines Schlüsselbaums die zur Schlüsselbereitstellung notwendigen Nachrichten und Ereignisse. Das Modul `KeyManagement` enthält eintreffende Nachrichten und Ereignisse vom Modul `MessageDispatcher`. Ein Multiplexing der zum Empfang notwendigen Schnittstellen findet auf der Basis der Unix-Funktion `select()` statt. Eingehende Nachrichten werden in dem Modul `GroupManagementFramework`, das die Schnittstelle zwischen den Modulen `KeyManagement` und `MessageDispatcher` darstellt, in die Datenstruktur `MngtMessage` umgewandelt (vgl. Abbildung 43). Über diese wird der Informationsfluss innerhalb von MIKE abgewickelt.

5.1.1 Softwarearchitektur des Schlüsselbaums

Die Zerlegung der Implementierung des Schlüsselbaums in Klassen ist durch das Konzept vorgegeben. Die Knoten des Baums werden mittels der Klasse `TreeElement` realisiert. Die Klasse `KeyTree` verknüpft die einzelnen Knoten zum Schlüsselbaum und stellt Methoden zu dessen Bearbeitung zur Verfügung. Diese können in Methoden zur Schüsselbaummanipulation, z.B. `delSubTree()`, `addSubTree()`, und Methoden zur Ermittlung von Schlüsselbaumeigenschaften, z.B. `getFilling()`, unterteilt werden. Die Bereitstellung der zur Erneuerung des Schlüsselbaums notwendigen Schlüssel sowie die Verarbeitung empfangener Schlüssel zur Aktualisierung des Schlüsselbaums, d.h. die Schlüsselbaumoperation `GetUpdateKeys` und `SetUpdateKeys`, sind vom Betriebsmodus abhängig. Diese sind deshalb in den abgeleiteten Klassen `DistKeyTree` und `AgreeKeyTree` implementiert. Die für das Modul `KeyManagement` benötigten kryptographischen Funktionen sind in den Klassen `DistKeyTree` und `AgreeKeyTree` virtuell, d.h. nur zum Überladen, deklariert. Dadurch werden die Funktionen

kryptographischer Bibliotheken nur in den Klassen `RealDistKeyTree` und `RealAgreeKeyTree` verwendet (Abbildung 59) und ein Austausch der kryptographischen Bibliothek ist einfach möglich. In MIKE wird die kryptographische Bibliothek Crypto++ [Esk06] eingesetzt.

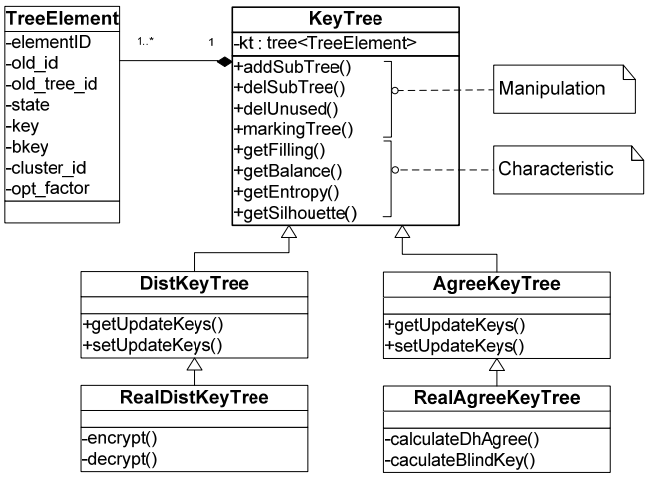

Abbildung 59: Klassendiagramm der Implementierung des Schlüsselbaums

5.1.2 Softwarearchitektur der Zustandsautomaten

Ein objektorientierter Entwurf ist mit der Zerlegung eines Systems in Klassen verbunden. Hierbei spielen viele Faktoren, z.B. Kapselung, Granularität, Flexibilität, Laufzeitverhalten etc., eine Rolle. Während die Zerlegung des Schlüsselbaums in Klassen durch das Konzept vorgegeben ist, wurde für die Zustandsautomaten der beiden Verfahren zur Schlüsselbereitstellung die Zerlegung unter Zuhilfenahme von Design Patterns vorgenommen. Ein Design Pattern beschreibt eine in der Praxis erfolgreiche, generische Lösung für ein mehr oder weniger häufig auftretendes, wiederkehrendes Entwurfsproblem und stellt eine Vorlage zur Problemlösung dar [Gam98]. Das Modul `KeyManagement` enthält die Zustandsautomaten für die Realisierung beider Betriebsarten. Zur Implementierung wurde die Vorlage State, d.h. ein Behavioral Pattern, verwendet. Eine Adaption der generischen Lösung führte zu der nachfolgend beschriebenen Struktur (Abbildung 60):

- Die Klasse `ConnectionKeyMngt` repräsentiert die Kontextklasse und definiert eine Schnittstelle zur Annahme von eingehenden Anfragen. Derartige Anfragen sind Ereignisse oder Managementnachrichten. Außerdem dient diese Klasse zum Speichern von Konfigurationen und Ergebnissen der Anfragenbearbeitung. Sie verwaltet weiterhin die Klassen mit dem zustandsspezifischen Verhalten.
- Die Klasse `StateKeyMngt` definiert eine Schnittstelle zur Kapselung des Verhaltens.
- Die Unterklassen von `StateKeyMngt`, z.B. `DistStateClUpdateWait`, implementieren das jeweilige zustandsspezifische Verhalten des Schlüsselmanagements. Diese sind die Zustandsobjekte. Das gesamte an den Zustand gebundene Verhalten ist nur in dem jeweiligen Zustandsobjekt enthalten.

Die beschriebenen Klassen interagieren in nachfolgend beschriebener Art miteinander. Eingehende Anfragen werden vom `ConnectionKeyMngt` an das aktuelle Zustandsobjekt, d.h. eine Unterklasse von `StateKeyMngt` delegiert. Hierbei übergibt sich ein Objekt der Klasse `ConnectionKeyMngt` selbst als Argument, um Parameter zur Bearbeitung der Anfrage bereitzustellen. Die Anfrage wird dann in dem Zustandsobjekt bearbeitet. Durch die Bearbeitung gewonnene Ergebnisse werden in dem Objekt der Klasse `ConnectionKeyMngt` abgespeichert. Änderungen des Zustands werden durchgeführt, indem die Zustandsobjekte in dem Kontextobjekt ausgetauscht werden.

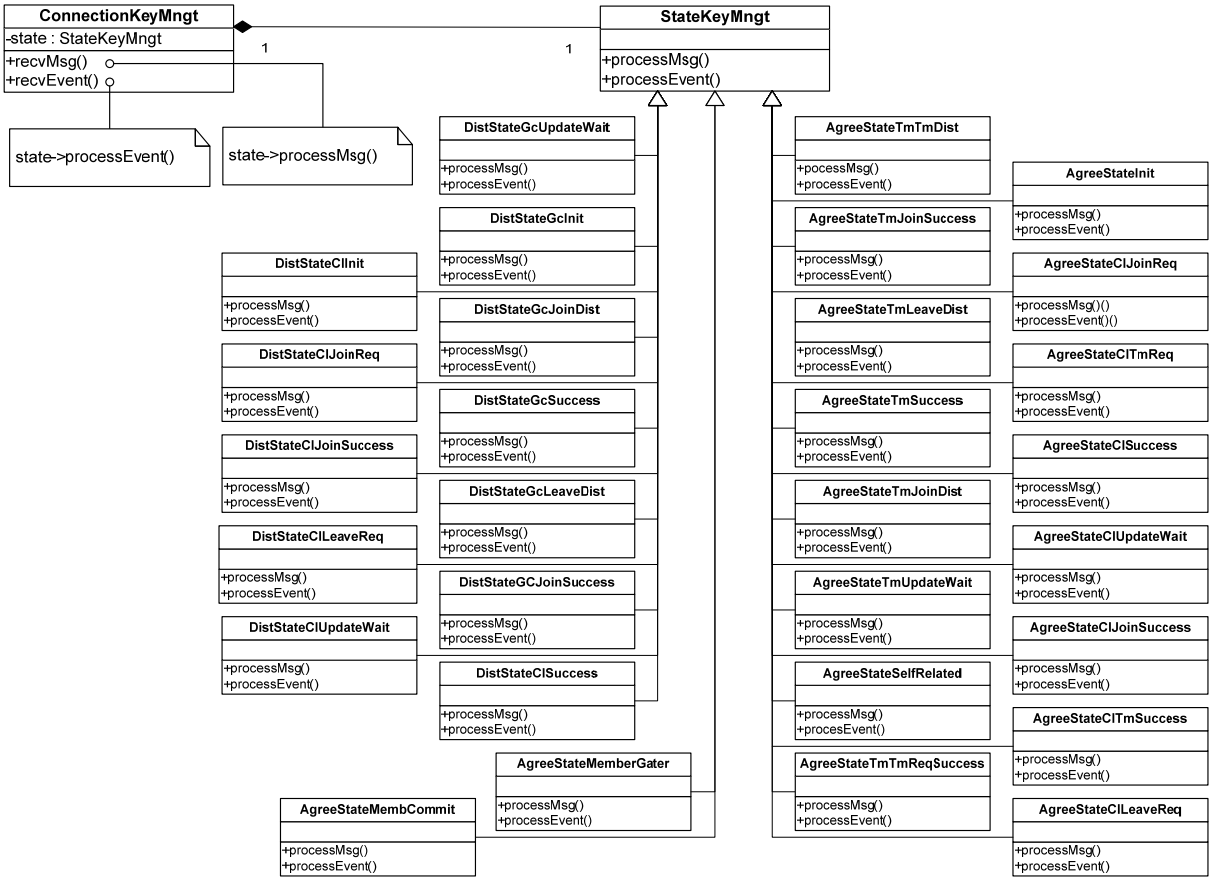

Abbildung 60: Klassendiagramm der Zustandsautomaten

5.2 Realisierung des Moduls GroupPolicyDatabase

Das Modul `GroupPolicyDatabase` stellt Methoden zur Durchsetzung des bei sicherheitsrelevanten Aufgaben festgelegten Verhaltens sowie der Berechtigung zum Gruppenbeitritt zur Verfügung. Diese sind in der Klasse `GrpPolicyDB` implementiert. Die Methoden des Moduls müssen hierzu auf einen Datenspeicher mit der Gruppensicherheitsvorschrift zugreifen, der Verhaltensregeln und Berechtigungen enthält. In der Realisierung von MIKE dient die Datei GrpPolicy.conf (Abbildung 61) als Datenspeicher. Die Sicherheitsvorschrift, die die Datei enthält, ist in einem selbst entwickelten Format spezifiziert. Alternativ wäre die der Spezifikation Ismene Policy Description Language [Pra00], Cryptographic Context Negotiation Template [Bal99] oder Group Security Policy Token Version 1 möglich. Methoden zu deren Auswertung sind in der von der Klasse

`GrpPolicyDB` abgeleiteten Klasse `FileGrpPolicyDB` implementiert. Andere Datenspeicher können durch einen Austausch der Klasse `FileGrpPolicy` einfach integriert werden.

```
[general]
#--Policy identification--
# Application data multicast address
DataGrpAddr=ff1e::2:1
#--Authorisation for group actions--
# GC/TM address
TmGcAddr=3ffe::1
#--Mechanisms for group security services--
# Key management multicast address
MngtGrpAddr=ff1e::1:1
# Tree degree (key distribution)
DistTreeDegree=3
# Tree degree (key agreement)
AgreeKeyDegree=2
# Diffie Hellman parameters
Prime=02110085118505BFB43204CABC8CC052A4AD
Generator=020102
# Group key lifetime
KeyLifetime=86400
# Data protection mechanisms
KeyInterface=McastTransportEspSA.txt
#--Access control--
# CA certificate location
CertCa=cacert.pem
# Certificate revocation list location
CrlCa=crl.pem
[3ffe::1]
# User certificate location
Cert=001cert.pem
# Signature key location
AsymPrivKey=001privkey.pem
```

Abbildung 61: Gruppensicherheitsvorschrift

5.3 Testumgebung für die Schlüsselbereitstellungsverfahren

Um mit dem erstellten Schlüsselmanagement MIKE zusätzlich grundlegende Funktionalitätstests sowie Messungen ohne den Einfluss der Netzwerkkommunikation durchführen zu können, wurden zwei Arten des Moduls `MessageDispatcher` erstellt. Das als `MikeMsgDispatcher` bezeichnete Modul ermöglicht eine Interaktion von auf verschiedenen Rechnern befindlichen MIKE-Prozessen. Wird dieses gegen das Modul `MikeTestDispatcher` ausgetauscht, können auf einem Rechner mehrere Prozesse von MIKE gestartet werden und interagieren. Für jeden dieser Prozesse wird dann eine Warteschlange mit dem Prinzip FIFO initialisiert. Der Nachrichtenaustausch erfolgt über diese Warteschlange, die den Netzwerkanschluss der MIKE-Prozesse emuliert (Abbildung 62). Wird MIKE mit einem emulierten Netzwerk betrieben, ist es möglich, in einem Selbsttest alle Verfahren auf ihre Funktionsfähigkeit zu prüfen.

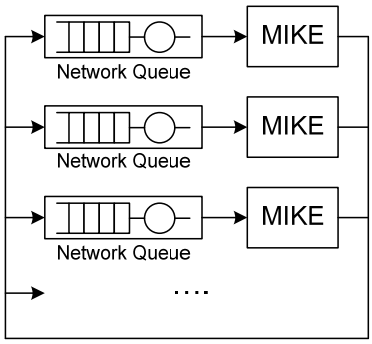

Abbildung 62: Emulation des Schlüsselmanagements MIKE auf einem Rechner

5.4 Visualisierung

Mechanismen zur Bereitstellung eines gemeinsamen Schlüssels in einer Gruppe sind sehr komplex. Zur Visualisierung dieser Mechanismen wurde deshalb für den Betrieb von MIKE bei einem emulierten Netzwerk mittels des Qt-Toolkit [Tro06] eine graphische Darstellung implementiert. Diese bietet die Möglichkeit, sowohl die Datenübertragungen als auch den Schlüsselbaum zu visualisieren. Die graphische Darstellung wird als eigener Thread gestartet. Für jedes stattfindende Ereignis wird ein Ereignisobjekt erzeugt und dem Thread zur graphischen Darstellung übermittelt. Dieser verarbeitet die Ereignisse dann sequenziell und stellt sie auf dem Kommunikationsdisplay dar. Hierbei kann zwischen permanent sichtbaren Objekten, z.B. Teilnehmern, und temporär sichtbaren Objekten, z.B. Datenübertragungen, unterschieden werden. Teilnehmer werden durch Rechtecke symbolisiert, die mit der IP-Adresse des entsprechenden Teilnehmers gekennzeichnet sind. Die Rechtecke werden in einem Kreis mit dem Radius r um den Mittelpunkt (mp_x,mp_y) des Kommunikationsdisplays angeordnet. Die Positionen werden bei U Teilnehmern wie folgt berechnet:

$$x = mp_x + r \cdot Cos(\tfrac{2\cdot\pi}{U} \cdot i) \quad y = mp_y + r \cdot Sin(\tfrac{2\cdot\pi}{U} \cdot i) \text{ mit } 0 < i \leq U - 1$$

Die Füllfarbe der Rechtecke wird je nach Zustand des Teilnehmers gewählt, d.h. entweder grün bei Kenntnis des Gruppenschlüssels oder rot, falls dieser nicht bekannt ist. Unterschiedliche graphische Symbole innerhalb der Rechtecke zeigen an, ob der Teilnehmer GC/TM oder Nutzer ist (Abbildung 63). Die Übertragung von Daten zwischen den Nutzern des Schlüsselmanagements wird durch einen Pfeil von der Quelle zum Ziel angezeigt.

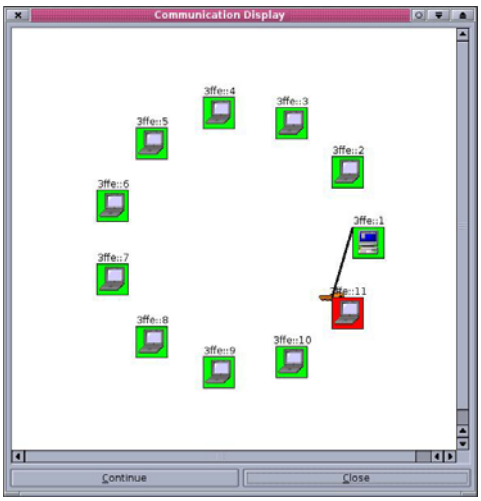

Abbildung 63: Graphische Darstellung der ausgetauschten Datenpakete

Die Anzeige des Schlüsselbaums eines Nutzers wird durch Doppelklick des entsprechenden Teilnehmersymbols ermöglicht. Hierbei sind die Knoten des Schlüsselbaums durch Kreise dargestellt. Innere Knoten des Schlüsselbaums sind schwarz gefärbt. Im Gegensatz dazu besitzen Blätter, die einem Nutzer zugeordnet sind, eine blaue Farbe. Die als Platzhalter eingesetzten Nullknoten sind grau dargestellt. Durch Doppelklick auf einen Knoten des Schlüsselbaums können die im Knoten gespeicherten Daten angezeigt werden. Diese enthalten neben den Informationen zur Position und zum Status des Knotens auch den

gespeicherten Schlüssel bzw. Blindschlüssel. Der im Wurzelknoten abgelegte Gruppenschlüssel wird zusätzlich über dem Schlüsselbaum dargestellt (Abbildung 64).

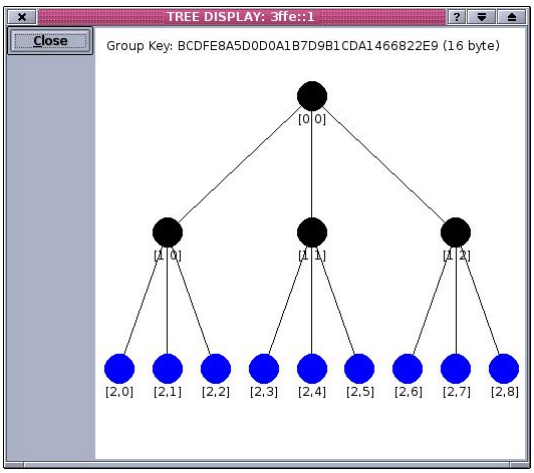

Abbildung 64: Graphische Darstellung des Schlüsselbaums

5.5 Kapitelzusammenfassung

Das Konzept MIKE wurde zur Verifizierung seiner Tragfähigkeit implementiert. In diesem Kapitel wurde die zu diesem Zweck mit der Methode des objektorientierten Designs entwickelte Software vorgestellt. Schwerpunkt der Vorstellung bildete die Softwarearchitektur des Schlüsselbaums und der Zustandsautomaten der Schlüsselbereitstellungsmechanismen. Die zur Unterstützung des Verständnisses der komplexen Abläufe in einem Gruppenschlüsselmanagement implementierte graphische Darstellung der Schlüsselbereitstellungsmechanismen wurde ebenfalls beschrieben.

6 Metriken und Szenarios zur Leistungsbewertung eines Gruppenschlüsselmanagements

Um eine Bewertung eines Gruppenschlüsselmanagements durchführen zu können, sind geeignete Bewertungskriterien zu definieren. In dieser Arbeit werden folgende Bewertungskriterien eingesetzt:

- Verlässlichkeit
  Um die Einsatzfähigkeit in kritischen Bereichen zu bewerten, ist die Verlässlichkeit (dependability) des Systems zu analysieren. Die Verlässlichkeit ist der Umfang, in dem von einem System erwartet werden kann, dass es die beabsichtigte Funktion mit der erforderlichen Genauigkeit über den Einsatzzeitraum ausführt. Quantitative Bewertungskenngrößen werden danach unterschieden, ob sie für reparierbare oder nicht-reparierbare Systeme gelten. In reparierbaren Systemen findet nach jedem Ausfall die Fehlerlokalisierung und Reparatur statt. Als Kenngröße reparierbarer Systeme wird deshalb die Reparaturzeit t_{TTR} (Time To Repair, TTR) verwendet. Diese setzt sich aus der Diagnosezeit $t_{Diagnosis}$ und der „eigentlichen“ Reparaturzeit t_{Repair} zusammen.
  $t_{TTR} = t_{Diagnosis} + t_{Repair}$
- Effizienz
  Für den Wechsel des Gruppenschlüssels ist ein schneller Mechanismus notwendig, da während des Schlüsselwechsels keine Nutzdaten ausgetauscht werden können. Aus diesem Grund ist der Zeitbedarf des Schlüsselwechsels wichtigster Gesichtspunkt der Effizienzanalyse. Der Einsatz des Schlüsselmanagements in Netzwerken mit limitierten Ressourcen fordert deren effiziente Nutzung. Deshalb ist zusätzlich die Effizienz des Schlüsselwechsels in Bezug auf benötigte Datenübertragungskapazität und Rechenleistung wichtig. Die zur Effizienzmessung verwendete Metrik wird in den nachfolgenden Abschnitten detailliert erläutert. Im Rahmen der Effizienzanalyse findet auch eine Bewertung der Flexibilität des Konzepts statt. Damit ein Schlüsselmanagement in unterschiedlichen Szenarios eingesetzt werden kann, muss es in Gruppen mit unterschiedlicher Dynamik und Größe einen effizienten Schlüsselwechsel ermöglichen.

6.1 Metriken zur Effizienzanalyse

Ein Schlüsselmanagement, das die in Abschnitt 3.2 definierten Sicherheitsanforderungen an einen Gruppenschlüssel erfüllt, muss in dynamischen Gruppen häufig einen Schlüsselwechsel durchführen. Wichtigster Gesichtspunkt der Effizienzanalyse ist deshalb die Bewertung des Schlüsselwechsels. Die hierbei eingesetzten Metriken werden in diesem Abschnitt unabhängig von einem konkreten Schlüsselmanagement definiert und anschließende die Messgrößen im Schlüsselmanagement MIKE identifiziert. Die festgelegten Metriken werden zur Effizienzanalyse in Kapitel 8 eingesetzt.

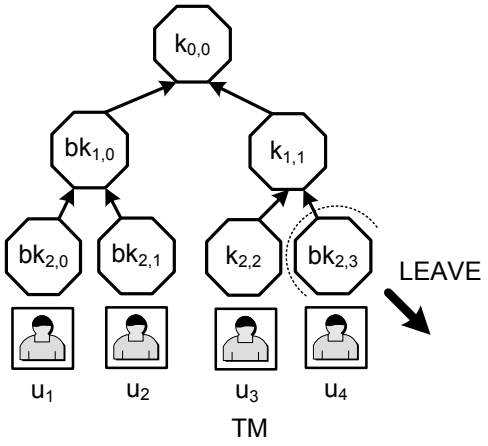

Abbildung 65: Teilnehmeroperation ohne erforderlichen Wechsel des TM

Bei der Identifizierung der Messgrößen im Schlüsselmanagement MIKE muss beim Modus Key Agreement im Gegensatz zu Key Distribution zwischen zwei Fällen unterschieden werden. Im günstigsten Fall ist bei der Durchführung des Schlüsselwechsels kein Wechsel des TM notwendig (Abbildung 65). Mit den in diesem Kapitel definierten Metriken wird der ungünstige Fall bewertet. In diesem Fall müssen beim Schlüsselwechsel zwei Nachrichten versandt werden, da auch ein Wechsel des TM notwendig ist.

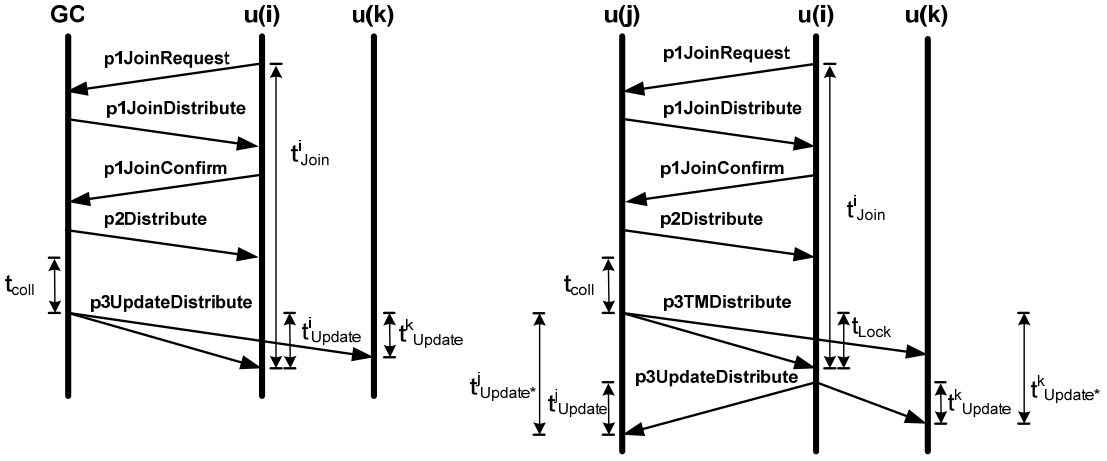

Abbildung 66: Verzögerungszeiten des Schlüsselmanagements MIKE im Modus Key Distribution (links) und Key Agreement (rechts)

Die Metriken zur Effizienzanalyse eines Schlüsselmanagements sind nachfolgend definiert und in Abbildung 66 die Messgrößen im Schlüsselmanagement MIKE veranschaulicht:

- Datenmenge für den Schlüsselwechsel
  Messwert ist die für die Etablierung eines neuen Gruppenschlüssels übertragene Datenmenge. Für den Modus Key Distribution von MIKE ist dieses die Datenmenge s_{Update}, die mit der Nachricht `p3UpdateDistribute` übertragen wird. Der Modus Key Agreement verwendet zur Übertragung der für den Schlüsselwechsel benötigten Informationen zwei Nachrichten. In diesem Fall ist die benötigte Datenmenge $s_{Update*}$ die Summe der mit den Nachrichten `p3TMDistribute` und `p3UpdateDistribute` übertragenen Datenmengen.

- Schlüsselwechselzeitdauer
  Messwert ist die Dauer für die Etablierung eines neuen Gruppenschlüssels. Hierdurch ist feststellbar, wie schnell auf Nutzeranfragen reagiert werden kann. Bei dieser Metrik muss beim Schlüsselmanagement MIKE zwischen den Betriebsmodi unterschieden werden. Im Modus Key Distribution ist dies die Zeitspanne t_{Update} für den Versand, den Empfang und die Verarbeitung der Nachricht `p3UpdateDistribute`. Zur Etablierung eines neuen Gruppenschlüssels sind im Modus Key Agreement die beiden Nachrichten `p3TMDistribute` und `p3UpdateDistribute` notwendig. In diesem Modus wird zuerst der TM für die Etablierung eines neuen Gruppenschlüssels festgelegt und anschließend der eigentliche Schlüsselwechsel vollzogen. In diesem Fall ist die Dauer des Schlüsselwechsels die Zeitspanne $t_{Update*}$.

- Streuung der Schlüsselwechselzeitdauer
  Messwert ist die Streuung der Zeitdauer für die Durchführung des eigentlichen Schlüsselwechsels. Durch den Einsatz unterschiedlicher Gruppenschlüssel zum Nutzdatenschutz können auf Grund der Streuung der für den Schlüsselwechsel benötigen Zeitdauer, Daten verworfen werden. Ein Informationsverlust durch einen inkonsistenten Gruppenschlüssel ist die Folge. Als Index der Streuung wird der Abstand zwischen dem 10- und 90-Percentile der Zeitdauer für die Durchführung des Schlüsselwechsels verwendet.

- Blockierung des Systems durch den Schlüsselwechsel
  Messwert ist die Zeitdauer, während der auf Grund der Schlüsselwechseldurchführung keine Nutzeranfragen bearbeitet werden können. Das Schlüsselmanagement MIKE ist im Modus Key Agreement während des Wechsels des TM für die Verarbeitung einer Teilnehmeroperation blockiert. Diese Zeitspanne wird mit t_{Lock} bezeichnet. Die Systemblockierung ergibt sich aus der Differenz der gesamten Übertragungs- und Verarbeitungszeit der beiden Nachrichten `p3TMDistribute` und `p3UpdateDistribute` und der Nachricht `p3UpdateDistribute`. Diese sekundäre Metrik kann somit mit $t_{Lock}=t_{Update*}-t_{Update}$ aus den bereits definierten Metriken ermittelt werden.

Ein System ist desto effizienter zu bewerten, je kleiner die mit der Metrik ermittelten Werte sind. Zur Bewertung eines Schlüsselmanagementsystems werden in [Ami02] noch die nachfolgenden zwei Metriken verwendet:

- Zeitdauer für den Gruppenbeitritt
  Die gesamte Dauer für den Gruppenbeitritt ist als die Zeitspanne vom Versand der Anfrage zum Gruppenbeitritt bis zur Etablierung des neuen Gruppenschlüssels definiert. Bei dem im Rahmen dieser Arbeit erstellten Schlüsselmanagement MIKE ist dieses die Zeitspanne t_{Join} vom Versand der Nachricht `p1JoinRequest` bis zum Empfang und der Verarbeitung der Nachricht `p3UpdateDistribute`.

- Zeitdauer für den Gruppenaustritt
  Die gesamte Dauer für den Gruppenaustritt ist als die Zeitspanne vom Versand der Anfrage zum Gruppenaustritt bis zur Etablierung des neuen Gruppenschlüssels definiert. Beim Schlüsselmanagement MIKE ist dieses die Zeitspanne t_{Leave} vom Versand der

Nachricht `p1LeaveRequest` bis zum Empfang und der Verarbeitung der Nachricht `p3UpdateDistribute`.

Nachteil dieser beiden Metriken ist, dass die ermittelten Messwerte von dem eingesetzten Verfahren zur Authentisierung der Teilnehmer während der Beitritts- bzw. Austrittsanfrage dominiert werden. Dieses verhindert eine exakte Bewertung der Mechanismen des Schlüsselwechsels. Einen Überblick über die aus den definierten Metriken resultierenden Messgrößen in MIKE vermittelt Tabelle 19.

t^i_{Join}	Gesamte Zeitdauer für den Gruppenbeitritt ermittelt vom Nutzer u_i
t^i_{Leave}	Gesamte Zeitdauer für den Gruppenaustritt ermittelt vom Nutzer u_i
t_{coll}	Zeitdauer der Sammelphase von Nutzeranfragen
t^i_{Update}	Zeitdauer der Übertragung und Verarbeitung der Nachricht `p3UpdateDistribute` nach einer Teilnehmeroperation ermittelt vom Nutzer u_i
s_{Update}	Größe der Nachricht `p3UpdateDistribute`
$t^i_{Update*}$	Zeitdauer der Übertragung und Verarbeitung der Nachricht `p3TMDistribute` und `p3UpdateDistribute` nach einer Teilnehmeroperation ermittelt vom Nutzer u_i (nur im Modus Key Agreement)
$s_{Update*}$	Größe der Nachricht `p3TMDistribute` und `p3UpdateDistribute` (nur im Modus Key Agreement)
t_{Lock}	Zeitdauer nach einer Teilnehmeroperation, in der keine Nutzeranfragen angenommen werden (nur im Modus Key Agreement)

Tabelle 19: Überblick über die Messgrößen in MIKE zur Effizienzanalyse

6.2 Einfachere Metriken zur Effizienzanalyse

Um eine Effizienzanalyse mit den im vorherigen Abschnitt definierten Metriken durchführen zu können, müssen aufwändige verteilte Messungen durchgeführt werden. Zur Abschätzung der Schlüsselwechseleffizienz werden deshalb alternativ einfachere Metriken definiert. Die festgelegten Metriken werden zur Effizienzanalyse in Kapitel 7 eingesetzt. Die für den Schlüsselwechsel benötigte Datenübertragungskapazität und Rechenleistung lässt sich mit den nachfolgenden Metriken abschätzen:

- Anzahl der übertragenen Schlüsselpakete für den Schlüsselwechsel
  Beim Schlüsselwechsel ausgetauschte Nachrichten transportieren Schlüsselpakete. Diese enthalten die Hilfsschlüssel aus dem Schlüsselbaum sowie Verwaltungsinformation. Über diese Anzahl, die Schlüssellänge der Hilfsschlüssel und die Größe der Verwaltungsinformationen lässt sich die Größe der im Rahmen des Schlüsselwechsels ausgetauschten Nachrichten abschätzen.
- Anzahl übertragener Nachrichten für den Schlüsselwechsel
  Dieser Messwert in Kombination mit der Anzahl der Schlüsselpakete kann zur Abschätzung der benötigten Datenübertragungskapazität verwendet werden.
- Anzahl der kryptographischen Operationen für den Schlüsselwechsel
  Die Anzahl der kryptographischen Operationen des Gruppenverwalters, die zur Bereitstellung des Gruppenschlüssels notwendig sind, bestimmt die hierfür benötigte Rechenleistung. Im Modus Key Distribution von MIKE entsprechen die

kryptographischen Operationen beim Schlüsselwechsel Verschlüsselung von Hilfsschlüsseln. Da hierbei symmetrische Verschlüsselungsverfahren eingesetzt werden, ist die benötigte Rechenleistung gering. Die Anzahl der in diesem Modus durchgeführten Operationen ist identisch mit der Anzahl der ausgetauschten Schlüsselpakete. Im Gegensatz dazu hat die Anzahl der kryptographischen Operationen im Modus Key Agreement einen erheblichen Einfluss auf die benötigte Rechenleistung. In diesem Modus muss der TM zur Gruppenschlüsselbereitstellung für jeden Knoten des Pfades zur Wurzel zwei kryptographische Operationen durchführen. Eine Operation ist zur Berechnung des Schlüssels mittels des DH-Algorithmus notwendig. Die zweite kryptographische Operation dient der Ermittlung des Blindschlüssels. Die hierfür benötigte Zeit ist insbesondere bei größeren Schlüssellängen erheblich (Tabelle 12). Ursache der hohen Rechenleistung ist, dass bei beiden Berechnungen Exponentiationen durchgeführt werden müssen.

Ein System ist als effizienter zu bewerten, je kleiner die mit der Metrik ermittelten Werte sind.

6.3 Modellierung des Nutzerverhaltens zur Effizienzanalyse

Zur Messung der Effizienz muss ein Gruppenschlüsselmanagement mit einer Last konfrontiert werden. Diese besteht aus Nutzeranfragen zum Gruppenbeitritt bzw. Gruppenaustritt und ist eine Folge des Nutzerverhaltens. Es wurden folgende drei Arten von Nutzerverhalten modelliert:

- Synthetisches Nutzerverhalten
- Ziviles Nutzerverhalten
- Militärisches Nutzerverhalten

Als synthetisches Nutzerverhalten (Synthetic Benchmark) (Abbildung 67) wird der sukzessive Beitritt von Teilnehmern zu einer Gruppe und anschließend der sukzessive Austritt der Teilnehmer in umgekehrter Reihenfolge bezeichnet.

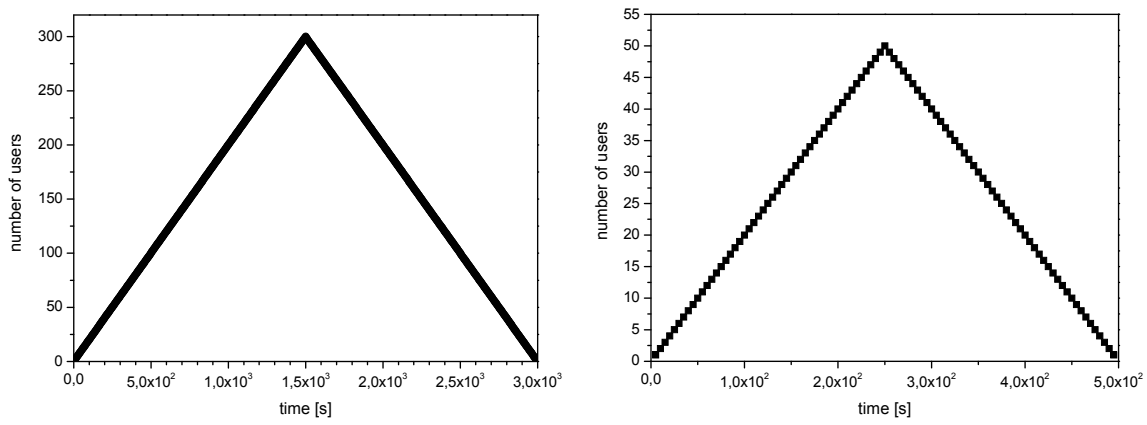

Abbildung 67: Synthetisches Nutzerverhalten mit 300 (links) und 50 (rechts) Teilnehmern

Mit diesem Nutzerverhalten lässt sich die Effizienz des Verfahrens bei verschiedenen Gruppengrößen und optimal balanciertem Schlüsselbaum bestimmen. In der Abbildung 67 ist

ein beispielhafter Verlauf des synthetischen Nutzerverhaltens mit einer maximalen Gruppengröße von 50 bzw. 300 Teilnehmern dargestellt. Die Teilnehmerzahl 300 wurde in Abschnitt 3.2 als maximal auftretende Gruppengröße identifiziert. Zur Bewertung der Effizienz eines Gruppenschlüsselmanagements wäre es jedoch unzureichend, nur synthetisches Nutzerverhalten zu betrachten. Vielmehr müssen diese Untersuchungen mit einer Folge von realistischen Nutzeranfragen durchgeführt werden. Zu diesem Zweck ist ein ziviles und ein militärisches Nutzerverhalten modelliert worden. Das zivile Nutzerverhalten (Civil Benchmark) basiert auf Untersuchungen von Audioübertragungen im MBONE, dem weltweiten Multicast-Netzwerk [Alm96]. Die Untersuchung liefert das Ergebnis, dass das Nutzerverhalten bei derartigen Übertragungen durch Wahrscheinlichkeitsverteilungen beschrieben werden kann. Die Zeitdauer zwischen zwei Beitrittsanfragen wird hierbei durch eine Exponentialverteilung modelliert. Folglich ist die Zeitdauer zwischen zwei Beitrittsanfragen unabhängig von der vorherigen. Die Zeitdauer der Gruppenmitgliedschaft wird durch eine Zipfverteilung beschrieben [Alm96]. Diese einer Hyperbel ähnelnde Verteilung wurde ausgewählt, weil kurze Teilnahmeperioden sehr viel häufiger auftreten als lange Teilnahmezeiten. Das zivile Nutzerverhalten beschreibt das Teilnehmerverhalten im Zeitraum 0-116400s. Um ein ziviles Nutzerverhalten für Gruppen mit unterschiedlicher Teilnehmeranzahl zur Verfügung zu haben, werden die in Tabelle 20 beschriebenen Parameter aus [Alm96] verwendet.

Nutzerverhalten	Ankunftsrate der Anfrage [1/min]	Mittlere Teilnahmedauer [min]	Durchschnittliche Teilnehmerzahl
Ziviles Nutzerverhalten Nr. 1	1/7,51	258	34
Ziviles Nutzerverhalten Nr. 2	1/1,24	150	121

Tabelle 20: Parameter des zivilen Nutzerverhaltens

In Abbildung 68 ist der Verlauf der Gruppengröße in Abhängigkeit von der Zeit für das zivile Nutzerverhalten mit den Parametern aus Tabelle 20 dargestellt.

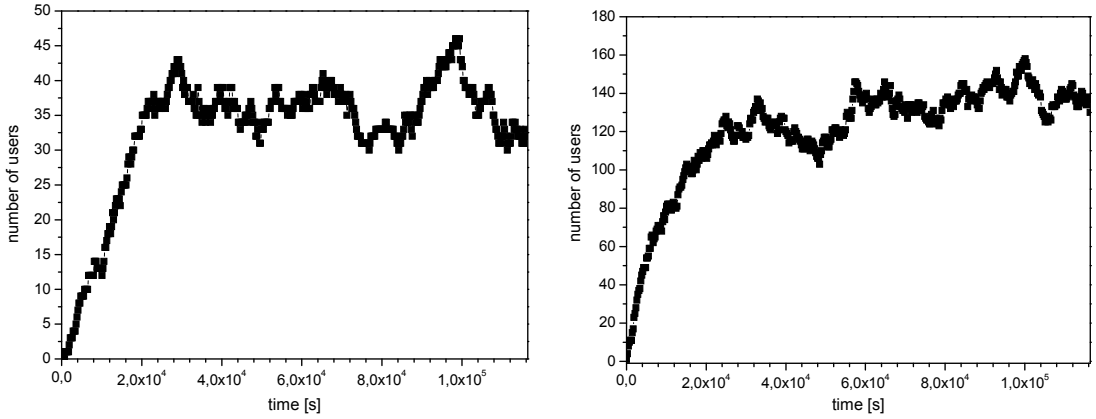

Abbildung 68: Ziviles Nutzerverhalten Nr. 1 (links) und Nr. 2 (rechts)

Das militärische Nutzerverhalten (Military Benchmark) beschreibt Nutzeranfragen zum Gruppenbeitritt bzw. Gruppenaustritt bei einem Einsatz von Streitkräften bei einem Auslandseinsatz im Rahmen von Konfliktverhütung und Krisenbewältigung. Es basiert auf der Auswertung des in [Ebe03] erstellten fiktiven militärischen Szenarios unter dem

Gesichtspunkt der Gruppenkommunikation [Cit05]. Konkret wird das Auftreten von Nutzeranfragen während der Bewachung eines Kontrollpunktes bei einem Auslandseinsatz beschrieben. Hierbei handelt es sich um einen Standarddienst bei einem derartigen Einsatz. Es wird angenommen, dass für alle an der Operation beteiligten Nutzer so lange eine Verbindung gewährleistet ist, wie sie an der sicheren Gruppenkommunikation des Kontrollpunktes teilnehmen. Hierzu werden sie von einem Schlüsselmanagement mit einem Gruppenschlüssel versorgt. Diese Verbindung stellen die beteiligten Nutzer über ein funkbasiertes Übertragungsmedium her. Für dessen Reichweite wird die typische Reichweite von VHF, d.h. 10 km, angenommen. An dem Kontrollpunkt tritt nach der Zeit t_{Start} folgendes Nutzerverhalten auf:

- Eine Fahrzeugpatrouille bestehend aus 15 Teilnehmern trifft am Kontrollpunkt ein. Es wird angenommen, dass diese sich mit einer Geschwindigkeit von 30 km/h bewegen. Aus der Geschwindigkeit und der Reichweite des funkbasierten Übertragungsmediums kann man ermitteln, dass eine Gruppenkommunikation der Fahrzeugpatrouille mit den Nutzern am Kontrollpunkt im Zeitraum $t_{Start}+t_{Patrol}$-1200s bis $t_{Start}+t_{Patrol}$+1200s stattfindet.
- Zur Abwehr eines bevorstehenden Angriffs verstärken Unterstützungstruppen bestehend aus 36 Teilnehmern den Kontrollpunkt. Zum Zeitpunkt $t_{Start}+ t_{JoinReinforcement}$ nehmen diese an der Gruppenkommunikation teil.
- Während des Angriffs von $t_{Start}+t_{Attack}$ bis $t_{Start}+t_{Attack}$+1200s werden bedingt durch Ausfall 25 % der Teilnehmer von der Gruppenkommunikation ausgeschlossen. Mit einer derartigen Ausfallrate muss in einem Einsatzland mit erhöhter Gefährdung gerechnet werden [Cit05].
- Nach Beendigung des Angriffs verlassen die Unterstützungstruppen den Kontrollpunkt. Zum Zeitpunkt $t_{Start}+t_{LeaveReinforcement}$ beenden die Unterstützungstruppen die Gruppenkommunikation.

Die Zeitpunkte t_{Patrol}, $t_{JoinReinforcement}$, t_{Attack}, und $t_{LeaveReinforcement}$ wurden so gewählt, dass es keine zeitlichen Überschneidungen gibt. Zwischen den genannten Ereignissen findet keine Änderung der Gruppengröße und Zusammensetzung statt. Die Zeitdauer zwischen zwei stattfindenden Ausschlüssen während des Angriffs wird durch eine Exponentialverteilung modelliert. Hierdurch wird die Unabhängigkeit zwischen zwei Ausschlüssen beschrieben. Das militärische Nutzerverhalten beschreibt das Teilnehmerverhalten im Zeitraum 0-9600 s. Um ein militärisches Nutzerverhalten für Gruppen mit unterschiedlicher Teilnehmeranzahl zur Verfügung zu haben, werden die in Tabelle 21 beschriebenen Parameter verwendet.

Nutzerverhalten	Teilnehmer am Kontrollpunkt	Ankunftsrate der Anfrage [1/min]	Anzahl der ausgeschlossenen Teilnehmer
Militärisches Nutzerverhalten Nr. 1	30	1/4	16
Militärisches Nutzerverhalten Nr. 2	90	1/4	31

Tabelle 21: Parameter des militärischen Nutzerverhaltens

In Abbildung 69 ist der Verlauf der Gruppengröße in Abhängigkeit von der Zeit für das militärische Nutzerverhalten mit den Parametern aus Tabelle 21 dargestellt.

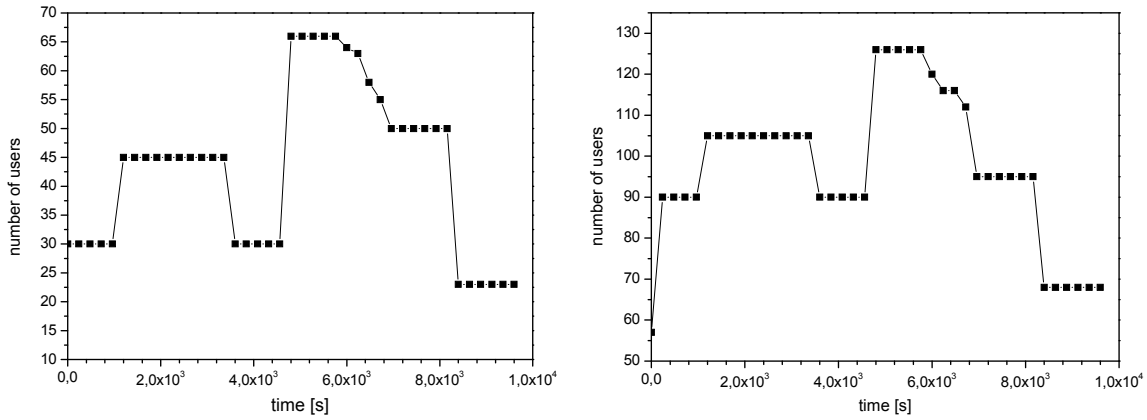

Abbildung 69: Militärisches Nutzerverhalten Nr. 1 (links) und Nr. 2 (rechts)

Um ein Schlüsselmanagement mit den drei beschriebenen Arten des Nutzerverhaltens zu konfrontieren, wurde für jedes Nutzerverhalten ein Lastgenerator realisiert. Diese Lastgeneratoren füllen eine Warteschlange mit den durchzuführenden Operationen. Die Elemente der Warteschlange bestehen aus dem Tripel Zeitstempel, IP-Adresse des Ausführenden und Nutzeranfrage. Sollen die Lastgeneratoren zur Effizienzanalyse eingesetzt werden, muss eine Schnittstelle zur Verarbeitung der Ereigniswarteschlange in das zu untersuchende Schlüsselmanagement integriert werden. Diese Schnittstellte sorgt dafür, dass das Schlüsselmanagement an den durch die Zeitstempel festgelegten Zeitpunkten die entsprechende Teilnehmeroperation ausführt (Abbildung 70).

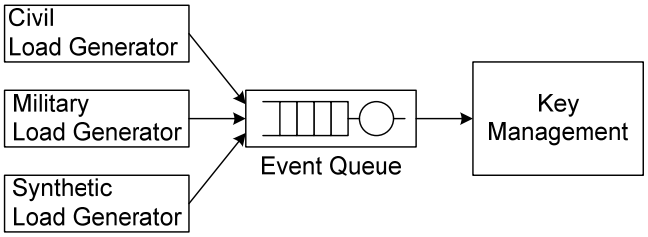

Abbildung 70: Anschluss der Lastgeneratoren an ein Schlüsselmanagement

6.4 Kapitelzusammenfassung

Zur Leistungsbewertung eines Schlüsselmanagements werden die zwei Bewertungskriterien Verlässlichkeit und Effizienz festgelegt. Soll ein Schlüsselmanagement in Netzwerken mit beschränkten Ressourcen eingesetzt werden, ist die Effizienzanalyse ein wichtiger Aspekt. Diese wird in dynamischen Gruppen durch die Effizienz des Schlüsselwechsels dominiert. Inhalte der für die Effizienzanalyse des Schlüsselwechsels definierten Metrik sind die Bestimmung der Dauer des Schlüsselwechsels, der Informationsverlust durch einen inkonsistenten Gruppenschlüssel und die Blockierungsdauer durch den Schlüsselwechsel. Zusätzlich wurde eine einfacher zugängliche Metrik definiert, die ebenfalls eine Abschätzung der Effizienz des Schlüsselwechsels ermöglicht. Die Notwendigkeit für die Effizienzanalyse, ein Schlüsselmanagement mit einer Last zu konfrontieren, führte zu der Entwicklung von drei Lastgeneratoren. Diese erzeugen Nutzeranfragen entsprechend eines synthetischen, zivilen und militärischen Nutzerverhaltens.

7 Szenariospezifische Optimierung des Schlüsselmanagements MIKE

Eine Leistungsverbesserung für das Schlüsselmanagement MIKE hinsichtlich der benötigten Datenübertragungskapazität bzw. Rechenleistung kann erzielt werden, wenn es an die militärische Verwendung angepasst wird. Hierzu wurden zwei Optimierungen eingeführt. Die Idee der Optimierungen besteht in der Nutzung des Wissens, dass militärische Gruppen aus Teilgruppen mit bekannten Verhaltensmustern bestehen und Zeiträume mit starker Fluktuation auftreten. Der Aufbau militärischer Einheiten aus Teilgruppen mit bekanntem Verhalten ergibt sich aus der hierarchischen Organisationsstruktur der Streitkräfte. Starke Fluktuationen sind zu erwarten, wenn zum Beispiel Truppenteile aus militärischen Verbänden herausgenommen und entsprechend der zu erwartenden Aufgaben anderen Verbänden zugeteilt werden. Die folgenden zwei Verbesserungen nutzen dieses militärische Gruppenverhalten aus:

- Sammelverarbeitung von Nutzeranfragen
  Dieser Mechanismus sammelt Nutzeranfragen, verarbeitet diese gemeinsam und verteilt im Anschluss einen neuen Gruppenschlüssel. Dieses bedeutet, dass der Gruppenschlüssel nicht nach jeder Änderung der Teilnehmeranzahl gewechselt wird.
- Nutzerverhaltensbasierte Schlüsselbaumkonstruktion
  Dieses Optimierungskonzept sorgt dafür, dass sehr dynamische Nutzer näher an der Wurzel des Schlüsselbaums platziert werden. Der verringerte Abstand zur Wurzel bewirkt einen geringeren Aufwand beim Schlüsselwechsel.

Die Idee beider Optimierungen besteht in der Verbesserung der Verarbeitung des Schlüsselbaums. Aus diesem Grund bewirken die Optimierungen eine Verbesserung beider Betriebsmodi [Aur05]. Verbesserungen, die nur einen Betriebsmodus betreffen, in dem sie dessen spezifische Eigenschaften ausnutzen, werden nicht betrachtet. Dennoch könnten derartige bereits existierende Verbesserungen in das Schlüsselmanagement integriert werden. Weiterhin werden bei den Optimierungen keine Annahmen über den Grad des Schlüsselbaums getroffen. Im Betriebsmodus Key Distribution kann der Grad frei gewählt werden. In Abschnitt 7.2 zeigt sich, dass durch eine falsche Wahl des Grads des Schlüsselbaums ein deutlicher Effizienzverlust auftreten kann. Wie bereits erwähnt, wurden bisher Optimierungen entweder für zentrale oder verteilte Schlüsselbereitstellungsverfahren entworfen nicht aber für beide gleichzeitig. Zum Beispiel verbessern die Konzepte in [Yan01], [Poo01] und [Kwa06] die Effizienz von zentralen Verfahren zur Schlüsselvereinbarung, während mit den Methoden in [Lia06] und [Wan05] eine Leistungssteigerung von verteilten Verfahren erzielt wird.

Soll das Schlüsselmanagement an die Verwendung in Streitkräften angepasst werden, so muss eine Schlüsselbereitstellung auch möglich sein, wenn Teilnehmer temporär nur über eine Simplexverbindung verfügen. Diese Situation entsteht beim Betrieb der Kommunikationsmittel im Zustand EMCON, bei dem zum Schutz vor Aufkärungsmaßnahmen Daten empfangen aber nicht versendet werden dürfen. Die folgende vorgestellte Verbesserung ermöglicht den Betrieb in einer solchen Kommunikationsstruktur:

- Ressourcengesteuerte Auswahl des TM/GC
  Dieses Optimierungskonzept sorgt dafür, dass Nutzer, die nur über eine Simplexverbindung verfügen, nicht als TM/GC ausgewählt werden.

7.1 Messaufbau zur Effizienzanalyse der szenariospezifischen Optimierungen

Um eine Effizienzanalyse der szenariospezifischen Optimierungen ohne den Einfluss der Netzwerkkommunikation durchführen zu können, werden die MIKE-Prozesse über Warteschlangen, die den Netzwerkanschluss der jeweiligen Prozesse emulieren, verbunden. Diese Möglichkeit des Nachrichtenaustauschs wurde bereits in Abschnitt 5.3 vorgestellt. Zusätzlich wurde in MIKE eine Schnittstellte zur Verarbeitung der Ereigniswarteschlange der in Abschnitt 6.3 vorgestellten Lastgeneratoren integriert. Bei entsprechender Konfiguration stellt diese sicher, dass an den durch die Zeitstempel festgelegten Zeitpunkten die entsprechende Teilnehmeroperation ausgeführt wird. Ein einfaches Messsystem in MIKE ermittelt die Anzahl der kryptographischen Operationen, der ausgetauschten Schlüssel und der übertragenen Nachrichten (Abbildung 71).

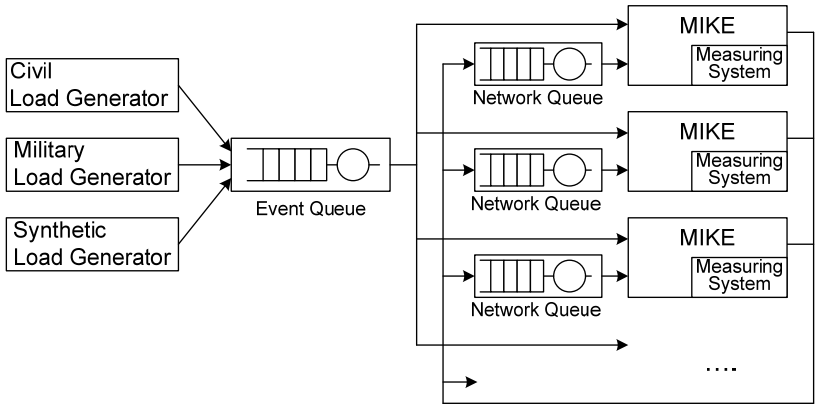

Abbildung 71: Messaufbau zur Effizienzanalyse der szenariospezifischen Optimierungen

7.2 Optimaler Grad des Schlüsselbaums

Bevor mit der Darstellung der Sammelverarbeitung von Nutzeranfragen bzw. der nutzerverhaltensbasierten Schlüsselbaumkonstruktion begonnen wird, wird zunächst der optimale Grad d des Schlüsselbaums in Gruppen mit bis zu 300 Teilnehmern ermittelt. Im Modus Key Agreement ist der Grad des Schlüsselbaums durch das Verfahren auf d=2 festgelegt. Im Betriebsmodus Key Distribution hingegen kann der Grad frei gewählt werden. Dies ist abhängig von der Größe der Gruppe, in der der Schlüsselbaum verwendet wird. Zur Bestimmung des optimalen Grads des Schlüsselbaums wird die Anzahl der ausgetauschten Schlüsselpakete bei der Teilnehmeroperation LEAVE in einer Gruppe der Größe von 2 bis 300 Teilnehmer gemessen (Abbildung 72, links). Anschließend wird die Anzahl der ausgetauschten Schlüsselpakete für alle Gruppengrößen summiert (Abbildung 72, rechts). Diese Summe ist bei der Teilnehmeroperation LEAVE für den Schlüsselbaum vom Grad drei am geringsten. Allerdings ist der Unterschied bei Verwendung eines Baums mit einem um eins erhöhten bzw. verkleinerten Grad gering. Bei der Verwendung eines Schlüsselbaums von Grad d=5 zeigt sich allerdings ein deutlicher Effizienzverlust. Die Analyse einer anderen

Teilnehmeroperation hinsichtlich des optimalen Grads des Schlüsselbaums würde das gleiche Ergebnis liefern. In den nachfolgenden Untersuchungen wird deshalb im Modus Key Distribution ein Baum vom Grad drei eingesetzt. Weiterhin ist in Abbildung 72 zu erkennen, dass sich die Anzahl der ausgetauschten Schlüsselpakete stufenweise erhöht. Die Stufen entstehen, wenn im Schlüsselbaum eine neue Ebene eingefügt wird. Jede dieser Stufen besitzt d-1 Niveaus.

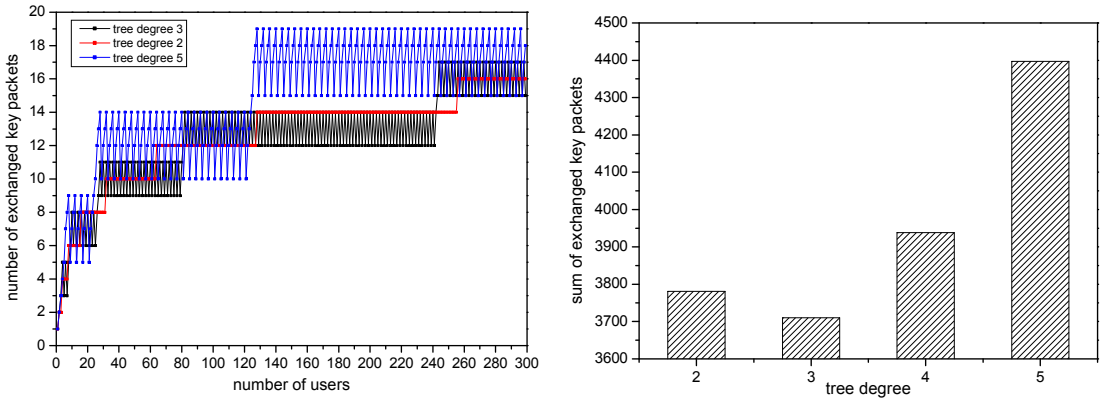

Abbildung 72: Ausgetauschte Schlüsselpakete bei unterschiedlichen Gruppengrößen (links) und Summe der ausgetauschten Schlüsselpakete (rechts) bei der Teilnehmeroperation LEAVE im Modus Key Distribution

7.3 Sammelverarbeitung von Nutzeranfragen

Im Idealfall sollte ein Nutzer, der beabsichtigt, die Gruppe zu verlassen, sofort von der Gruppe ausgeschlossen werden, und ein neuer Nutzer sollte in eine Gruppe so bald wie möglich nach seiner Anfrage integriert werden. Dies bedeutet einen sofortigen Schlüsselwechsel nach jedem Gruppenbeitritt bzw. –austritt, d.h. zum Zeitpunkt T_i wird der neue Gruppenschlüssel k_i verteilt. Eine derartige Verarbeitung von Nutzeranfragen wird als Einzelverarbeitung (single rekeying) bezeichnet. Einen Schlüsselwechsel sofort nach jeder Änderung der Gruppenteilnehmeranzahl durchzuführen, hat Nachteile, wenn in einem kurzen Zeitraum eine Vielzahl von Teilnehmeroperationen, d.h. MULTIPLE JOINs bzw. MULTIPLE LEAVEs, auftreten. Die Gründe hierfür sind die große benötigte Rechenleistung, der Verbrauch von Datenübertragungskapazität und der mögliche Verlust eines gemeinsamen Gruppenschlüssels durch die Vielzahl in kurzer Zeit durchgeführter Schlüsselwechsel. Eine Sammelverarbeitung (batched rekeying) von Nutzeranfragen ist in diesem Fall eine geeignete Verarbeitungsmethode (Abbildung 73). Hierbei werden Anfragen zum Gruppenaustritt bzw. -eintritt über einen vorher festgelegten Zeitraum gesammelt und anschließend gemeinsam verarbeitet, d.h. nach der Sammelphase $\Delta\tilde{T}=\tilde{T}_i-\tilde{T}_{i-1}$ wird der Gruppenschlüssel $\tilde{k}_i$ verteilt. Bei der Sammelverarbeitung lassen sich ereignisgesteuerte und periodische Sammelverarbeitung unterscheiden. Während bei der ereignisgesteuerten Sammelverarbeitung der Schlüsselwechsel durch ein Ereignis, z.B. durch die r-te Nutzeranfrage, initiiert wird, erfolgt bei der periodischen Sammelverarbeitung regelmäßig nach einem vorher festgelegten Zeitraum ein Schlüsselwechsel. Eine Kombination von ereignisgesteuerter und periodischer Sammelverarbeitung ist möglich. Dies würde in Gruppen mit geringer Dynamik verhindern, dass eine große Verzögerung beim Gruppenbeitritt bzw. –austritt entsteht. Da der TM/GC

nicht sofort einen Schlüsselwechsel durchführt, bleibt ein Teilnehmer, der eine Gruppe verlassen möchte, bis zum Ende der Sammelphase in der Gruppe (Leave Bonus) und ein neuer Teilnehmer muss bis zu deren Ende warten, bis er in die Gruppe aufgenommen wird (Join Latency). Die Sammelverarbeitung von Nutzeranfragen ist ein Kompromiss zwischen Leistungsfähigkeit und Sicherheit. Die Sicherheitsvorschriften Forward Secrecy und Backward Secrecy werden in einem System, das eine Sammelverarbeitung von Nutzeranfragen durchführt, nicht strikt erfüllt.

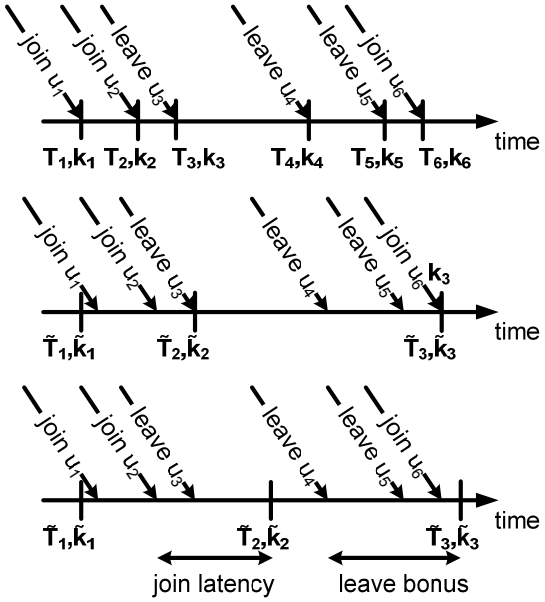

Abbildung 73: Einzelverarbeitung (oben), ereignisgesteuerte (Mitte) und periodische (unten) Sammelverarbeitung von Nutzeranfragen

7.3.1 Algorithmus für die Sammelverarbeitung von Nutzeranfragen

Um eine Sammelverarbeitung von Nutzeranfragen zu ermöglichen, ist ein Algorithmus für den Schlüsselwechsel zu entwerfen. Im Falle des zentralen Verfahrens wurde die Sammelverarbeitung bereits als Mittel zur Effizienzsteigerung erkannt und schon ein Algorithmus vorgeschlagen [Yan01]. Im Rahmen der bisherigen Entwicklung von verteilten Mechanismen zur Bereitstellung von Gruppenschlüsseln werden in kurzer Zeit auftretende Nutzeranfragen als potentielle Fehlerquelle betrachtet [Kim00] und nicht als Möglichkeit zur Durchführung einer Sammelverarbeitung.

Die Herausforderung bei der Integration einer Sammelverarbeitung von Nutzeranfragen in das Konzept MIKE besteht darin, für beide Betriebsmodi einen derartigen Algorithmus zu ermöglichen.

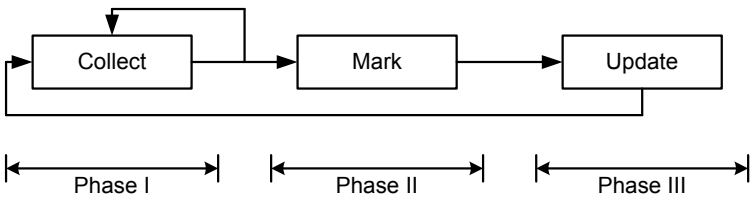

Abbildung 74: Schema des Sammelverarbeitungsalgorithmus

Der hierfür entworfene Algorithmus kann in die drei Phasen Sammelphase, Markierungsphase und Schlüsselaktualisierungsphase eingeteilt werden (Abbildung 74). Beim Entwurf des Algorithmus wurde das Ziel verfolgt, dass dieser sich für die Betriebsmodi nur in der Schlüsselaktualisierungsphase, d.h. in Phase drei, unterscheidet. Die Funktion des Sammelverarbeitungsalgorithmus wird im Folgenden detaillierter erläutert.

Sammelphase

Während der Sammelphase werden eingehende Nutzeranfragen gesammelt. Am Ende dieser Phase werden die Nutzerblätter von Teilnehmern, die die Gruppe verlassen, im Baum gelöscht. Für Teilnehmer, die der Gruppenkommunikation beitreten, werden neue Knoten in den Schlüsselbaum eingefügt. Hierbei werden zuerst alle Nutzerblätter gelöscht, bevor die Neuen eingefügt werden. Die Wahl des Einfügepunkts neuer Knoten wird in Abschnitt 7.4 diskutiert.

Markierungsphase

Nachdem der TM/GC die eingehenden Anfragen gesammelt hat, muss ermittelt werden, welche Schlüssel des Baums hinzugefügt, gelöscht oder geändert werden müssen. Als Folge der mehrfachen Anfragen gibt es mehrere Pfade im Schlüsselbaum, die zu aktualisieren sind. Zu diesem Zweck wurde ein Markierungsalgorithmus entwickelt, der auf den Knotenzuständen NEW, REPLACED, DELETE, UPDATE1, UPDATE2 und UNMOD basiert. Zur Kennzeichnung einer notwendigen Knotenaktualisierung sind zwei Zustände notwendig, da im Modus Key Distribution die Schlüsselbaumerneuerung nach einem Gruppenbeitritt anders durchgeführt wird als nach einem Gruppenaustritt. Zur Ermittlung des Knotenzustands werden die Zustände der Kinder ausgewertet. Der Markierungsalgorithmus wurde mit dem Ziel entworfen, dass er für beide Betriebsmodi einsetzbar ist. Er wird bei allen Nutzerblättern $\{v_i \mid v_i \in KT \wedge v_i \text{ != UNMOD}\}$ angewandt, d.h. $\lfloor L\text{-}J \rfloor$-mal, wenn die Anzahl der JOINs mit J und die Anzahl der LEAVEs mit L bezeichnet wird. Der Markierungsalgorithmus besitzt die Komplexität $O(Log_d(U))$, wenn er in einen Schlüsselbaum mit dem Grad d und einer Nutzeranzahl U eingesetzt wird. Der nachfolgende Pseudocode beschreibt die Arbeitsweise des Markierungsalgorithmus:

```
for each w_k ∈ path(v_i) do
   switch (w_k-1)
      case NEW
         if w_k = = UNMOD then do w_k = UPDATE2
         if w_k = = UPDATE1 then do w_k = UPDATE1
         if w_k = = UPDATE2 then do w_k = UPDATE2
         if w_k = = NEW then do w_k = UPDATE2
      case DELETE
         w_k = UPDATE1
      case REPLACED
         w_k = UPDATE1
      case UPDATE1
          w_k = UPDATE1
      case UPDATE2
         if w_k = = UNMOD then do w_k = UPDATE2
         if w_k = = UPDATE1 then do w_k = UPDATE1
```

if w_k = = UPDATE2 then do w_k = UPDATE2
if w_k = = NEW then do w_k = UPDATE2

Ein Beispiel für die Funktionsweise des Markierungsalgorithmus ist in Abbildung 75 dargestellt. In dieser ist ein markierter Schlüsselbaum abgebildet, nachdem zwei Nutzer die Gruppe verlassen haben und ein neuer Nutzer hinzugekommen ist.

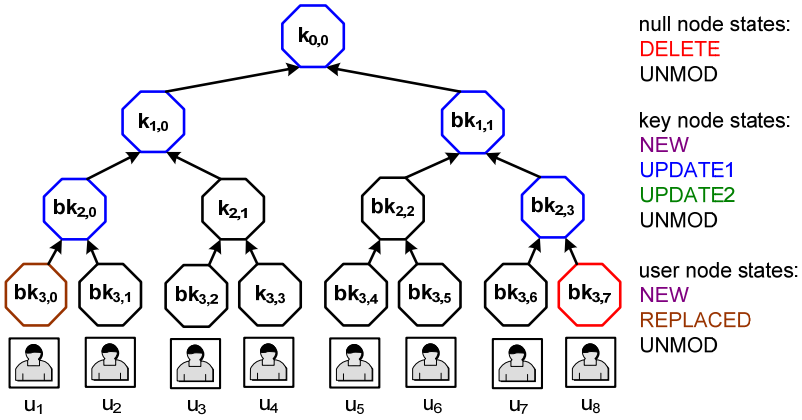

Abbildung 75: Markierter Schlüsselbaum bei der Sammelverarbeitung von zwei Teilnehmeroperationen LEAVE und einer Teilnehmeroperation JOIN

Schlüsselbaumaktualisierungsphase

Anschließend werden, basierend auf den Knotenzuständen und in Abhängigkeit vom Betriebsmodus, die Hilfs- und Gruppenschlüssel erneuert. Die neuen Schlüssel werden dann den Teilnehmern der Gruppenkommunikation übermittelt. Der Algorithmus zur Aktualisierung des Schlüsselbaums wird erst formal vorgestellt und dann mit einem Beispiel für jeden Betriebsmodus veranschaulicht. Die Anzahl der JOINs wird hierbei mit J und die Anzahl der LEAVEs mit L bezeichnet. Außerdem werden die in Tabelle 2 und Tabelle 3 zusammengefassten Notationen verwendet.

Begonnen wird hierbei mit dem Betriebsmodus Key Agreement. In diesem Modus ermittelt der aktuelle TM auf der Basis der Knotenmarkierung, welcher Nutzer der nächste TM wird. Er übermittelt diese Information an die Gruppe und den neuen TM mit Hilfe der Nachricht des Typs `p3TMDistribute`. Zur Realisierung von Sammelverarbeitung im Modus Key Agreement wird vom neuen TM mit der Baumposition v_{ℓ_i,p_i} zur Erzeugung der Schlüsselwechselinformationen für die Nutzer der nachfolgende Algorithmus verwendet:

for each $w_{\ell_k,p_k} \in path(v_{\ell_i,p_i})$ do
  if w_{ℓ_k,p_k} != UNMOD then do
    if $w_{\ell_{k-1},p_{k-1}}$ = = $w_{\ell_k+1,2p_k}$ then do
      if $w_{\ell_k+1,2p_k+1}$ = = USER_NODE $\vee$ $w_{\ell_k+1,2p_k+1}$ = = UNMOD then do
        $b\tilde{k}_{\ell_k,p_k}$=BK($\tilde{k}_{\ell_k,p_k}$), $\tilde{k}_{\ell_k,p_k}$=DH($bk_{\ell_k+1,2p_k+1}$,$\tilde{k}_{\ell_k+1,2p_k}$), w_{ℓ_k,p_k} := UNMOD
    else do
      if $w_{\ell_k+1,2p_k}$ = = USER_NODE $\vee$ $w_{\ell_k+1,2p_k}$ = = UNMOD then do
        $b\tilde{k}_{\ell_k,p_k}$=BK($\tilde{k}_{\ell_k,p_k}$), $\tilde{k}_{\ell_k,p_k}$=DH($bk_{\ell_k+1,2p_k}$,$\tilde{k}_{\ell_k+1,2p_k+1}$), w_{ℓ_k,p_k} := UNMOD

Dieser ersetzt den in Abschnitt 4.4.1 Arbeitsschritt (3a) vorgestellten Algorithmus für die Schlüsselbaumoperation `GetUpdateKeys`. Ist der neue TM in der Lage, den Schlüsselbaum

vollständig zu aktualisieren, versendet dieser die Schlüsselwechselinformationen mit der Nachricht `p3UpdateDistribute`. Andernfalls versendet er die Nachricht `p3TMDistribute`. Diese wird dazu verwendet, einen weiteren TM zu benennen, der zur Vervollständigung des Schlüsselbaums beiträgt. Auch wenn mehrere TM zur Schlüsselbaumaktualisierung benötigt werden, muss der Markierungsalgorithmus nur einmal ausgeführt werden. In Abbildung 76 ist ein Beispiel zur Sammelverarbeitung dargestellt, bei dem zwei Nutzer eine Gruppe verlassen und ein neuer Nutzer beitritt, d.h. L=2, J=1. In dem Beispiel werden zunächst die beiden Nutzer gelöscht. Anschließend wird dem neuen Nutzer der gelöschte Knoten $v_{3,0}$ zugeordnet. Zur Erneuerung des Baums führt der aktuelle TM u_4 den Markierungsalgorithmus aus und erklärt den Nutzer u_1 durch die Nachricht `p3TMDistribute` zum neuen TM der Gruppe. Dieser berechnet durch eine zweimalige Anwendung des DH-Algorithmus die Schlüssel $\tilde{k}_{2,0}$, $\tilde{k}_{1,0}$. Da der Nutzer u_1 den Baum nicht vollständig berechnen kann, benennt er durch die Nachricht `p3TMDistribute` den Nutzer u_7 zum neuen TM. Hierbei verteilt er gleichzeitig die Blindschlüssel $b\tilde{k}_{2,0}$, $b\tilde{k}_{1,0}$. Der Nutzer u_8 vervollständigt die Baumberechnung und verteilt anschließend die Blindschlüssel $b\tilde{k}_{2,3}$ $b\tilde{k}_{1,1}$ mit der Nachricht `p3UpdateDistribute`. Nach dem Empfang dieser Nachricht kann jeder Nutzer den neuen Gruppenschlüssel $\tilde{k}_{0,0}$ berechnen. Der Nutzer u_3 berechnet den Gruppenschlüssel durch eine zweimalige Anwendung des DH-Algorithmus mit $\tilde{k}_{1,0}=DH(b\tilde{k}_{2,0},k_{2,1})$, $\tilde{k}_{0,0}=DH(b\tilde{k}_{1,1},\tilde{k}_{1,0})$.

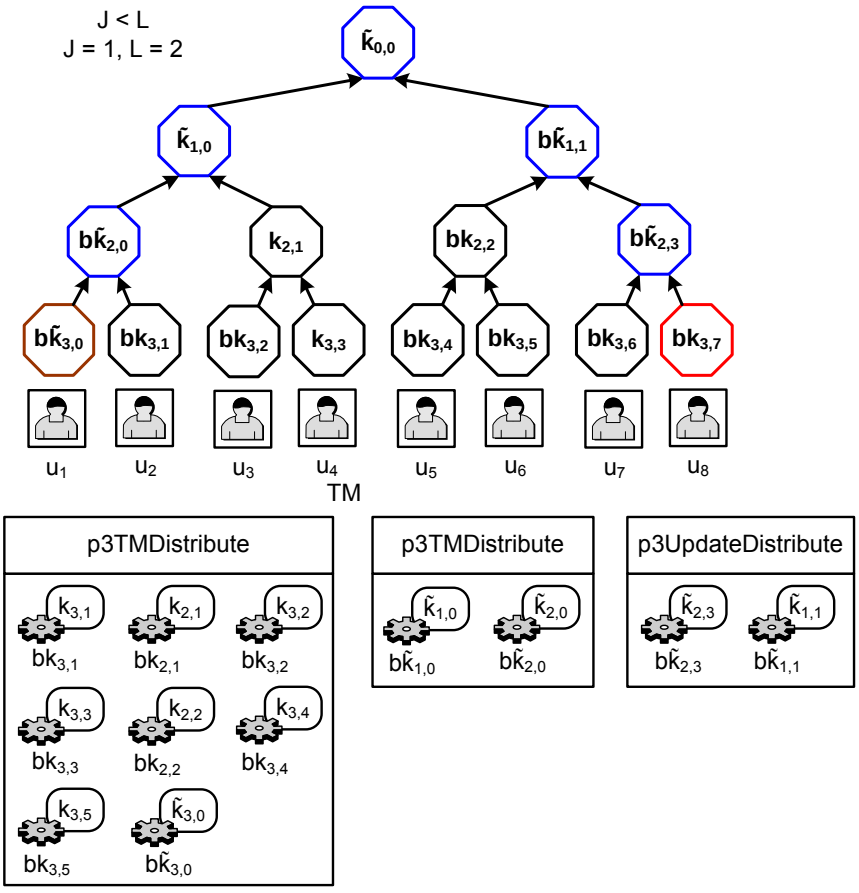

Abbildung 76: Sammelverarbeitung von zwei Teilnehmeroperationen LEAVE und einer Teilnehmeroperation JOIN im Modus Key Agreement

Zur Realisierung von Sammelverarbeitung im Modus Key Distribution wird zur Erzeugung der Schlüsselwechselinformationen für die Nutzer der nachfolgende Algorithmus verwendet:

for ℓ:=0 to h do
  for p:= 0 to $p_{max}(\ell)$ do
    if v_{ℓ_i,p_i} = =KEY_NODE & v_{ℓ_i,p_i} = = UPDATE1 then do
      for each $c_{\ell_k,p_k} \in$ child(v_{ℓ_i,p_i}) do
        $E(\tilde{k}_{\ell_i,p_i}, k_{\ell_k,p_k})$, v_{ℓ_i,p_i} := UNMOD
    if v_{ℓ_i,p_i} = = KEY_NODE & v_{ℓ_i,p_i} = =UPDATE2 then do
      $E(\tilde{k}_{\ell_i,p_i}, k_{\ell_i,p_i})$, v_{ℓ_i,p_i} := UNMOD
    if v_{ℓ_i,p_i} = =KEY_NODE & v_{ℓ_i,p_i} == NEW then do
      $E(\tilde{k}_{\ell_i,p_i}, k_{\ell_j,p_j})$, v_{ℓ_i,p_i} = UNMOD (k_{ℓ_j,p_j}= moved USER_LEAF key)
    if v_{ℓ_i,p_i} = =USER_NODE & v_{ℓ_i,p_i} = =NEW then do
      for each $w_{\ell_k,p_k} \in$ path(v_{ℓ_i,p_i}) do
        $E(\tilde{k}_{\ell_k,p_k}, k_{\ell_i,p_i})$, v_{ℓ_i,p_i} := UNMOD

Dieser ersetzt den in Abschnitt 4.4.2 Arbeitsschritt (2c) vorgestellten Algorithmus für die Schlüsselbaumoperation `GetUpdateKeys`. In Abbildung 77 ist ein Beispiel zur Sammelverarbeitung im Modus Key Distribution dargestellt. In dem abgebildeten Beispiel verlassen zwei Nutzer die Gruppe und ein neuer Nutzer tritt der Gruppe bei. Zunächst werden die beiden Nutzer gelöscht. Anschließend wird dem neuen Nutzer der gelöschte Knoten $v_{2,3}$ zugeordnet. Zur Erneuerung des Baums führt der GC den Markierungsalgorithmus aus und generiert die zufälligen Schlüssel $\tilde{k}_{1,1}$, $\tilde{k}_{1,2}$ und $\tilde{k}_{0,0}$. Den Nutzern wird der Schlüsselwechsel mit der Nachricht `p3UpdateDistribute` mitgeteilt. Auf Basis der Knotenmarkierung werden die verschlüsselten Schlüssel $E(\tilde{k}_{0,0},k_{1,0})$, $E(\tilde{k}_{0,0},\tilde{k}_{1,1})$, $E(\tilde{k}_{0,0},\tilde{k}_{1,2})$, $E(\tilde{k}_{1,1},\tilde{k}_{2,3})$, $E(\tilde{k}_{1,1},k_{2,4})$, $E(\tilde{k}_{1,1},k_{2,5})$, $E(\tilde{k}_{1,2},k_{2,6})$, $E(\tilde{k}_{1,2},k_{2,7})$ zur Nachricht hinzugefügt. Nach dem Empfang dieser Nachricht können nur die in der Gruppe befindlichen Nutzer den neuen Gruppenschlüssel $\tilde{k}_{0,0}$ berechnen. Der Nutzer u_1 berechnet den Gruppenschlüssel mit $\tilde{k}_{0,0}=D(E(\tilde{k}_{0,0},k_{1,0}),k_{1,0})$.

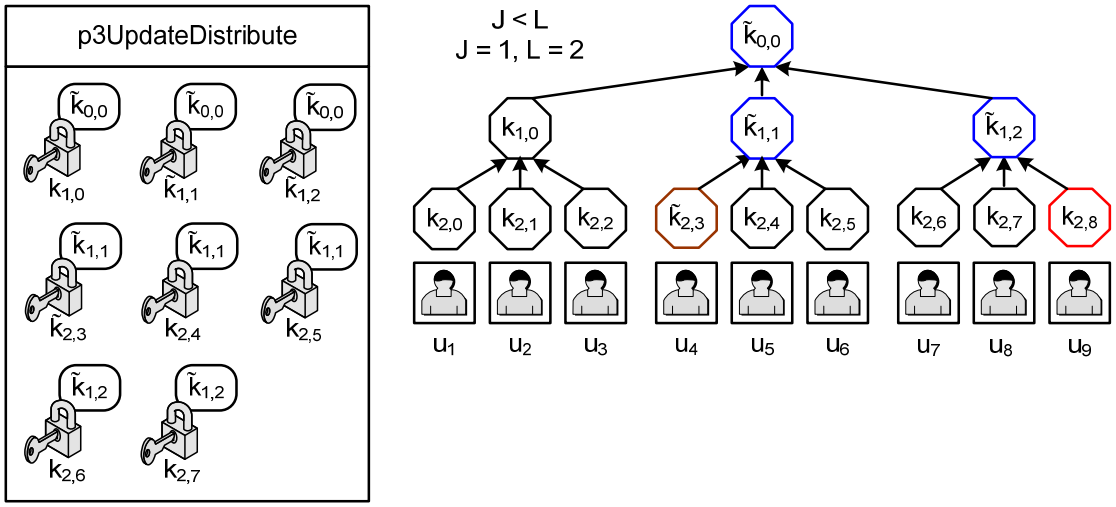

Abbildung 77: Sammelverarbeitung von zwei Teilnehmeroperationen LEAVE und einer Teilnehmeroperation JOIN im Modus Key Distribution

7.3.2 Theoretische Abschätzung der Effizienz

In diesem Abschnitt soll der Effizienzgewinn sowohl für den Betriebsmodus Key Agreement als auch Key Distribution im ungünstigsten Fall betrachtet werden. Der ungünstigste Fall liegt vor, wenn nur Nutzer die Gruppe verlassen und diese gleichmäßig über die Blätter des Schlüsselbaums verteilt sind. In diesem Fall ist ein Austausch von Knoten im Baum nicht möglich und sich überschneidende Pfade zur Wurzel sind am geringsten. Zur Abschätzung betrachten wir ein Schlüsselmanagementsystem, das einen vollständigen Baum vom Grad d mit $U=d^h$ Nutzern einsetzt. Diesen verlassen $L=d^\ell$ Nutzer ($L \leq U/d$), die gleichmäßig über den Schlüsselbaum verteilt sind.

Zunächst wird für den Betriebsmodus Key Agreement der Effizienzgewinn bei der Anzahl der kryptographischen Operationen, d.h. der Exponentiationen, des TMs bei der Sammelverarbeitung von Teilnehmeroperationen im ungünstigen Fall berechnet. Grundlage dieser Berechnung ist, dass zwei Exponentiationen pro Knoten des Schlüsselbaums notwendig sind. Verlassen $L=d^\ell$ Nutzer den Schlüsselbaum und werden diese Teilnehmeroperationen zusammen verarbeitet, dann müssen alle Knoten der Ebenen 0 bis ℓ aktualisiert werden, d.h. es müssen $2 \cdot (2^{\ell+1} - 1) - 1$ Exponentiationen zur Berechnung von Schlüssel und Blindschlüssel durchgeführt werden (vgl. Term A). In den Ebene ℓ+1 bis h-2 müssen für 2^ℓ Konten Schlüssel und Blindschlüssel berechnet werden (vgl. Term B). In der Ebene h-1 müssen für 2^ℓ Knoten aus den bereits vorhandenen Schlüsseln die Blindschlüssel generiert werden (vgl. Term C).

$$O_b = \underbrace{2 \cdot (2^{\ell+1} - 1) - 1}_{\text{Term A}} + \underbrace{2 \cdot 2^\ell \cdot (h - 2 - \ell)}_{\text{Term B}} + \underbrace{2^\ell}_{\text{Term C}}$$

$$O_b = L + 2 \cdot L \cdot Log_2 \frac{U}{L} - 3$$

Wird bei jedem Gruppenaustritt der L Nutzer jeweils ein Schlüsselwechsel durchgeführt, dann gilt für die gesamte Anzahl der durchgeführten kryptographischen Operationen bei der Einzelverarbeitung O_s:

$$O_s = L \cdot 2 \cdot (h - 1) \Rightarrow$$
$$O_s = 2 \cdot L \cdot (Log_2 U - 1)$$

Der Effizienzgewinn bezüglich der kryptographischen Operationen ΔO_A im Modus Key Agreement kann damit wie folgt errechnet werden:

$$\Delta O_A = O_s - O_b \Rightarrow$$
$$\Delta O_A = 2 \cdot L \cdot (h - 1) - 2 \cdot (2L - 1) + 1 - 2 \cdot L \cdot (h - 2 - \ell) - L \Rightarrow$$
$$\Delta O_A = L \cdot (2 \cdot Log_2 L - 3) + 3$$

Nachdem die Effizienzsteigerung des Modus Key Agreement analysiert wurde, wird nun die Effizienzsteigerung durch Sammelverarbeitung für den Modus Key Distribution bezüglich der Anzahl der vom GC durchgeführten kryptographischen Operation, d.h. der Verschlüsselungen, im ungünstigen Fall abgeschätzt. Für die Berechnung wird zusätzlich zu den zu Beginn des Abschnitts gemachten Annahmen für den Schlüsselbaumgrad d≥3

angenommen. Verlassen $L=d^{\ell}$ Nutzer den Schlüsselbaum und werden diese zusammen verarbeitet, dann müssen alle Knoten der Ebene 0 bis ℓ, d.h. $\frac{d^{\ell+1}-1}{d-1}$ Knoten mit d Kindern verschlüsselt werden (vgl. Term A). In den Ebenen ℓ+1 bis h-2 müssen d^{ℓ} Knoten mit d Kindern (vgl. Term B) und in der Ebene h-1 müssen d^{ℓ} Knoten mit d-1 Kindern verschlüsselt werden (vgl. Term C). Die gesamte Anzahl der durchgeführten kryptographischen Operationen O_b bei der Sammelverarbeitung kann berechnet werden mit:

$$O_b = \underbrace{\frac{d^{\ell+1}-1}{d-1}\cdot d}_{\text{Term A}} + \underbrace{d^{\ell}\cdot(h-2-l)\cdot d}_{\text{Term B}} + \underbrace{d^{\ell}\cdot(d-1)}_{Term\ C} \Rightarrow$$

$$O_b = d^{\ell}\cdot d\cdot(h-\ell) + \frac{d^{\ell}-d}{d-1} \Rightarrow$$

$$O_b = L\cdot d\cdot Log_d\frac{U}{L} + \frac{L-d}{d-1}$$

Wird bei jedem Austritt der L Nutzer jeweils ein Schlüsselwechsel durchgeführt, dann gilt für die gesamte Anzahl der durchgeführten kryptographischen Operationen bei der Einzelverarbeitung O_s:

$$O_s = L\cdot(d\cdot h-1) \Rightarrow$$

$$O_s = L\cdot(d\cdot Log_d U-1)$$

Im Modus Key Distribution kann der Effizienzgewinn bei der Anzahl der durchgeführten kryptographischen Operationen ΔO_D damit wie folgt errechnet werden:

$$\Delta O_D = O_s - O_b \Rightarrow$$

$$\Delta O_D = L\cdot(d\cdot h-1) - L\cdot d\cdot(h-l) - \frac{L-d}{d-1} \Rightarrow$$

$$\Delta O_D = L\cdot(d\cdot Log_d L - \frac{1-\frac{1}{L}}{1-\frac{1}{d}}) \approx L\cdot(d\cdot Log_d L - \frac{1}{1-\frac{1}{d}})$$

Im Betriebsmodus Key Distribution ist die Anzahl der vom GC durchgeführten kryptographischen Operationen identisch mit der Anzahl der ausgetauschten Schlüsselpakete.

Die theoretische Analyse zeigt, dass im ungünstigen Fall die Komplexität der eingesparten kryptographischen Operationen bei beiden Betriebsmodi $O(L\cdot Log_d L)$ ist.

7.3.3 Ergebnisse der Effizienzmessung

Zur Messung der Effizienz der Sammelverarbeitung wird der in Abschnitt 7.2 vorgestellte Messaufbau eingesetzt. Weiterhin werden die in Abschnitt 6.2 definierten einfachen Metriken, d.h. die Anzahl der ausgetauschten Schlüsselpakete, der übertragenen Nachrichten und der durchgeführten kryptographischen Operationen verwendet. Begonnen wird die Analyse der Sammelverarbeitungseffizienz mit dem synthetischen Nutzerverhalten. Hierbei wurde bei einer Gruppengröße von 60 Nutzern im Modus Key Agreement bzw. 140 Nutzern im Modus Key Distribution die Anzahl der Teilnehmeroperationen JOIN und LEAVE sukzessive bis zu einem Anteil von 10 % der Gruppengröße erhöht. Die bei der Messung

gewonnenen Ergebnisse spiegeln den günstigen Fall wider, da die Teilnehmeroperationen in einem zusammenhängenden Teil des Baums durchgeführt werden. Der ungünstige Fall wurde in Abschnitt 7.3.2 betrachtet. Dadurch überschneiden sich die Pfade auf dem Weg zur Wurzel öfter als bei einer Gleichverteilung der Nutzer im Baum. Für den Betriebsmodus Key Agreement ist das Ergebnis der Messung in Abbildung 78 dargestellt. Bei gleicher Anzahl der Teilnehmeroperationen JOIN und LEAVE ist bei allen Messgrößen die Einsparung am größten. Im Modus Key Agreement wird durch Sammelverarbeitung die Anzahl der übertragenen Schlüsselpakete auf die im Schlüsselbaum enthaltene Knotenzahl reduziert. Weiterhin kann die Zahl der ausgetauschten Nachrichten bei J JOINs und L LEAVEs auf den Wert $\max(J, L) + 1$ reduziert werden.

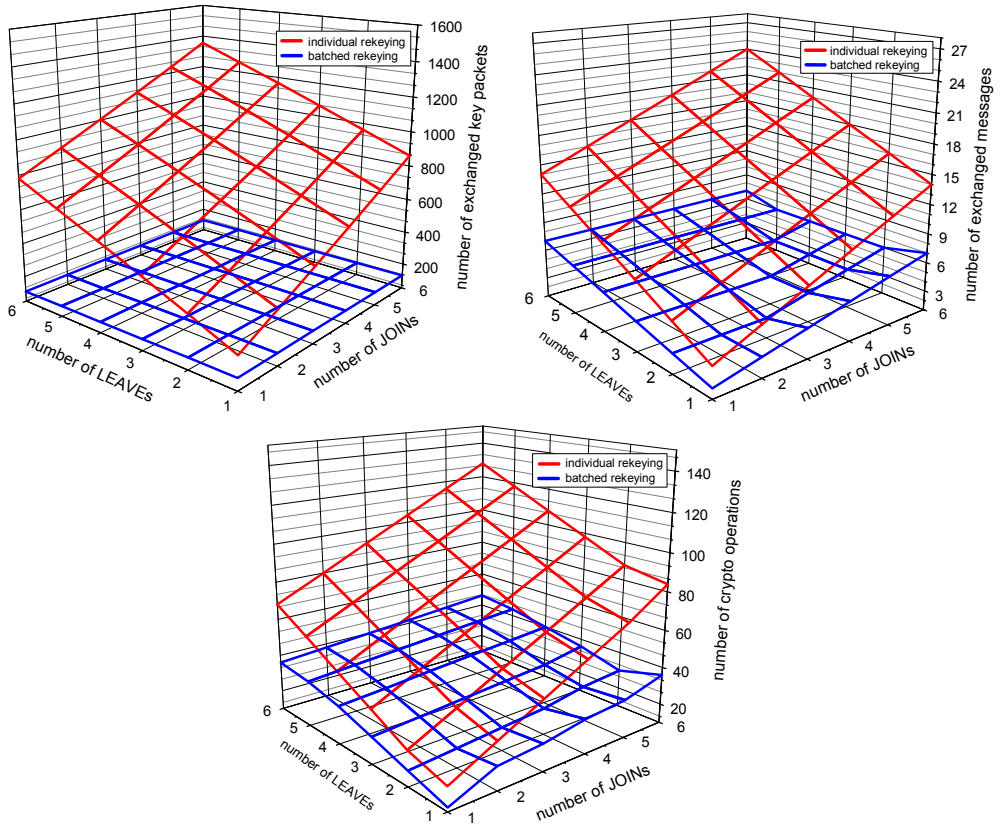

Abbildung 78: Anzahl der ausgetauschten Schlüsselpakete (links oben), der übertragenen Nachrichten (rechts oben) und der kryptographischen Operationen (unten) bei Einzel- und Sammelverarbeitung im Modus Key Agreement

Auch im Betriebsmodus Key Distribution ist bei gleicher Anzahl der Teilnehmeroperationen JOIN und LEAVE in allen Messgrößen die Einsparung am größten (Abbildung 79). Bei diesem Modus wird durch die Sammelverarbeitung die Anzahl der beim Schlüsselwechsel übertragenen Nachrichten auf eine Nachricht reduziert.

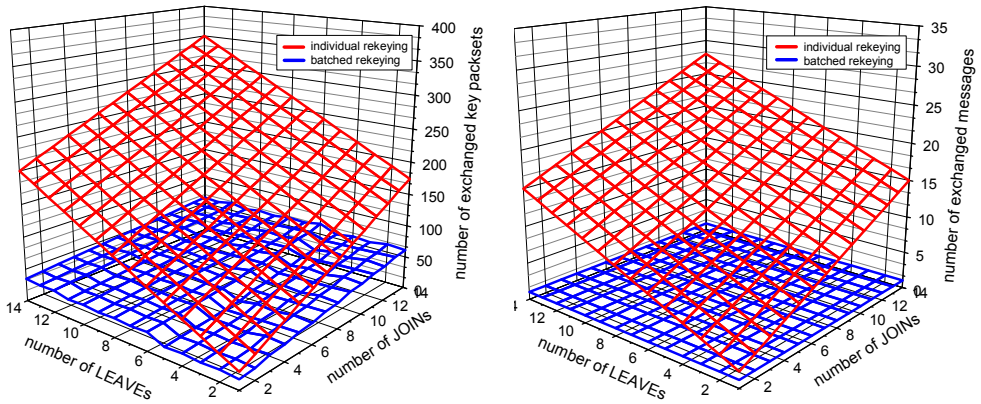

Abbildung 79: Anzahl der ausgetauschten Schlüsselpakete (links) und der übertragenen Nachrichten (rechts) bei Einzel- und Sammelverarbeitung im Modus Key Distribution

Um die Effizienz der Sammelverarbeitung bei realen Bedingungen zu analysieren, wurde MIKE mit dem in Abschnitt 6.3 modellierten zivilen und militärischen Nutzerverhalten konfrontiert und die Nutzeranfragen zum Gruppenbeitritt und –austritt sowohl mit Einzel- als auch mit Sammelverarbeitung verarbeitet. Hierbei wurde die Dauer der Sammelphase von 60s bis 960s variiert und die Anzahl der ausgetauschten Schlüsselpakete, übertragenen Nachrichten sowie die vom TM/GC durchgeführten kryptographischen Operationen gemessen. In beiden Betriebsmodi bewirkt die Sammelverarbeitung eine Verminderung der Extremwerte bei Anzahl der übertragenen Schlüsselpakete, der ausgetauschten Nachrichten und der durchgeführten kryptographischen Operationen. In Abbildung 80 sind hierzu exemplarisch zwei Messresultate dargestellt. Weiterhin ist in der Grafik erkennbar, dass bei dem untersuchten militärischen Nutzerverhalten die Anzahl der Schlüsselwechsel viel geringer ist.

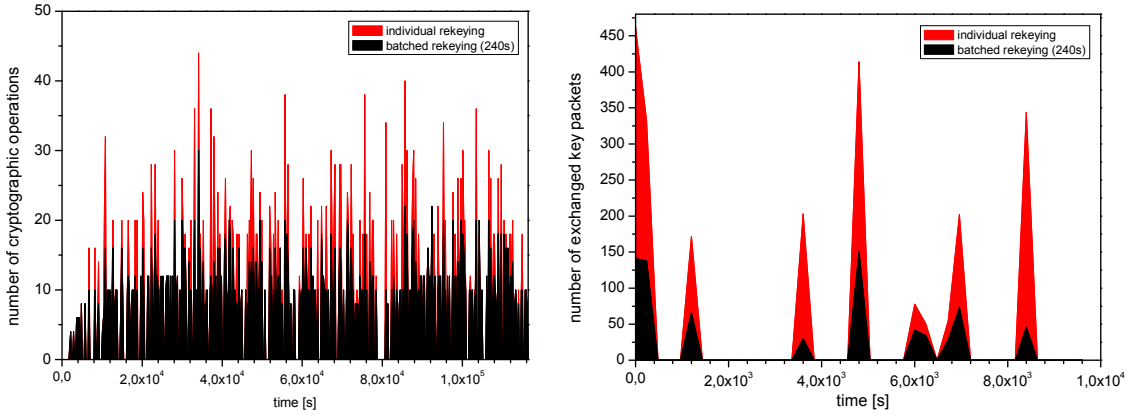

Abbildung 80: Anzahl der kryptographischen Operationen im Modus Key Agreement beim zivilen Nutzerverhalten Nr. 1 (links) und Anzahl der ausgetauschten Schlüsselpakete im Modus Key Distribution beim militärischen Nutzerverhalten Nr. 2 (rechts)

Zur weiteren Untersuchung wurde die Summe der ausgetauschten Schlüsselpakete, die übertragenen Nachrichten sowie der vom TM/GC durchgeführten kryptographischen Operationen während der gesamten Dauer des Nutzerverhaltens sowohl für Sammel- als auch für Einzelverarbeitung ermittelt. Anschließend wurden aus diesen Werten der Anteil der bezüglich der Einzelverarbeitung eingesparten Schlüsselpakete, Nachrichten und

kryptographischen Operationen errechnet. Für den Betriebsmodus Key Agreement ist das Ergebnis in Abbildung 81 dargestellt. Während bei dem untersuchten zivilen Nutzerverhalten der Effizienzgewinn maßgeblich durch die Länge der Sammelperiode bestimmt wird, kann bei dem militärischen Nutzerverhalten schon mit einer kurzen Sammelphase ein erheblicher Effizienzgewinn erzielt werden. Im Modus Key Agreement ist bei beiden Arten des Nutzerverhaltens der Effizienzgewinn bezüglich der ausgetauschten Schlüsselpakete am größten.

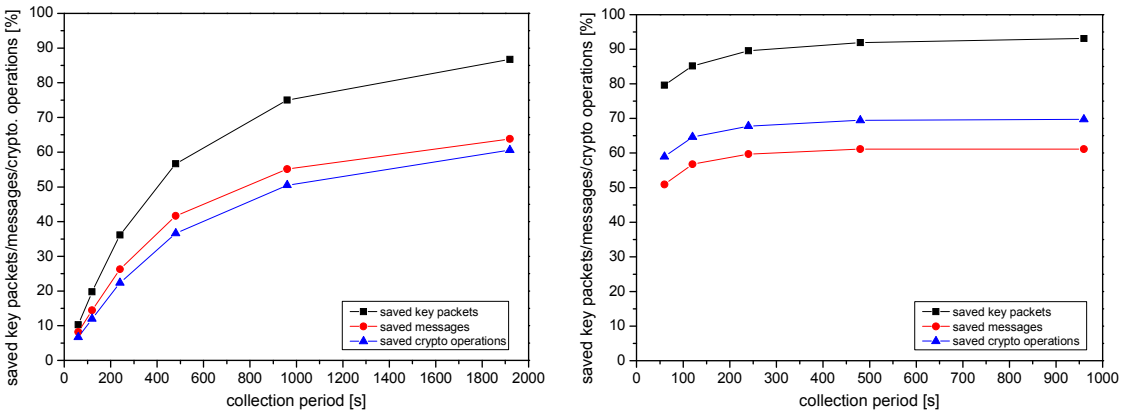

Abbildung 81: Effizienzgewinn durch Einsatz der Sammelverarbeitung bei zivilem (links) und militärischem (rechts) Nutzerhalten Nr. 1 im Modus Key Agreement

Auch im Modus Key Distribution kann bei dem untersuchten militärischen Nutzerverhalten durch den Einsatz der Sammelverarbeitung schon mit kurzen Sammelphasen ein erheblicher Effizienzgewinn erzielt werden (Abbildung 82). In diesem Betriebsmodus kann bei der Anzahl der übertragenen Nachrichten die größte Verbesserung erzielt werden. Die durchgeführten Analysen zeigen, dass durch die Sammelverarbeitung von Nutzeranfragen die in Abschnitt 3.2 geforderten effizienten Schlüsselwechsel bei Umgliederungen von Truppenteilen ermöglicht wird. Generell kann durch die Sammelverarbeitung ein Effizienzgewinn mit geringer Join Latency bzw. Leave Bonus erzielt werden, wenn das Nutzerverhalten eine Vielzahl von Teilnehmeroperationen in kurzer Zeit, im Folgenden als Burst-Teilnehmeroperationen bezeichnet, aufweist.

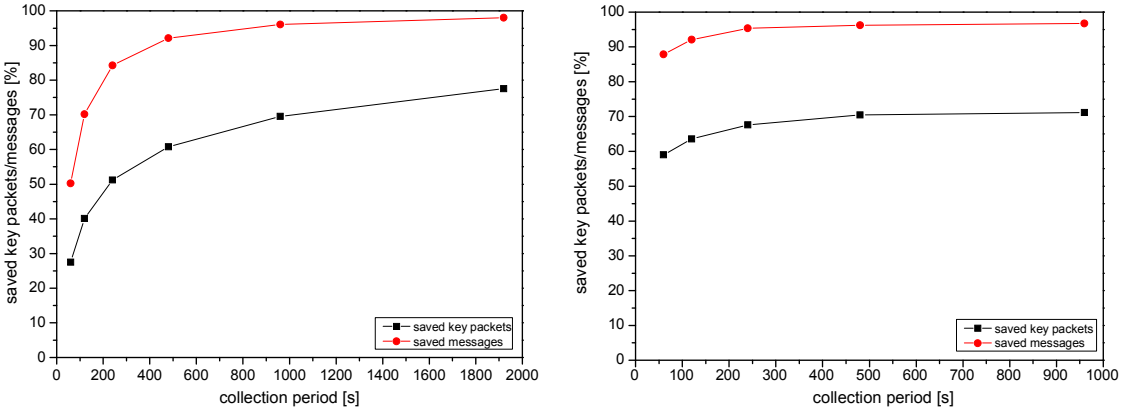

Abbildung 82: Effizienzgewinn durch Sammelverarbeitung bei zivilem (links) und militärischem (rechts) Nutzerverhalten Nr. 2 im Modus Key Distribution

7.4 Nutzerverhaltenbasierte Schlüsselbaumkonstruktion

In baumbasierten Schlüsselmanagementsystemen kann eine Effizienzsteigerung erzielt werden, indem sehr dynamische Nutzer, d.h. Nutzer mit einer größeren Wahrscheinlichkeit, die Gruppe zu verlassen, näher an der Wurzel des Schlüsselbaums platziert werden. Der durch diese Maßnahme verringerte Abstand zur Wurzel bewirkt einen geringeren Aufwand beim Schlüsselwechsel. Dieser geringere Aufwand besteht aus einer verringerten Anzahl an durchgeführten kryptographischen Operationen. Im Modus Key Distribution hat die geringe Anzahl an Verschlüsselungen durch den GC auf Grund der geringen benötigten Rechenleistung für die Durchführung einer Verschlüsselung nur unerheblichen Einfluss auf die Effizienz. Allerdings bewirkt die in diesem Modus an die kryptographischen Operationen gekoppelte Anzahl der ausgetauschten Schlüsselpakete eine Effizienzsteigerung. Für den optimalen Schlüsselbaumaufbau wurde in [Poo01] das nachfolgende Theorem aufgestellt:

- In einer Gruppe $\{u_1,\ldots,u_U\}$, bei der ein Nutzer u_i die Wahrscheinlichkeit p_i besitzt, die Gruppe zu verlassen, ist die optimale mittlere Anzahl an zugewiesenen Hilfsschlüsseln gegeben durch die Entropie E_d:

$$E_d = \sum_{i=1}^{U} p_i \cdot \ell_i$$

Hierbei bezeichnet ℓ_i die Pfadlänge des Nutzers u_i zur Schlüsselbaumwurzel. Zusammen mit dem Wurzelschlüssel ist damit die optimale mittlere Schlüsselanzahl pro Nutzer E_d+1. Die optimale Lösung des Problems der mittleren Pfadlängenminimierung des Schlüsselbaums kann durch Huffman-Bäume erzielt werden. Für den Spezialfall, dass in einer Gruppe alle Nutzer die gleiche Wahrscheinlichkeit p_i haben, ist der Einsatz eines balancierten Schlüsselbaums die optimale Lösung. Ausgangspunkt bei der Entwicklung des Theorems in [Poo01] ist die baumbasierte zentrale Gruppenschlüsselbereitstellung. In dem Beweis des Theorems werden jedoch keine Annahmen bezüglich der Schlüsselerzeugung beim Gruppenschlüsselwechsel gemacht. Aus diesem Grund wird das Theorem auch im Modus Key Agreement angewandt.

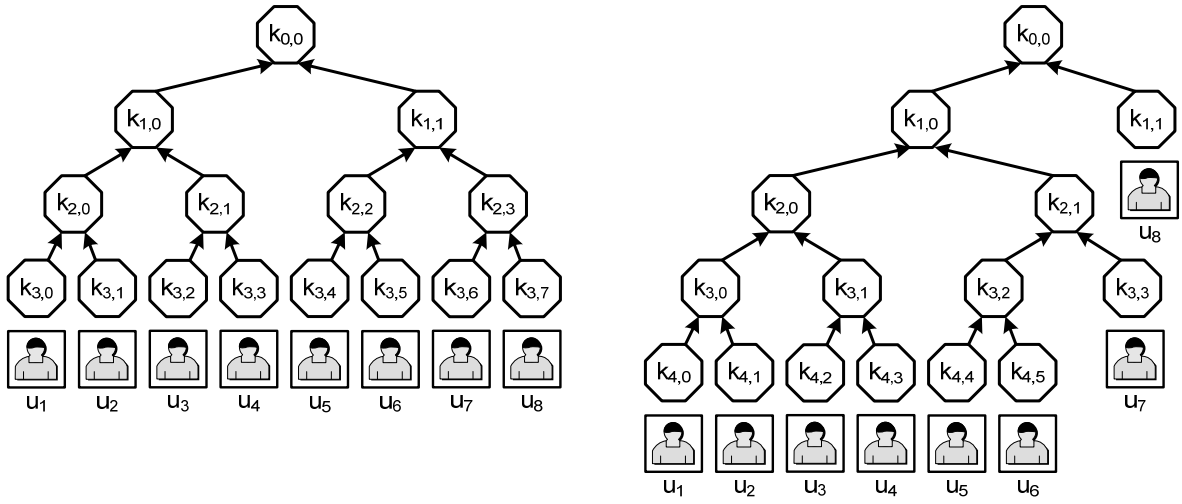

Abbildung 83: Schlüsselbaum ohne (links) und mit (rechts) nutzerverhaltenbasierter Struktur

Bevor die Effizienz der nutzverhaltenbasierten Schlüsselbaumkonstruktion genauer analysiert wird, wird die Optimierung anhand eines extremen Beispiels veranschaulicht. Hierzu werden die Kosten für den Gruppenbeitritt und den anschließenden Austritt des Nutzers u_8 in einer Gruppe mit sieben Teilnehmern betrachtet. Den Nutzern $\{u_1,\ldots,u_7\}$ ist die Austrittswahrscheinlichkeit null und dem Nutzer u_8 die Austrittswahrscheinlichkeit eins zugeordnet. Der zur Schlüsselverwaltung eingesetzte Baum ist einmal balanciert und einmal nutzerverhaltenbasiert aufgebaut (Abbildung 83). Der Überblick über die entstehenden Kosten für die Teilnehmeroperationen des Nutzers u_8 in Tabelle 22 zeigt die Effizienz der nutzerverhaltenbasierten Baumkonstruktion.

Verfahren	Baumstruktur	max. Anzahl der kryptographischen Operationen	Anzahl der übertragenen Schlüsselpakete
MIKE – Key Agreement	balanciert	3	12
MIKE – Key Agreement	nutzerverhaltenbasiert	0	12
MIKE – Key Distribution	balanciert	4	4
MIKE – Key Distribution	nutzerverhaltenbasiert	2	2

Tabelle 22: Kosten in einem balancierten und nutzerverhaltenbasierten Schlüsselbaum

7.4.1 Algorithmen zur nutzerverhaltenbasierten Schlüsselbaumkonstruktion

Soll das im vorherigen Abschnitt vorgestellte Theorem praktisch eingesetzt werden, so sind die nachfolgenden Herausforderungen zu bewältigen:

- Bestimmung der Wahrscheinlichkeitsfunktion des Gruppenaustritts für vorhandene und zukünftige Teilnehmer der Gruppenkommunikation.
- Anpassung der Baumstruktur an die mit der Wahrscheinlichkeitsfunktion ermittelten Werte. Diese Anpassung erfordert eine Veränderung der Position der Nutzer im Schlüsselbaum und damit zusätzliche Schlüsselwechsel.

Zur Umsetzung des Ziels, dynamische Nutzer näher an der Schlüsselbaumwurzel zu platzieren, wurden zusätzlich zum Einfügepunktauswahlalgorithmus Balance-Insert zwei weitere Algorithmen entworfen. Der Algorithmus Balance-Insert ist das Standardverfahren zur Auswahl des Einfügepunktes. Bei diesem werden neue Knoten mit dem Ziel eingefügt, den Baum zu balancieren. Soweit nicht explizit erwähnt, wird der Algorithmus Balance-Insert in den übrigen Kapiteln bei der Bewertung der Effizienz eingesetzt. Der Algorithmus 2-Partition-Insert verfolgt das Ziel, dynamische Nutzer ohne Kenntnis der Wahrscheinlichkeitsfunktion des Gruppenaustritts näher an der Wurzel des Schlüsselbaums einzufügen. Der Algorithmus Entropy-Insert sucht einen Einfügepunkt durch Anwendung des Theorems zur Entropie des Schlüsselbaums.

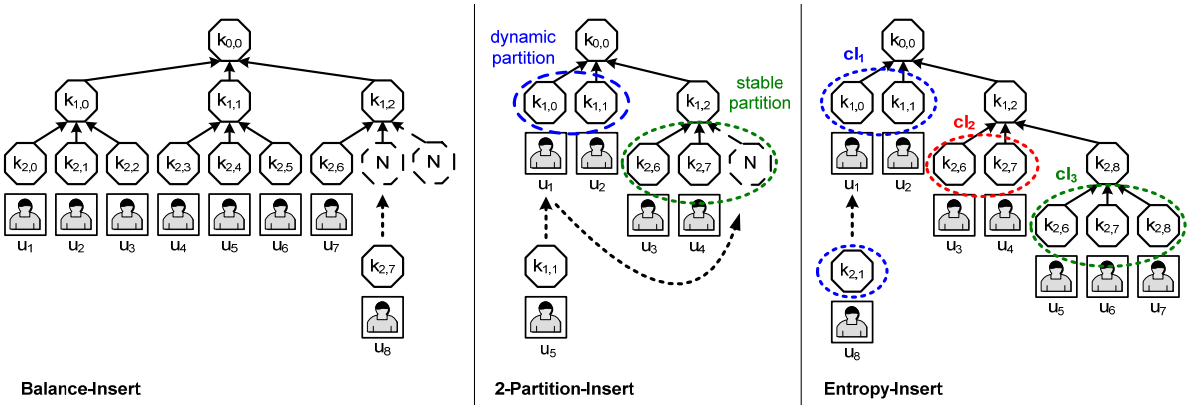

Abbildung 84: Einfügepunktauswahlalgorithmen

Im Folgenden werden die in Abbildung 84 dargestellten drei Algorithmen zur Einfügepunktauswahl genauer erläutert:

Balance-Insert

Der Algorithmus nimmt an, dass alle Nutzer den gleichen Optimierungsfaktor besitzen. Wie in [Poo01] gezeigt wurde, ist in diesem Fall ein balancierter Baum die optimale Datenstruktur zur Verwaltung des Gruppenschlüssels. Es ist deshalb Ziel des Algorithmus, die Differenz zwischen maximaler und minimaler Pfadlänge terminaler Knoten zur Wurzel zu minimieren. Zur Ausführung des Algorithmus müssen daher weder die Aufteilung der Nutzer in Cluster noch die Optimierungsfaktoren bekannt sein. Wird dieser Algorithmus in Kombination mit der Sammelverarbeitung von Teilnehmeroperationen eingesetzt, werden gelöschte Knoten durch neue Knoten ersetzt.

2-Partition-Insert

Für den 2-Partition-Insert-Algorithmus wird der Schlüsselbaum in zwei Partitionen unterteilt. Diese werden als dynamische und stabile Partition bezeichnet. Nutzer werden beim Gruppenbeitritt zunächst in die dynamische Partition eingefügt. Parameter dieses Einfügepunktauswahlalgorithmus ist die Größe der dynamischen Partition $S_{d\text{-}Part}$. Ist die dynamische Partition bereits vollständig besetzt und soll dennoch ein weiterer Nutzer in den Schlüsselbaum eingefügt werden, wird der Nutzer, der sich am längsten in der dynamischen Partition befindet, in die stabile Partition verschoben. Der neue Nutzer wird anschließend in die entstehende Lücke in der dynamischen Partition eingefügt. Vorbild für diesen Algorithmus bildet das zweistufige Speichersystem in Rechnern. Bei einem derartigen System kann der langsame Speicher eine große Anzahl von Speicherseiten aufnehmen, während der schnelle Speicher (Cache) nur eine k-elementige Teilmenge von Speicherseiten aufnehmen kann. Zur Ausführung des Algorithmus müssen weder die Aufteilung der Nutzer in Teilmengen noch die Optimierungsfaktoren bekannt sein. Wird der beschriebene Algorithmus in Kombination mit der Sammelverarbeitung von Teilnehmeroperationen eingesetzt, so werden nur innerhalb einer Partition gelöschte Knoten durch neue Knoten ersetzt.

Entropy-Insert

Die Idee des Algorithmus Entropy-Insert besteht darin, beim Einfügen eines neuen Nutzers die Wahrscheinlichkeit seines Gruppenaustritts zu berücksichtigen. Hierzu berechnet der Algorithmus für jeden Knoten j die Kostenfunktion *K(j)*. Es wird derjenige Knoten als Einfügepunkt gewählt, bei dem das Ergebnis der Kostenfunktion minimal ist. Zur

Bestimmung der Wahrscheinlichkeit, dass ein Nutzer die Gruppe verlässt, werden nachfolgende Vereinfachungen durchgeführt:

- Jedem Nutzer u_i wird ein zeitunabhängiger normierter Optimierungsfaktor o_i zugewiesen. Diesen wertet der Einfügepunktauswahlalgorithmus bei der Ermittlung des geeigneten Einfügepunkts aus.
- Die Teilnehmer der Gruppenkommunikation $\{u_1,\ldots,u_U\}$ mit der Teilnahmedauer t mit $t_1(cl_k)<t\leq t_2(cl_k)$ werden zu Clustern $cl_k=\{u_{1_k},\ldots,u_{i_k}\}$ zusammengefasst. Die Zeiten $t_1(cl_k)$ und $t_2(cl_k)$ sind die obere und untere Schranke für die Teilnahmedauer der Nutzer. Zur eindeutigen Bezeichnung eines Clusters wird diesem die Nummer cId_k zugeordnet. Jedem dem Cluster cl_k zugeordneten Nutzer wird der gleiche Optimierungsfaktor zugewiesen.
- Der Optimierungsfaktor o_i eines Clusters cId_i wird als das normierte Inverse der mittleren Teilnahmedauer festgelegt.

Das Zusammenfassen von Nutzern zu Clustern und die Zuweisung eines Optimierungsfaktors sind im zivilen Einsatzbereich nur schwer möglich, da in diesem Bereich das Nutzerverhalten meistens unkoordiniert ist. Ist die Teilnahme an einer sicheren Gruppenkommunikation Inhalt einer vertraglich vereinbarten Leistung, bieten Vertragslaufzeiten eventuell eine Möglichkeit zur Festlegung von Clustern und Optimierungsfaktoren. Fraglich ist, ob in einem derartigen Einsatzbereich die durch die nutzerverhaltenbasierte Schlüsselbaumkonstruktion erzielte Effizienzsteigerung notwendig ist. In einem militärischen Einsatzbereich ergibt sich die Zusammenfassung von Nutzern zu Clustern aus der organisatorischen Struktur. Eine Ermittlung der Optimierungsfaktoren ist möglich, da sich die Nutzer koordiniert und nach einem vorher festgelegten Muster verhalten. Beispielsweise ist es bei dem in Abschnitt 6.3 vorgestellten militärischen Nutzerverhalten möglich, den Teilnehmern der Fahrzeugpatrouille einen Optimierungsfaktor zuzuordnen, da ihre Teilnahmedauer an der Gruppe bekannt ist.

Zunächst wird die für den Algorithmus Entropy-Insert benötigte Kostenfunktion $K_D(j)$ für den Betriebsmodus Key Distribution aufgestellt. Ausgangspunkt der Herleitung ist die Kostenfunktion $K''_D(j)$. Mittels $K''_D(j)$ werden die Kosten für den Austritt aller im Schlüsselbaum befindlichen Nutzer berechnet, falls der neue Nutzer u_{U+1} an die Position j eingefügt wird. Die Kosten ergeben sich aus den Kosten für den Austritt der U Teilnehmer der Gruppenkommunikation (Term A″), den Kosten für den Austritt des neuen Nutzers u_{U+1} (Term B″) abzüglich der durch eine Sammelverarbeitung gesparten Kosten (Term C″). In Term C″ bezeichnet n_j^{save} die Anzahl der eingesparten Knotenaktualisierungen durch sich bei der Sammelverarbeitung überlagernde Pfade zur Wurzel. $K''_D(j)$ kann zu $K'_D(j)$ umgeformt werden, indem die Kosten für den Austritt der U Teilnehmer aufgeteilt werden in die bereits bestehenden Kosten für den Austritt (Term A1') sowie der zusätzlichen Austrittskosten durch das Hinzufügen eines neuen Nutzers (Term A2'). Diese zusätzlichen Kosten (Term A2') entstehen durch die Verschiebung von U′ Teilnehmern auf eine andere Ebene des Schlüsselbaums. Die eingesetzte Kostenfunktion $K_D(j)$ ergibt sich aus der Kostenfunktion $K'_D(j)$, wenn man berücksichtigt, dass der Term A1' konstant ist und beim Einfügen von Knoten bereits im Schlüsselbaum befindliche Nutzer maximal um eine Ebene, d.h. $\tilde{\ell}=1$, verschoben werden.

$$K''_D(j) = \underbrace{d \cdot \sum_{i=1}^{U} \ell_i \cdot o_i}_{\text{Term A''}} + \underbrace{d \cdot \ell_j \cdot o_{U+1}}_{\text{Term B''}} - \underbrace{n_j^{save} \cdot o_{U+1}}_{\text{Term C''}} \Rightarrow$$

$$K'_D(j) = \underbrace{\sum_{i=1}^{U} \ell_i \cdot o_i}_{\text{Term A1'}} + \underbrace{\sum_{i=1}^{U'} \tilde{\ell}_i \cdot o_i}_{\text{Term A2'}} + \underbrace{\ell_j \cdot o_{U+1}}_{\text{Term B'}} - \underbrace{\tfrac{1}{d} \cdot n_j^{save} \cdot o_{U+1}}_{\text{Term C'}} \Rightarrow$$

$$K_D(j) = \underbrace{\sum_{i=1}^{U'} o_i}_{\text{Term A}} + \underbrace{\ell_j \cdot o_{U+1}}_{\text{Term B}} - \underbrace{\tfrac{1}{d} \cdot n_j^{save} \cdot o_{U+1}}_{\text{Term C}}$$

Die Analyse des in [Sel02] vorgeschlagenen Algorithmus zur Einfügepunktauswahl zeigt, dass dieser nicht den Term C berücksichtigt. Dieser Term bewertet die eingesparten Kosten, wenn bei der Sammelverarbeitung gelöschte Knoten durch neue Knoten ersetzt werden können. Da im Modus Key Distribution der Aufwand für den Gruppenaustritt größer als für den Gruppenbeitritt ist, wird in der Kostenfunktion $K_D(j)$ der Effizienzgewinn durch die Ersetzung von gelöschten mit neuen Knoten (Term C) nur mit dem Faktor $\frac{1}{d}$ berücksichtigt. Die Bedeutung von Term C in der Kostenfunktion kann veranschaulicht werden, wenn die Kosten für den Einfügepunkt bei der Sammelverarbeitung der Teilnehmeroperationen LEAVE und JOIN in einer Gruppe mit acht Teilnehmern betrachtet werden. Bei dieser Berechnung wird jedem Teilnehmer u_i der Optimierungsfaktor o_i=0,1 zugewiesen. Für die in Abbildung 85 dargestellten potentiellen Einfügepunkte In_1 und In_2 berechnet die Kostenfunktion $K_D(j)$ ohne Term C die Kosten $K_D(In_1)$=$K_D(In_2)$=0,2. Werden die beiden Teilnehmeroperationen gemeinsam verarbeitet, ist der Einfügepunkt In_1 aber effizienter.

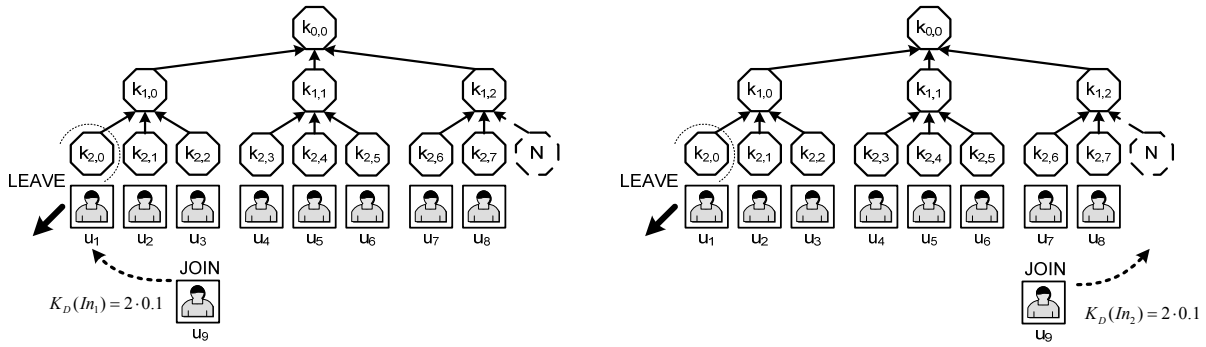

Abbildung 85: Mögliche Einfügepunkte In_1 (links) und In_2 (rechts) des neuen Nutzers bei der Sammelverarbeitung der Teilnehmeroperationen LEAVE und JOIN

Für den Betriebsmodus Key Agreement wird die Kostenfunktion $K_A(j)$ für den Algorithmus Entropy-Insert in analoger Weise hergeleitet. Term C berücksichtigt in diesem Modus den Effizienzgewinn, wenn bei der Sammelverarbeitung gelöschte Knoten durch neue Knoten ersetzt werden.

$$K''_A(j) = \underbrace{2 \cdot \sum_{i=1}^{U} \ell_i \cdot o_i}_{\text{Term A''}} + \underbrace{2 \cdot \ell_j \cdot o_{U+1}}_{\text{Term B''}} - \underbrace{2 \cdot n_j^{save} \cdot o_{U+1}}_{\text{Term C''}} \Rightarrow$$

$$K'_A(j) = \underbrace{\sum_{i=1}^{U} \ell_i \cdot o_i}_{\text{Term A1'}} + \underbrace{\sum_{i=1}^{U'} \tilde{\ell}_i \cdot o_i}_{\text{Term A2'}} + \underbrace{\ell_j \cdot o_{U+1}}_{\text{Term B'}} - \underbrace{n_j^{save} \cdot o_{U+1}}_{\text{Term C'}} \Rightarrow$$

$$K_A(j) = \underbrace{\sum_{i=1}^{U'} o_i}_{\text{Term A}} + \underbrace{\ell_j \cdot o_{U+1}}_{\text{Term B}} - \underbrace{n_j^{save} \cdot o_{U+1}}_{\text{Term C}}$$

Der Aufwand für den Gruppenbeitritt und den Gruppenaustritt ist im Modus Key Agreement gleich groß. Aus diesem Grund wird bei der Sammelverarbeitung der Effizienzgewinn in der Kostenfunktion $K_A(j)$ durch die Ersetzung von gelöschten durch neue Knoten (Term C) mit dem Faktor 1 berücksichtigt.

Die bisherigen Betrachtungen zur Auswahl des Einfügepunkts wurden vor dem Hintergrund der Teilnehmeroperation JOIN und MULTIPLE JOIN durchgeführt. Bei der Teilnehmeroperation MERGE wird der Einfügepunktauswahlalgorithmus Balance-Insert verwendet. In diesem Fall wird das Ziel, den Schlüsselbaum zu balancieren, mit der Einschränkung verfolgt, den Schlüsselbaum um maximal eine Ebene zu erweitern. Ist dies nicht möglich, wird für die bereits bestehende Teilgruppe und die neue Teilgruppe ein neuer gemeinsamer Wurzelknoten gebildet. Der Vorteil dieser Vorgehensweise besteht darin, dass beim Beitritt einer großen Teilgruppe zu einer kleinen Teilgruppe eine geringere Zunahme der Schlüsselbaumhöhe erfolgt.

7.4.2 Charakterisierung des Schlüsselbaums

Die Effizienz des Schlüsselmanagements ist vom Aufbau des Schlüsselbaums, der vom Einfügepunktauswahlalgorithmus bestimmt wird, abhängig. Da ein Schlüsselbaum eine komplexe Struktur besitzt, ist die Interpretation der Auswirkungen von unterschiedlichen Strategien bei der Einfügepunktauswahl auf den Schlüsselbaum eine große Herausforderung. Daraus erwächst die Motivation, geeignete Kenngrößen zur Charakterisierung des Schlüsselbaums zu wählen bzw. zu definieren. Die vier Kenngrößen Füllzustand, Links-Rechts-Balance, normierte Entropie und Silhouette Index wurden hierzu ausgewählt. Im Folgenden werden diese Kenngrößen für einen Schlüsselbaum vom Grad d, der Höhe h, mit U Nutzern definiert. Weiterhin wird der Wertebereich angegeben und die Auswahl begründet:

Füllzustand F $$F = \frac{U}{U_{\max}} \text{ mit } U_{\max} = d^h$$

Die Kenngröße Füllzustand (filling) gibt das Verhältnis zwischen im Schlüsselbaum befindlichen Nutzern und den maximal möglichen Nutzern an. Für den Wertebereich der Kenngröße gilt $0<F\leq 1$. Ein Wert eins der Kenngröße F bedeutet, dass der Schlüsselbaum vollständig gefüllt ist und dass zum Einfügen zusätzlicher Nutzer der Baum um eine Ebene erweitert werden muss. Ein Beispiel für die Berechnung der Kenngröße F ist in Abbildung 86 dargestellt.

Rechts-Links-Balance B $$B = \frac{\min(U_{left}, U_{right})}{\max(U_{left}, U_{right})}$$

Die Kenngröße Rechts-Links-Balance (right-left-balance) gibt das Verhältnis zwischen der Anzahl der im rechten und linken Teilbaum befindlichen Nutzer bezogen auf den Wurzelknoten an. Ist bei Schlüsselbäumen mit ungeradem Grad d eine eindeutige Zuordnung eines Nutzers zu einem der beiden Teilbäume nicht möglich, wird dieser bei der Berechnung nicht berücksichtigt. Der Wertebereich der Kenngrößen beträgt $0<B\leq 1$. Befinden sich im rechten und linken Teilbaum gleich viele Nutzer, nimmt die Kenngröße B den Wert eins an. Ein Beispiel für die Berechnung der Kenngröße B ist in Abbildung 86 dargestellt.

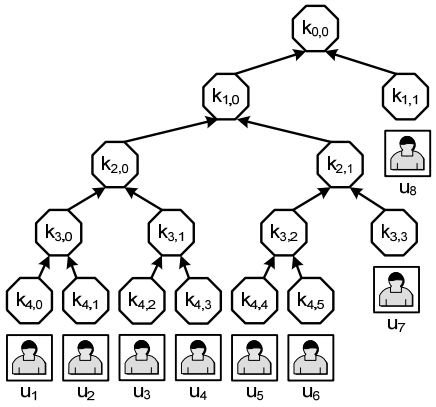

Abbildung 86: Schlüsselbaum mit F=0,500 und B=0,143

Entropie E_{norm} $$E_{norm} = \frac{E}{E_{\max}} \text{ mit } E = \sum_{i=0}^{U} o_i \cdot \ell_i \,,\; E_{\max} = \sum_{i=0}^{U} o_{\max} \cdot \ell_i$$

Die Kenngröße Entropie (entropy) wird verwendet, um zu bewerten, in welchem Umfang dynamische Nutzer näher an der Wurzel des Schlüsselbaums platziert wurden. Durch die Normierung der Kenngröße ergibt sich der Wertebereich $0<E\leq 1$. Je kleiner der Wert der Kenngröße E ist, desto besser ist die Struktur des Schlüsselbaums bezüglich der Entropie.

Silhouette Index S $$S = \tfrac{1}{C}\sum_{k=1}^{C}\left(\tfrac{1}{CU_k}\sum_{i=1}^{CU_k} s_k(i)\right) \text{ mit } s_k(i) = \frac{d(v_{k_i}, cl_{close(i)}) - d(v_{k_i}, cl_k)}{\max(d(v_{k_i}, cl_{close(i)}), d(v_{k_i}, cl_k))}$$

Die Kenngröße Silhouette Index wird verwendet, um die Haufenbildung von Schlüsselbaumknoten eines Clusters zu bewerten, wenn der Baum aus C unterschiedlichen Clustern besteht. Bei der Definition der Kenngröße wird die Anzahl der Knoten des Clusters cl_k mit CU_k bezeichnet. Für die Zuordnung der Knoten des Schlüsselbaums zu dem entsprechenden Cluster wird in jedem Knoten die zugeordnete Cluster-Bezeichnung gespeichert. Mit der Kenngröße Silhouette Index wird die Dichte der Knotenanordnung eines Clusters mit seiner Abtrennung im Baum verglichen. Hierbei sind $d(v_{k_i},cl_k)$ der mittlere Abstand des Knotens i zu allen anderen Knoten des gleichen Clusters cl_k und $d(v_{k_i},cl_{close(i)})$ der mittlere Abstand des Knotens i zum am nächsten gelegenen Cluster $cl_{close(i)}$. Der Abstand $d(v_{k_i},cl_k)$ wird berechnet mit:

$$d(v_{k_i}, cl_k) = \frac{1}{CU_k}\sum_{j=1, j\neq i}^{CU_k} \left\| path(v_{k_i}, v_{k_j}) \right\|$$

Die Beiträge zur Kenngröße Silhouette Index $s_A(i)$ sind in Abbildung 87 veranschaulicht. In dieser Abbildung ist der Knoten i Teil des Clusters A und das Cluster C ist das zu i am Nächsten gelegene Cluster. Für den in Abbildung 87 dargestellten Fall ergibt sich $d(v_{A_i},cl_A)$ aus den rot und $d(v_{A_i},cl_{close(i)})$ aus den blau dargestellten Kanten. Der Wertebereich der Kenngröße beträgt $-1 \leq S \leq 1$. Enthält der Schlüsselbaum nur Knoten des gleichen Clusters, wird der Kenngröße der Wert null zugeordnet. Ist der Wert der Kenngröße S nahe bei eins, bedeutet dieses eine gute Häufung der Nutzer.

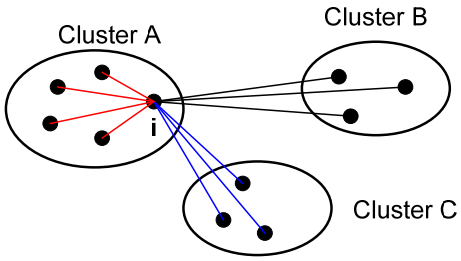

Abbildung 87: Beiträge zur Berechnung $s_A(i)$, wenn i zum Cluster A gehört

Nachdem die Kenngrößen zur Charakterisierung des Schlüsselbaums vorgestellt wurden, soll nun die getroffene Auswahl begründet werden. Die Kenngröße Füllzustand und Rechts-Links-Balance wurden ausgewählt, weil sie die Struktur des Baumes charakterisieren. Hierzu sind zwei Kenngrößen notwendig. Eine Kenngröße charakterisiert die Ausgeglichenheit bezüglich des rechten und linken Teilbaums, während die andere die Balance der Blätter bewertet. Die Kenngröße Entropie wird verwendet, um zu bewerten, in welchem Umfang dynamische Nutzer näher an der Wurzel des Schlüsselbaums platziert wurden. Bietet ein Schlüsselmanagement auch die Möglichkeit zur Durchführung einer Sammelverarbeitung, ist diese effizienter, wenn Teilnehmer, die innerhalb einer Sammelphase die Gruppe verlassen, im Schlüsselbaum „nah beieinander angeordnet“ sind. Der Effizienzgewinn wird durch sich überlagernde Pfade und damit einer geringeren Anzahl zu aktualisierender Knoten verursacht. Zur Quantifizierung der Haufenbildung der Knoten des gleichen Clusters eignen sich so genannte Cluster Validity Indices. In [Bou04] werden eine Vielzahl dieser Indizes vorgestellt. Die oben vorgestellte Kenngröße Silhouette Index wurde aus zwei Gründen ausgewählt. Zum einen sollte sie eine Bewertung der Häufung von Knoten auf Grund der Knotennachbarschaft vornehmen, zum anderen sollte die Kenngröße zur Quantifizierung der Haufenbildung unabhängig von der Anzahl der Cluster sein.

7.4.3 Ergebnisse der Effizienzmessung

Zur Messung der Effizienz der nutzerverhaltensbasierten Schlüsselbaumkonstruktion wird der in Abschnitt 7.2 vorgestellte Messaufbau eingesetzt. Als Metrik wird im Betriebsmodus Key Agreement die Summe der kryptographischen Operationen des TMs bzw. im Betriebsmodus Key Distribution die Summe der übertragenen Schlüsselpakete für die Dauer des Nutzerverhaltens verwendet.

Zunächst werden die Parameter der Einfügepunktauswahlalgorithmen bei der Effizienzuntersuchung erläutert. Für den Algorithmus 2-Partition-Insert liefert keines der beiden Nutzerverhalten eine Vorgabe für die Größe der dynamischen Partition. Es wurden deshalb für die Partitiongröße die Werte $S_{d\text{-}Part}=5$, $S_{d\text{-}Part}=10$ und $S_{d\text{-}Part}=15$ verwendet. Mit

$S_{d\text{-}Part}$ wird die maximale Anzahl der Blätter in der dynamischen Partition bezeichnet. Für den Algorithmus Entropy-Insert ist eine Aufteilung der Nutzer in Cluster notwendig. Für das militärische Nutzerverhalten erfolgt eine Aufteilung in drei Cluster. Diese Aufteilung resultiert aus der organisatorischen Struktur der Nutzer. Bei dem militärischen Nutzerverhalten setzt sich die Gruppe aus den Nutzern des Kontrollpunkts, der Fahrzeugpatrouille sowie den Unterstützungstruppen zusammen (vgl. Abschnitt 6.3).

Nutzerverhalten	Optimierungsfaktoren			
	o_1 (cId=1)	o_2 (cId=2)	o_3 (cId=3)	Δo_{max}
ziviles Nutzerverhalten Nr. 1 (Aufteilung 1)	0,638938	0,227438	0,133624	0,505314
ziviles Nutzerverhalten Nr. 2 (Aufteilung 1)	0,759595	0,149358	0,091047	0,668548
ziviles Nutzerverhalten Nr. 2 (Aufteilung 2)	0,820068	0,126350	0,053582	0,766486
ziviles Nutzerverhalten Nr. 2 (Aufteilung 3)	0,836633	0,127256	0,036110	0,800523
militärisches Nutzerverhalten Nr. 1, 2	0,142857	0,285714	0,571429	0,428571

Tabelle 23: Optimierungsfaktoren

Die mittlere Teilnahmedauer ergibt sich beim militärischen Nutzerverhalten aus der durchzuführenden Aufgabe und ist damit dem Nutzer bekannt. Ein Nutzer kann somit seinen Optimierungsfaktor durch Invertieren und Normieren berechnen und bei der Gruppenanmeldung übermitteln. Die bei der Messung unter Einfluss des militärischen Nutzerverhaltens verwendeten Optimierungsfaktoren sind in Tabelle 23 zusammengefasst. Bestandteil dieses Nutzerverhaltens ist der Ausschluss von kompromittierten Nutzern. Die daraus resultierende verkürzte Teilnahmedauer ist den Nutzern nicht bekannt und wird deshalb auch bei der Berechnung der Optimierungsfaktoren nicht berücksichtigt. Eine Aufteilung der Nutzer in Cluster und die Zuweisung eines Optimierungsfaktors ist beim zivilen Nutzerverhalten nur schwer möglich. Um bei diesem Nutzerverhalten unabhängig von der Problematik der Zuweisung von Optimierungsfaktoren den Effizienzgewinn des Konzepts bewerten zu können, wird im Vorfeld einer Messung das vom Lastgenerator erzeugte zivile Nutzerverhalten zur Ermittlung der Optimierungsfaktoren ausgewertet. Anschließend wird dann mit den ermittelten Faktoren die Messung durchgeführt. Bei dieser Auswertung werden die Nutzer auf der Basis ihrer Teilnahmedauer einem der drei Cluster $\{cId_1=1, cId_2=2, cId_3=3\}$ zugeordnet und ihnen der für das Cluster verwendete Optimierungsfaktor zugewiesen.

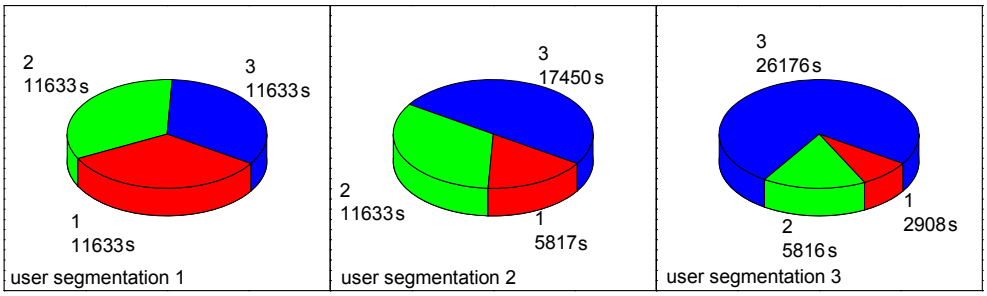

Abbildung 88: Arten der Aufteilung der Nutzer in die drei Cluster $cId_1=1$, $cId_2=2$, $cId_3=3$ und mittlere Teilnahmedauer beim zivilen Nutzerverhalten

Die Einteilung der Nutzer in drei Cluster beim zivilen Nutzerverhalten wurde aus Gründen der Vergleichbarkeit mit Messungen unter Einfluss des militärischen Nutzerverhaltens

vorgenommen. Da die obere und untere Schranke für die Teilnahmedauer bzw. mittlere Teilnahmedauer der drei Cluster nicht durch das Nutzerverhalten vorgegeben ist, wurden die in Abbildung 88 dargestellten drei verschiedenen Aufteilungen der Nutzer (user segmentation) untersucht. Durch Invertieren und Normieren der mittleren Teilnahmedauer ergeben sich die in Tabelle 23 zusammengefassten Optimierungsfaktoren für das zivile Nutzerverhalten.

In Tabelle 24 und Tabelle 25 ist der prozentuale Effizienzgewinn durch die Einfügepunktauswahlalgorithmus 2-Partion-Insert und Entropy-Insert im Vergleich zum Algorithmus Balance-Insert für den Modus Key Agreement bzw. Key Distribution dargestellt. Negative Werte bedeuten eine schlechtere Effizienz. Der Einfügepunktauswahlalgorithmus 2-Partition-Insert bewirkt für den Fall der Einzelverarbeitung von Teilnehmeroperationen bei beiden Betriebsmodi eine Effizienzminderung. Grund hierfür sind die mit den Veränderungen der Nutzerpositionen im Baum, d.h. die Verschiebung von der Dynamic Partion in die Stable Partion, verbundenen hohen Kosten. Im Fall der Sammelverarbeitung von Teilnehmeroperationen bewirkt der Algorithmus 2-Partition-Insert mit entsprechender Größe der dynamischen Partition beim militärischen Nutzerverhalten eine Effizienzsteigerung. Diese wird dadurch hervorgerufen, dass beim militärischen Nutzerverhalten Nutzer, die dem gleichen Cluster zugeordnet sind, in die dynamische Partition eingefügt werden. Beim gemeinsamen Austritt dieser Nutzer ergibt sich auf Grund der überlagerten Pfade eine Verbesserung der Effizienz.

Nutzerverhalten	Algorithmus für die Einfügepunktauswahl	Einzelverarbeitung: Gewinn kryptograph. Operationen [%]	Sammelverarbeitung: Gewinn kryptograph. Operationen [%]
ziviles Nutzerverhalten Nr. 1	2-Partition-Insert, $S_{d\text{-}Part}$=5	-39,0	-36,8
	2-Partition-Insert, $S_{d\text{-}Part}$=10	-37,9	-40,0
	2-Partition-Insert, $S_{d\text{-}Part}$=15	-33,7	-36,4
	Entropy-Insert, Aufteilung 1	0,2	6,7
militärisches Nutzerverhalten Nr. 1	2-Partition-Insert, $S_{d\text{-}Part}$=5	-38,6	5,2
	2-Partition-Insert, $S_{d\text{-}Part}$=10	-37,8	5,6
	2-Partition-Insert, $S_{d\text{-}Part}$=15	-35,5	6,9
	Entropy-Insert	11,8	32,5

Tabelle 24: Effizienzgewinn durch den Einfügepunktauswahlalgorithmus im Vergleich zu Balance-Insert im Modus Key Agreement

Für den Einfügepunktauswahlalgorithmus Entropy-Insert kann in allen untersuchten Fällen eine Verbesserung der Effizienz bezüglich der Anzahl der kryptographischen Operationen im Modus Key Agreement bzw. bezüglich der Anzahl der ausgetauschten Schlüsselpakete im Fall Key Distribution festgestellt werden. Allerdings fällt die erzielte Verbesserung für den Modus Key Agreement unter Einfluss des zivilen Nutzerverhaltens mit 0,2 % sehr klein aus. Grund hierfür ist eine durch die geringe Nutzerzahl auftretende geringe Pfadlänge zur Wurzel des Schlüsselbaums. Weiterhin lässt sich feststellen, dass der Effizienzgewinn durch den Algorithmus Entropy-Insert im Fall der Sammelverarbeitung größer ist als bei der Einzelverarbeitung von Teilnehmeroperationen. Bevor die Ursache für den Effizienzgewinn durch den Einfügepunktauswahlalgorithmus Entropy-Insert unter dem Einfluss des

militärischen Nutzerverhaltens genauer analysiert wird, wird noch die gewählte Aufteilung der Nutzer in Cluster beim zivilen Nutzerverhalten untersucht. Betrachtet man den erzielten Effizienzgewinn bei unterschiedlichen Nutzeraufteilungen, so kann dieser durch eine geeignete Wahl bis auf 7,4 % erhöht werden. Diese Steigerung ist darauf zurückzuführen, dass bei dem Algorithmus Entropy-Insert in Kombination mit der Aufteilung 3 die dynamischen Nutzer besser identifiziert werden. Grund der verbesserten Identifizierung dynamischer Nutzer ist eine im Vergleich zu den anderen Aufteilungen höhere maximale Differenz zwischen den Optimierungsfaktoren. Je größer die maximale Differenz zwischen den Optimierungsfaktoren ist, desto mehr kann durch die nutzerverhaltenbasierte Schlüsselbaumkonstruktion eine Leistungsverbesserung erzielt werden.

Nutzerverhalten	Algorithmus für die Einfügepunktauswahl	Einzelverarbeitung: Gewinn Schlüsselpakete [%]	Sammelverarbeitung: Gewinn Schlüsselpakete [%]
ziviles Nutzerverhalten Nr. 2	2-Partition-Insert, $S_{d\text{-}Part}$=5	-25,1	-13,7
	2-Partition-Insert, $S_{d\text{-}Part}$=10	-27,7	-22,3
	2-Partition-Insert, $S_{d\text{-}Part}$=15	-32,0	-25,9
	Entropy-Insert, Aufteilung 1	4,4	6,4
	Entropy-Insert, Aufteilung 2	5,8	-
	Entropy-Insert, Aufteilung 3	7,4	-
militärisches Nutzerverhalten Nr. 2	2-Partition-Insert, $S_{d\text{-}Part}$=5	-33,0	-0,9
	2-Partition-Insert, $S_{d\text{-}Part}$=10	-40,4	0,7
	2-Partition-Insert, $S_{d\text{-}Part}$=15	-37,1	1,3
	Entropy-Insert	7,3	8,5

Tabelle 25: Effizienzgewinn durch den Einfügepunktauswahlalgorithmus im Vergleich zu Balance-Insert im Modus Key Distribution

Zur Interpretation der Auswirkung von unterschiedlichen Strategien bei der Einfügepunktauswahl wurden in Abschnitt 7.4.2 vier Kenngrößen zur Charakterisierung des Schlüsselbaums definiert. Die Ursachen für den Effizienzgewinn durch den Einfügepunktauswahlalgorithmus Entropy-Insert werden nun mit Hilfe dieser Kenngrößen erläutert. Bei der Interpretation der Kenngrößen ist zu beachten, dass bei der Kenngröße Entropie im Gegensatz zu den Übrigen „je kleiner desto besser“ gilt. In Abbildung 89 ist der Verlauf der Kenngrößen im Modus Key Distribution während des zivilen Nutzerverhaltens für den Einfügepunktauswahlalgorithmus Balance-Insert und Entropy-Insert dargestellt. Wie erwartet, vermindert der Algorithmus Entropy-Insert die Kenngröße Entropie des Schlüsselbaums. Weiterhin wird die Kenngröße Füllzustand verschlechtert. In der Abbildung 89 kann man erkennen, dass die Minimierung der Kenngröße Entropie eine Verschlechterung der Kenngrößen Rechts-Links-Balance und Silhouette Index verursacht. Vergleicht man den Verlauf der vier Kenngrößen während des zivilen Nutzerverhaltens in Abbildung 89 mit dem während des militärischen Nutzerverhaltens in Abbildung 90, so stellt man fest, dass nach einer kurzen Phase zu Beginn sich nur noch wenig ändert. Im Gegensatz dazu ändern sich die Werte der Kenngrößen beim militärischen Nutzerverhalten sehr stark.

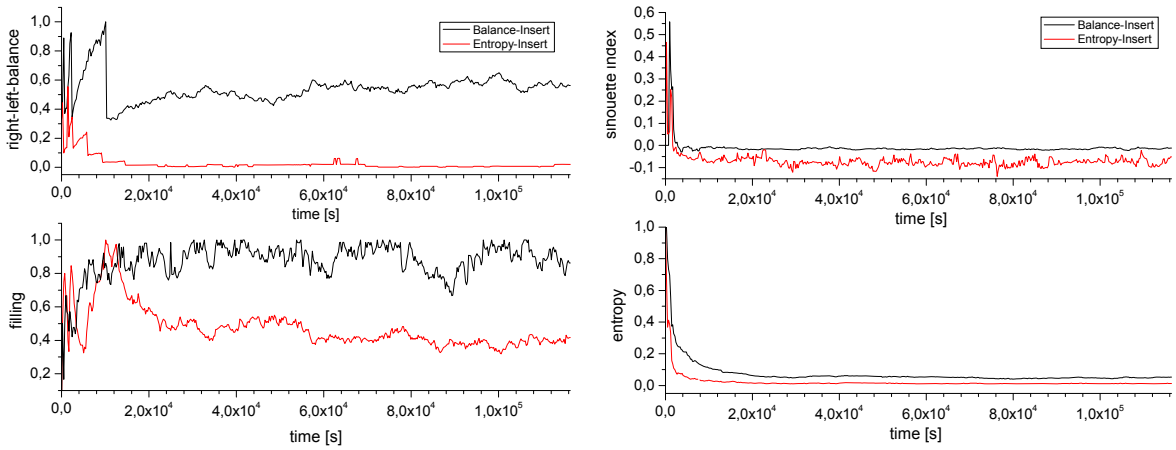

Abbildung 89: Kenngrößen zur Charakterisierung des Schlüsselbaums während des zivilen Nutzerverhaltens im Modus Key Distribution

Schwerpunkt der weiteren Erläuterungen sind die Analyse der Auswirkungen des Algorithmus Entropy-Insert auf die Baumstruktur mittels der Kenngrößen Entropie und Silhouette Index beim militärischen Nutzerverhalten. Insbesondere soll der Unterschied zwischen Einzel- und Sammelverarbeitung verdeutlicht werden. In Abbildung 90 zeigt der für den Betriebsmodus Key Distribution dargestellte Verlauf der Kenngrößen das bekannte Verhalten, dass eine Verbesserung der Kenngröße Entropie mit einer Verschlechterung der Kenngröße Silhouette Index verbunden ist. Betrachtet man den Verlauf der Kenngrößen bei der Sammelverarbeitung, kann man feststellen, dass durch den Term C der Kostenfunktion $K_D(j)$ aus Abschnitt 7.4.1 eine Effizienzsteigerung durch eine erhebliche Verbesserung der Kenngröße Silhouette Index zu Lasten der Kenngröße Entropie erzielt wird. Ein hoher Wert der Kenngröße Silhouette Index bewirkt, dass Nutzer mit gleichem Optimierungsfaktor, bzw. gleicher Wahrscheinlichkeit, die Gruppen zu verlassen, im Baum dicht beieinander liegen. Der aus dieser Anordnung resultierende Effizienzgewinn wird durch überlappende Pfade zur Wurzel bewirkt.

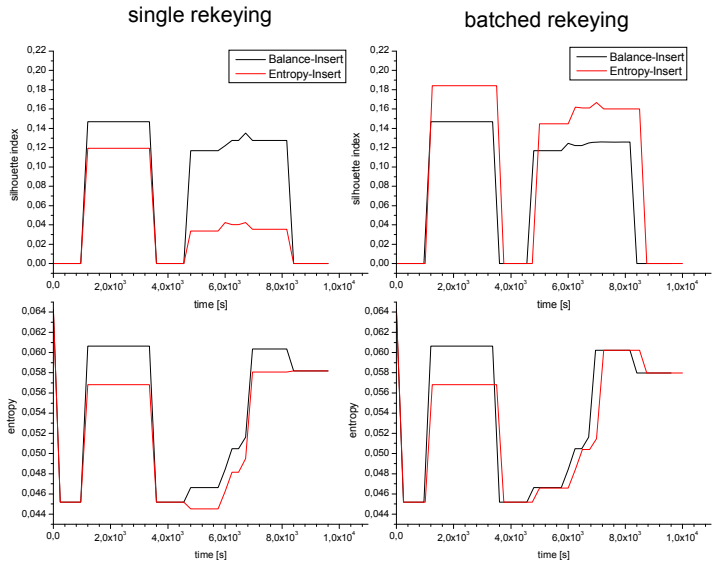

Abbildung 90: Kenngrößen Entropie und Silhouette Index während des militärischen Nutzerverhaltens im Modus Key Distribution

Aus Abbildung 91 entnimmt man, dass der Effekt im Modus Key Agreement noch größer ist. In diesem Modus sorgt der Algorithmus Entropy-Insert dafür, dass es bei der Sammelverarbeitung zu einer erheblichen Verbesserung der Kenngröße Silhouette Index, d.h. Häufung der Nutzer des gleichen Clusters, zu Lasten der Kenngröße Entropie kommt. Diese Wirkungsweise der Kostenfunktion $K_A(j)$ aus Abschnitt 7.4.1 kann durch gleichen Aufwand für Gruppenbeitritt und Gruppenaustritt erklärt werden. Bei der Sammelverarbeitung im Modus Key Agreement wird im Gegensatz zum Modus Key Distribution der Einfügepunkt maßgeblich durch den Effizienzgewinn beim Einfügen beeinflusst.

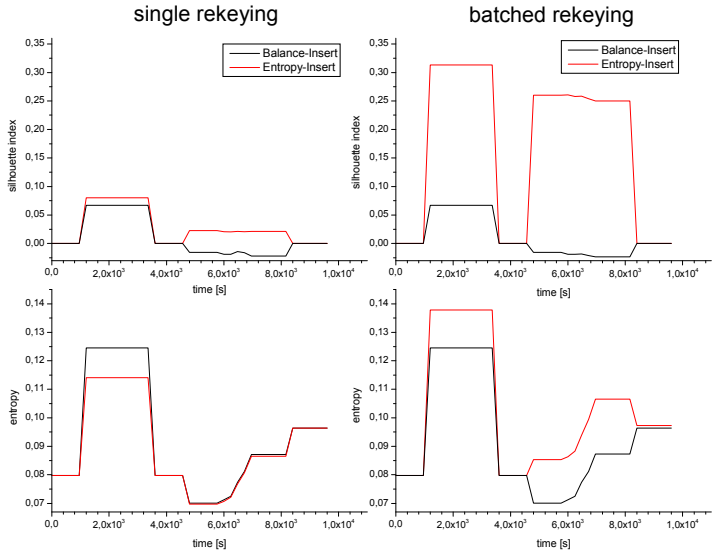

Abbildung 91: Kenngrößen Entropie und Silhouette Index während des militärischen Nutzerverhaltens im Modus Key Agreement

7.5 Ressourcengesteuerte Auswahl des Transaction Managers

Ein Schlüsselmanagement, das in den Streitkräften eingesetzt werden soll, muss auch eine Schlüsselverwaltung in Gruppen ermöglichen, in denen einige Gruppenteilnehmer temporär nur über eine Simplexverbindung verfügen. Eine derartige Situation ist die Folge des Betriebs der Kommunikationsmittel im Zustand EMCON, in dem zum Schutz vor Aufklärungsmaßnahmen Daten empfangen aber nicht versendet werden dürfen. In diesem Abschnitt wird die Gewährleistung der Schlüsselbereitstellung durch das Schlüsselmanagement MIKE für den Fall, dass sich Nutzer im Zustand EMCON befinden, erläutert. Die ressourcengesteuerte Auswahl des TM wurde entworfen, um einen Betrieb im Modus Key Agreement mit Nutzern, die sich im Zustand EMCON befinden, zu ermöglichen. Allerdings kann sie auch eingesetzt werden, um zu verhindern, dass Nutzer mit anderen Ressourcenlimitierungen, z.B. geringe Rechenleistung oder Datenübertragungskapazität, die ressourcenintensive Aufgabe des TM übernehmen. Deshalb wird der Mechanismus als ressourcengesteuerte Auswahl des Transaction Managers bezeichnet.

Ein Einsatz des Betriebsmodus Key Distribution in Gruppen mit einer derartigen Charakteristik ist ohne Erweiterung des Konzepts möglich, da zur Übermittlung des Gruppenschlüssels keine Duplexkommunikation notwendig ist. Bei der Konfiguration eines

MIKE-Prozesses ist lediglich darauf zu achten, dass der Gültigkeitszeitraum der Gruppenanmeldung entsprechend groß gewählt wurde.

7.5.1 Algorithmus für eine ressourcengesteuerte Auswahl des Transaction Managers

Um einen Einsatz des Betriebsmodus Key Agreement in Gruppen mit Nutzern im Zustand EMCON zu ermöglichen, muss verhindert werden, dass diese Teilnehmer die Aufgabe des TM übernehmen. Um dies zu gewährleisten, wurde ein Algorithmus zur ressourcengesteuerte Auswahl des Transaction Managers entworfen. Dieser beinhaltet folgende zwei Funktionalitäten:

- Knoten, denen Nutzer im Zustand EMCON zugeordnet sind, werden nicht als Einfügepunkt ausgewählt.
- Kann beim Austritt eines Nutzers kein TM für die Schlüsselbereitstellung gefunden werden, wird der Knoten des gelöschten Nutzers durch Nutzerverschiebung ersetzt. Hierzu wird der am nächsten gelegene Knoten gewählt, dessen Geschwisterknoten sich nicht im Zustand EMCON befindet.

Der Algorithmus gewährleistet die Schlüsselbereitstellung in Gruppen, bei denen der Anteil der Nutzer im Zustand EMCON kleiner als 50 % ist. Der Algorithmus ressourcengesteuerte Auswahl des Transaction Managers beinhaltet keine Signalisierung des Nutzerzustands EMCON. Eine Möglichkeit, den TM über den EMCON-Zustand zu informieren, besteht darin, dieses bei der Gruppenanmeldung zu übermitteln. In Abbildung 92 wird die Funktionsweise des Algorithmus an zwei Beispielen erläutert. Im ersten Beispiel tritt der Nutzer u_4 einer Gruppe mit drei Nutzern bei, wobei sich der Nutzer u_3 im Zustand EMCON befindet. Der Einfügepunktauswahlalgorithmus Balance-Insert würde den Knoten $v_{1,1}$ zur Integration des neuen Nutzers in den Baum auswählen. Der Algorithmus zur ressourcengesteuerten Auswahl des TM verhindert dies und wählt stattdessen den Knoten $v_{2,0}$. Im zweiten Beispiel verlässt der Nutzer u_6 eine Gruppe mit sieben Nutzern, wobei sich die Nutzer u_3, u_5, u_7 im Zustand EMCON befinden. Da der Nutzer u_5 auf Grund des Zustands EMCON nicht die Aufgabe des TM übernehmen kann, wird der Nutzer u_2 an die Baumposition $\ell=3$, $p=5$ verschoben und aktualisiert als TM den Pfad zur Schlüsselbaumwurzel.

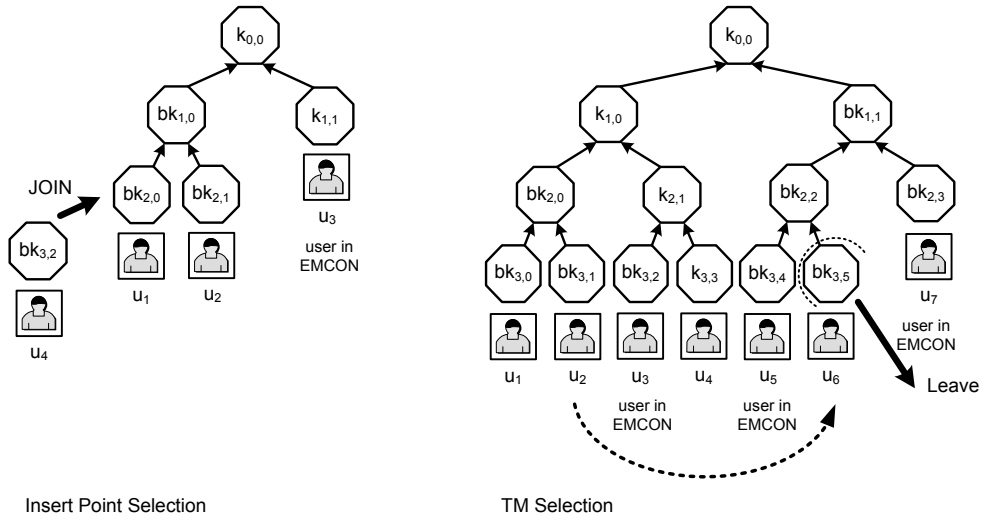

Abbildung 92: Einfügepunktauswahl (links) und Nutzerverschiebung im Schlüsselbaum (rechts) mit Teilnehmern im Zustand EMCON

7.5.2 Messung der Effizienz

In diesem Abschnitt wird die Effizienz der ressourcengesteuerten Auswahl des TM in Kombination mit Einfügepunktwahlalgorithmus Balance-Insert und Einzelverarbeitung von Nutzeranfragen untersucht. Zur Messung wird der in Abschnitt 7.2 vorgestellte Messaufbau eingesetzt. Als Metrik wird die Summe der kryptographischen Operationen des TMs während der Dauer des militärischen Nutzerverhaltens verwendet. Die Auswirkung der ressourcengesteuerten Auswahl des TM wird nur in Gruppen mit militärischem Nutzerverhalten untersucht, da hier dessen Einsatzbereich liegt. Für die Messung wird angenommen, dass 10 % bis 40 % der Nutzer nach ihrem Gruppenbeitritt in den Zustand EMCON wechseln und diesen Zustand erst kurz vor ihrem Gruppenaustritt wieder verlassen.

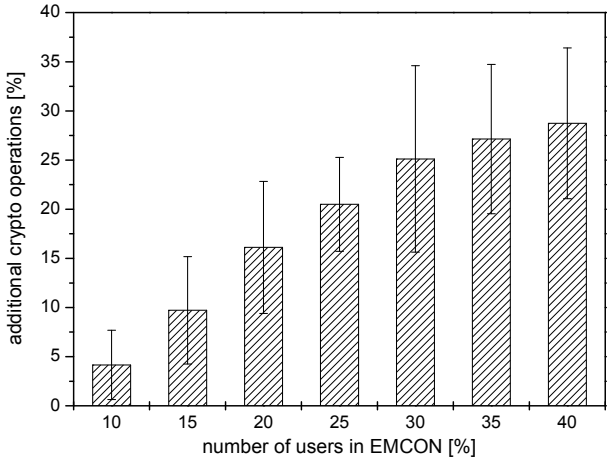

Abbildung 93: Zuwachs der Anzahl der durchzuführenden kryptographischen Operationen beim militärischen Nutzerverhalten Nr. 1 mit Teilen der Gruppe im Zustand EMCON

In Abbildung 93 sind die zusätzlich durchzuführenden kryptographischen Operationen dargestellt, falls sich Teile der Gruppe im Zustand EMCON befinden. Die angegebenen Prozentzahlen beziehen sich auf eine Gruppe der gleichen Größe, in der sich kein Nutzer im

Zustand EMCON befindet. Die Abbildung zeigt, dass, wie in Abschnitt 3.2 gefordert, auch in Gruppen, in denen sich Teile der Nutzer im Zustand EMCON befinden, eine Schlüsselbereitstellung durch das Konzept MIKE ermöglicht wird. Allerdings sind bei der Schlüsselbereitstellung zusätzliche kryptographische Operationen notwendig. Ursache hierfür ist die erhöhte Pfadlänge zur Wurzel, da nicht alle Knoten des Baums als Einfügepunkt zur Verfügung stehen. Außerdem wird die Anzahl der kryptographischen Operationen erhöht, wenn durch Nutzerverschiebung gewährleistet werden muss, dass ein Nutzer im Zustand EMCON nicht TM wird. Das in Abbildung 93 dargestellte Ergebnis wurde in einer Messreihe ermittelt, bei der die Nutzer im Zustand EMCON zufällig gewählt wurden. Eine Streuung der Messwerte tritt auf, da die notwendigen zusätzlichen kryptographischen Operationen von der Baumposition eines Nutzers abhängen.

7.6 Kapitelzusammenfassung

Inhalt des aktuellen Kapitels sind Maßnahmen zur Leistungsverbesserung des Schlüsselmanagements MIKE durch Anpassung an die militärische Verwendung. Hierbei wurde eine Effizienzsteigerung für beide Betriebsarten erzielt, indem die Verarbeitung des Schlüsselbaums verbessert wurde. Zum einen wurde zur Optimierung des Schlüsselmanagements die Sammelverarbeitung von Nutzeranfragen entwickelt, um einen Algorithmus für oftmals auftretende Burst-Teilnehmeroperationen bereitzustellen. Zum anderen wurde ein Algorithmus zur nutzerverhaltenbasierten Schlüsselbaumkonstruktion entworfen, der in Kombination mit der Sammelverarbeitung einsetzbar ist, und berücksichtigt, dass militärische Gruppen aus Teilgruppen mit vorher bekanntem Verhaltensmuster bestehen. Die Effizienzanalyse der beiden Optimierungstechniken zeigt, dass durch die Sammelverarbeitung im Vergleich zur nutzerverhaltenbasierten Schlüsselbaumkonstruktion ein größerer Effizienzgewinn erzielt werden kann. Die Auswirkung dieser nutzerverhaltenbasierten Schlüsselbaumkonstruktion auf die Struktur des Schlüsselbaums wurde mittels der charakteristischen Schlüsselbaummerkmale Füllzustand, Rechts-Links-Balance, Entropie und Silhouette Index veranschaulicht. Um den Betrieb des Schlüsselmanagements auch im Modus Emission Control zu ermöglichen, wurde die ressourcengesteuerte Auswahl des Transaction Managers entwickelt. Das Ergebnis der Effizienzanalyse zeigt, dass zusätzliche kryptographische Operationen notwendig sind, falls einige Nutzer im Zustand EMCON sind und nicht als TM zur Verfügung stehen.

8 Leistungsbewertung des Schlüsselmanagements MIKE

In diesem Kapitel wird eine Leistungsbewertung mittels der in Kapitel 6 definierten Metriken durchgeführt. Hierbei wird die Analyse der Verlässlichkeit und Effizienz des Schlüsselmanagements jeweils in einem separaten Abschnitt durchgeführt. Das Kapitel wird abgeschlossene mit einem Vergleich des Schlüsselmanagement MIKE mit exitierenden Konzepten.

Als Einführung in die Leistungsbewertung wird in kompakter Form dargestellt, dass das Konzept MIKE die in Abschnitt 3.3 festgelegten Anforderungen erfüllt. Durch die im Konzept enthaltenen Mechanismen kann in einer dynamischen Gruppe ein teilnehmersensitiver Schlüsselwechsel durchgeführt werden. Es ist deshalb die Forderung 1 der Tabelle 6, d.h. Schlüsselgeheimhaltung, erfüllt. Die Authentizität der ausgetauschten Managementnachrichten wird durch digitale Signaturen gewährleistet. Zertifikate dienen zur Gültigkeitsüberprüfung der eingesetzten Authentisierungsschlüssel. Die Verwendung von digital signierten Nachrichten beim 3-Wege-Anmeldevorgang in Kombination mit Zufallsfallszahlen schützt vor Angriffen durch Übertragungswiederholung während der Gruppenanmeldung. Dadurch wird die Forderung 2, Tabelle 6 nach einer störsicheren Zugriffskontrolle realisiert. Diese kann durch den in ISAKMP enthaltenen Cookie-Mechanismus verbessert werden, der einen Schutz des Schlüsselmanagements gegen Fluten mit Beitrittsanfragen ermöglicht. Eine entsprechende Schnittstelle zum IPSec-Kernel-Modul ermöglicht dem Konzept MIKE die Gruppenschlüsselbereitstellung für das Sicherheitsprotokoll IPSec (vgl. Forderung 8, Tabelle 6). Um, wie in Forderung 4 der Tabelle 6 beschrieben, einen robusten Schlüsselwechsel zu gewährleisten, kompensieren Block Erasure Code Paketverluste bei der Übertragung der Schlüsselwechselinformationen. Mit dem Betriebsmodus Key Agreement wird entsprechend der Forderung 5 in Tabelle 6 ein reparierbares Schlüsselmanagement für kleine Gruppen zur Verfügung gestellt. Der zweite Betriebsmodus Key Distribution gewährleistet auch noch eine Schlüsselbereitstellung bei steigender Gruppengröße (vgl. Forderung 3, Tabelle 6). Der Betriebsmodus Key Distribution besitzt allerdings einen Single Point of Failure und erfüllt somit eigentlich nicht die Forderung Reparierbarkeit. Dieser Nachteil muss akzeptiert werden, um auch in größeren Gruppen eine Schlüsselbereitstellung zu ermöglichen. Um in einem kurzen Zeitraum auftretende Mehrfachanfragen zum Gruppenbeitritt bzw. Gruppenaustritt effizient verarbeiten zu können, wurde in Kapitel 7 die Sammelverarbeitung von Nutzeranfragen vorgestellt. Diese dient dazu, die Forderung 7 in Tabelle 6 zu erfüllen. Die ebenfalls in diesem Kapitel beschriebene ressourcengesteuerte Auswahl des TM wurde entworfen, um die Forderung 8, d.h. Schlüsselübermittlung in Gruppen mit Nutzern im Zustand EMCON, zu erfüllen. Da bei dem Konzept MIKE keine Neuverschlüsselung des Gruppenschlüssels durchgeführt wird, ist die Schlüsselbereitstellung auch für Realzeit-Anwendungen einsetzbar (Forderung 9, Tabelle 6).

8.1 Analyse der Verlässlichkeit

Ein Schlüsselmanagementsystem, das in einem kritischen Einsatzbereich verwendet werden soll, muss hinsichtlich seiner Verlässlichkeit bewertet werden. Hierzu wird die in Kapitel 6

eingeführte Reparaturzeit t_{TTR} verwendet und die Fehlerklasse Zusammenbruchsfehler (Crash) angenommen. Im Modus Key Agreement ist das System bezüglich eines Zusammenbruchsfehlers eines Nutzers und des TM reparierbar. Im Modus Key Distribution ist diese Reparierbarkeit nur für den Zusammenbruchsfehler eines Nutzers gewährleistet. Ein Ausfall des GC kann nicht abgefangen werden.

Die Reparaturzeit $t_{TTR_{KA}}$(user) bei einem Zusammenbruchsfehler eines Nutzers im Modus Key Agreement wird nach dem Überschreiten der Zeitschranke für die Gültigkeit der Gruppenmeldung $t_{UserTimeout}$ diagnostiziert. Zur Reparatur des Fehlers wird der Nutzer aus dem Baum entfernt und ein Schlüsselwechsel durchgeführt. Die für den Schlüsselwechsel benötigte Zeit beträgt $t_{Update*}$ (vgl. Abbildung 66). Somit ergibt sich:

$$t_{TTR_{KA}}(user) = t_{UserTimeout} + t_{Update*}$$

Bei einem Zusammenbruchsfehler des TM wird der in Abschnitt 4.6.1 beschriebene Mechanismus ausgeführt. Ein derartiger Fehler wird im ungünstigen bzw. im günstigen Fall nach der folgenden Zeit diagnostiziert:

- Ungünstiger Fall
  $$t_{Diagnosis} = t_{KeyTimeout} + t_{\max MsgTimeout(p1JoinRequest)}$$
- Günstiger Fall
  $$t_{Diagnosis} = t_{MsgTimeout(p3TMDistribute)} + t_{\max MsgTimeout(p1JoinRequest)}$$

Hierbei bezeicht $t_{KeyTimeout}$ die Gültigkeitsdauer des Gruppenschlüssels und $t_{maxMsgTimeout(p3TMDistribute)}$ die Zeit in der nach Erhalt der Nachricht `p3TMDistribute`, die Nachricht `p3UpdateDistribute` erwartet wird. Da zunächst ein Nutzer annimmt, dass ein TM existiert, versucht er eine erneute Gruppenanmeldung (vgl. Abschnitt 4.6.1). Nach der Zeit $t_{maxMsgTimeout(p1JoinRequest)}$ mit n erfolglosen Versuchen zur Gruppenanmeldung, d.h. mit n Übertragungen der Nachricht `p1JoinRequest`, wird der Mechanismus zur Wahl des TM gestartet. Zur Einigung auf einen neuen TM müssen bei U Nutzern mindestens 2·U Nachrichten `p3Join` übertragen werden. Wird eine Einigung über den TM erzielt, wird für die Bestätigung dieser Einigung die Zeit $t_{p3Commit}+t_{p3Commit\text{-}Ack}$ benötigt. Die im Baum verbleibenden Nutzer führen anschließend einen Schlüsselwechsel durch. Die für den Schlüsselwechsel benötigte Zeit beträgt $t_{Update*}$ (vgl. Abbildung 66). Die Reparaturzeit $t_{TTR_{KA}}$(TM) bei einem Zusammenbruchsfehler des TM kann abgeschätzt werden durch:

$$t_{TTR_{KA}}(TM) = t_{Diagnosis} + 2 \cdot U \cdot t_{p3Join} + t_{p3TMCommit} + (U-1) \cdot t_{p3TMCommit\text{-}Ack} + t_{Update*}$$

Der Zusammenbruchsfehler eines Nutzers im Modus Key Distribution wird nach dem Überschreiten der Zeitschranke für die Gültigkeit der Gruppenanmeldung $t_{UserTimeout}$ diagnostiziert. Zur Reparatur des Fehlers wird der Nutzer aus dem Baum entfernt und ein Schlüsselwechsel durchgeführt. Die für den Schlüsselwechsel benötigte Zeit beträgt t_{Update} (vgl. Abbildung 66). Somit ergibt sich für die Reparaturzeit $t_{TTR_{KD}}$(user):

$$t_{TTR_{KD}}(user) = t_{UserTimeout} + t_{Update}$$

Zur experimentellen Untersuchung der Verlässlichkeit des Schlüsselmanagements MIKE wird die Methode der software-implementierten Fehlerinjektion auf Systemebene verwendet [Ech98]. Hierzu werden künstlich erzeugte Fehler in das System eingebracht. Außerdem

werden bei dieser Methode Fehler an einem anderen Ort simuliert, anstatt diese direkt in eine Komponente zu initiieren. In nebenläufigen Systemen aus mehreren Prozessen kann diese Methode sehr vorteilhaft angewandt werden, da hierbei nicht das interne Verhalten eines Prozesses untersucht werden soll, sondern die Auswirkung auf andere Prozesse. Bei einem derartigen Vorgehen erwartet man keine vollständige Prüfung des Systems auf Verlässlichkeit. Außerdem wird mit dieser Methode nur eine qualitative Analyse der Verlässlichkeit in Bezug auf die Fehlertoleranz bei dem angenommenen Fehlermodell durchgeführt.

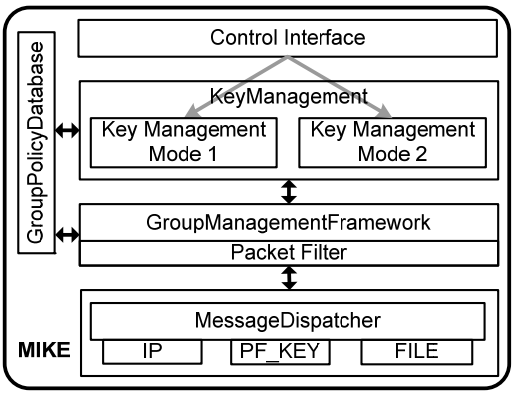

Abbildung 94: Module von MIKE mit integriertem Paketfilter zur Analyse der Verlässlichkeit

Zu diesem Zweck der Fehlerinjektion wurde in das Modul `GroupManagementFramework` ein Paketfilter eingebaut (Abbildung 94). Dieses kann einzelne Nachrichten eines MIKE-Prozesses verwerfen. Hierdurch können Fehler der Fehlerklasse Zusammenbruchsfehler-Wiederherstellung des Kommunikationssystems erzeugt werden.

Mit Hilfe des Paketfilters wurden die in Tabelle 26 beschriebenen Tests erfolgreich durchgeführt (vgl. Zustandsautomaten in Anhang). Dabei wurden nur die Fehlertoleranzverfahren für temporäre Kommunikationsfehler getestet.

Ereignis	Anzahl der Verluste	Reaktion Nutzer	Reaktion TM/GC
Verlust der Nachricht `p1JoinRequest`	≤ Maximale Anzahl der Übertragungen	Übertragungen von `p1JoinRequest`	-
Verlust der Nachricht `p1JoinRequest`	> Maximale Anzahl der Übertragungen	Übergang in den Zustand `AgreeStateInit`/ `DistStateClInit`	-
Verlust der Nachricht `p1JoinDistribute`	≤ Maximale Anzahl der Übertragungen	Übertragung von `p1JoinRequest`	-
Verlust der Nachricht `p1JoinDistribute`	> Maximale Anzahl der Übertragungen	Übergang in den Zustand `AgreeStateInit`/ `DistStateClInit`	Löschen der Verbindung beim nächsten Ereignis `EVENT_KEY_TIMEOUT`/ `EVENT_TM_TIMEOUT`/ `EVENT_KEY_UPDATE`
Verlust der Nachricht `p1JoinConfirm`	≤ Maximale Anzahl der Übertragungen	-	Übertragungen von `p1JoinDistribute`
Verlust der Nachricht `p1JoinConfirm`	> Maximale Anzahl der Übertragungen	-	Löschen der Verbindung beim nächsten Ereignis `EVENT_KEY_TIMEOUT`/ `EVENT_TM_TIMEOUT`/

			EVENT_KEY_UPDATE
Verlust der Nachricht p2Distribute	1	Erneute Gruppenanmeldung	-
Verlust der Nachricht p3UpdateDistribute, EVENT_KEY_TIMEOUT	1	Erneute Gruppenanmeldung	-
Verlust der Nachricht p3UpdateDistribute, Empfang einer Nachricht mit falscher Sequenznummer	1	Erneute Gruppenanmeldung	-
Verlust der Nachricht p3UpdateDistribute, EVENT_TM_TIMEOUT (nur Key Agreement)	1	Erneute Gruppenanmeldung	-
Verlust der Nachricht p3TMDistribute, Empfang einer Nachricht mit falscher Sequenznummer (nur Key Agreement)	1	Erneute Gruppenanmeldung	-

Tabelle 26: Tests zur Verifikation der Fehlertoleranzverfahren für temporäre Kommunikationsstörungen

8.2 Effizienzanalyse des Schlüsselmanagements MIKE

Die in Kapitel 6 definierten Metriken werden zur Durchführung der Effizienzanalyse eingesetzt. Diese lässt sich in die Bereiche theoretische und experimentelle Analyse unterteilen. Inhalt der theoretischen Analyse ist ein Vergleich beider Betriebsmodi untereinander und ein Vergleich der Komplexität der Protokolle mit der Komplexität von Protokollen, die aus der Literatur bekannt sind. Zur experimentellen Effizienzanalyse werden Simulationen und Messungen durchgeführt. Die Simulationen werden eingesetzt, um die Effizienz des Systems bei verschiedenen Netzwerkinfrastrukturen zu untersuchen. Das Ziel der experimentellen Analyse im praktischen Einsatz besteht darin, die Gültigkeit der in den Simulationen durchgeführten Abstraktionen zu beweisen.

8.2.1 Parameter der Effizienzanalyse

Die bei der Effizienzanalyse eingesetzten kryptographischen Algorithmen sowie die verwendeten Schlüssellängen haben einen entscheidenden Einfluss auf die Effizienz der Verfahren. Der Einsatz von größeren Schlüssellängen erfordert eine größere Rechenleistung, jedoch erhöht sich dadurch die Sicherheit der kryptographischen Verfahren. Die empfohlenen Schlüssellängen sind abhängig von dem angestrebten Sicherheitsniveau. Zur Effizienzanalyse der auf dem DH-Algorithmus basierenden Schlüsselbereitstellungsverfahren, z.B. MIKE – Key Agreement, wird der exponentielle (reguläre) DH-Algorithmus mit einer Schlüssellänge von 512 Bit eingesetzt. Für die Untersuchung der Verfahren zur Schlüsselbereitstellung auf Basis von symmetrischen Algorithmen, d.h. MIKE - Key Distribution, wird der Triple Data Encryption Standard (TDES, 3DES) mit 168 Bit Schlüssellänge verwendet. Bei digitalen

Signaturen, die zur Authentisierung der Managementnachrichten eingesetzt werden, werden die Algorithmen SHA-1 und RSA verwendet. Der Algorithmus RSA wird mit einer Schlüssellänge von 1024 Bit eingesetzt.

8.2.2 Theoretische Effizienzanalyse

Die theoretische Effizienzanalyse wurde unter dem Aspekt des Vergleichs beider Betriebsmodi durchgeführt. Diese Untersuchungen zur Komplexität basieren auf der in Tabelle 16 und Tabelle 17 zusammengefassten Komplexität der Protokolle bezüglich der Teilnehmerzahl. Das heißt, die Abhängigkeit der in Abschnitt 6.2 definierten und als einfachere Metriken bezeichneten Messwerte von der Teilnehmeranzahl wird analysiert. Die Abwicklung von Teilnehmeroperationen im Betriebsmodus Key Agreement benötigt mehr Rechenleistung als im Betriebsmodus Key Distribution. Bei beiden Modi ist die Komplexität der kryptographischen Operationen von der Größenordnung $O(\log_d(U))$ bei U Nutzern und der Verwendung eines Baums vom Grad d (Abbildung 95). Allerdings ermitteln im Modus Key Agreement die Nutzer den Gruppenschlüssel durch die iterative Anwendung des DH-Algorithmus. Dessen Ausführung ist rechenintensiver als die symmetrische Ver- bzw. Entschlüsselung, die im Modus Key Distribution zur Gruppenschlüsselermittlung eingesetzt wird (vgl. Tabelle 12). Weiterhin kennt im Modus Key Agreement jeder Nutzer die vollständige Baumstruktur und alle Blindschlüssel. Dies ist notwendig, damit jeder die Aufgaben des TM übernehmen kann und das System reparierbar ist. Nachteil dieses Ansatzes ist, dass beim Gruppeneintritt der Schlüsselbaum übertragen werden muss, d.h. die Komplexität der übertragenen Datenmenge ist O(U) (Abbildung 95). Wegen dieser Komplexität des Modus Key Agreement wurde im Konzept MIKE der Betriebsmodus Key Distribution als alternatives Verfahren zur Gruppenschlüsselbereitstellung integriert, falls die zu verwaltende Gruppe einem großen Teilnehmerzuwachs unterliegt oder Teilnehmer mit einer geringen Datenübertragungskapazität der Gruppe beitreten.

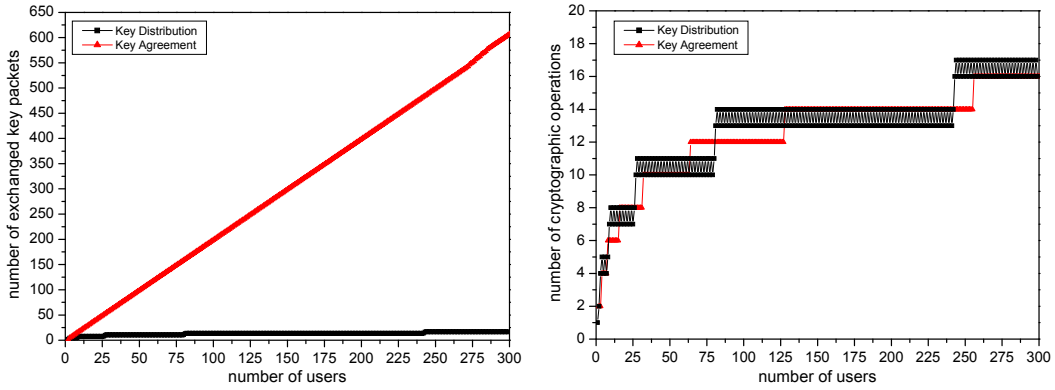

Abbildung 95: Vergleich der beiden Betriebsmodi bei der Teilnehmeroperation LEAVE

8.2.3 Simulationsumgebung für die Effizienzanalyse

Um eine Effizienzanalyse des Schlüsselmanagements MIKE für verschiedene Kommunikationsinfrastrukturen durchzuführen, wird der Netzwerksimulator ns-2 [Isi06] eingesetzt. Bei diesem handelt es sich um einen diskreten, ereignisgesteuerten Simulator, der in der Programmiersprache C++ geschrieben ist. Zur Beschreibung des Simulationsablaufs

wird die Programmiersprache OTcL eingesetzt. Im Simulator ist ein MIKE-Prozess in drei Schichten unterteilt (Abbildung 96).

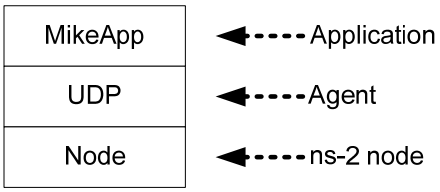

Abbildung 96: Schlüsselmanagement MIKE als Bestandteil der ns-2-Architektur

Die als `Node` bezeichnete Schicht wird verwendet, um im Simulator die Schichten eins und zwei des OSI-Modells nachzubilden. Der auf den Internetprotokollen IP und UDP basierende Nachrichtenaustausch wird durch so genannte Agenten realisiert. Für die Simulation wurde der vorhandene Agent `UDP` für die Verwendung in Kombination mit der Applikation `MikeApp` angepasst. Diese simuliert das Schlüsselmanagement MIKE.

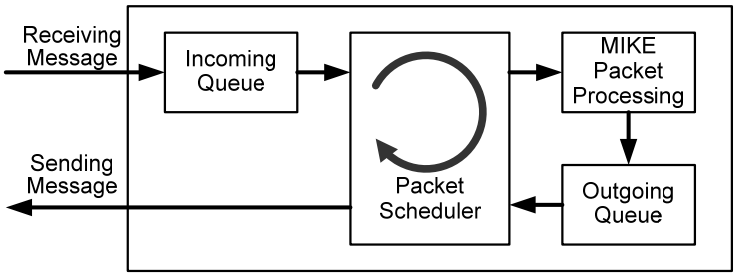

Abbildung 97: Abstraktion des Schlüsselmanagements MIKE zur Simulation

Für den Einsatz des Schlüsselmanagements MIKE in dem Simulator werden Abstraktionen durchgeführt. Diese Vereinfachungen haben das Ziel, rechenleistungsintensive kryptographische Operationen zu umgehen und somit die Laufzeit einer Simulation zu reduzieren bzw. eine Simulation mit großen Nutzerzahlen zu ermöglichen. Die Signierung der Schlüsselmanagementnachrichten ist ein Beispiel für eine derartige Vereinfachung. Um dennoch die für die kryptographischen Operationen benötigte Verarbeitungszeit beim Nachrichtenempfang bzw. Nachrichtenversand zu berücksichtigen, werden die zwei Warteschlangen `IncomingQueue` und `OutgoingQueue` eingeführt (Abbildung 97). Das Entleeren dieser Warteschlangen übernimmt das Modul `PacketScheduler`. Dieser entnimmt eine Nachricht erst dann aus der Warteschlange, wenn diese für die berechnete Verarbeitungszeit dort abgelegt war. Eingehende Nachrichten werden von der Funktion `PacketScheduler` dem Schlüsselmanagement MIKE zur Verarbeitung, ausgehende Nachrichten dem Agenten `UDP` zum Versand übergeben.

Kryptographische Operation	Dauer der Berechnung
Signaturberechnung	0,0040 s
Signaturprüfung	0,0015 s
Schlüsselberechnung (DH-Algorithmus)	0,0020 s
Blindschlüsselberechnung (DH-Algorithmus)	0,0030 s
Ver-/Entschlüsselung	0,000007 s

Tabelle 27: Parameter der kryptographischen Operationen

In Tabelle 27 sind die in der Simulation für die kryptographischen Operationen benötigten Verarbeitungszeiten zusammengefasst. Diese Zeiten sind von der verfügbaren Rechenleistung des Rechners, auf dem das Schlüsselmanagement betrieben wird, abhängig. Um mit der Effizienzanalyse im praktischen Einsatz vergleichbare Resultate zu erzielen, wurden die Verarbeitungszeiten vorher mit einem Testprogramm bestimmt.

8.2.4 Netzwerktopologie bei der Effizienzanalyse durch Simulation

Wie bereits erwähnt, werden die Simulationen dazu verwendet, das Verhalten des Schlüsselmanagements in Netzwerken mit unterschiedlichen Kommunikationsinfrastrukturen zu untersuchen. Hierbei werden die als Ethernet und VHF bezeichneten Infrastrukturen verwendet. Diese wurden bei der Festlegung der Anforderung in Abschnitt 3.3 als günstigste bzw. ungünstigste anzunehmende Kommunikationsinfrastruktur identifiziert. In [Ebe03] wird dargestellt, dass zukünftig Software Defined Radios (SDR) zur Datenübertragung eingesetzt werden. Dieses sind durch Software konfigurierbare Funkgeräte. Die Konfiguration eines SDR wird als Wellenform bezeichnet. Aus diesem Grund wird eine Kommunikation aus diesen Geräten ebenfalls bei der Effizienzanalyse durch Simulation berücksichtigt und als SDR bezeichnet. In Abbildung 98, links sind die bei der Effizienzanalyse durch Simulation eingesetzte Netzwerktopologie sowie deren Parameter für die Netzwerkinfrastruktur Ethernet, VHF und SDR graphisch dargestellt. Es wird angenommen, dass zur Konfiguration des SDR die Breitbandwellenform verwendet wird.

In [Ami02] wurde die Effizienz von Schlüsselmanagementverfahren beim Betrieb über Weitverkehrsnetze, im Folgenden als Wide Area Network (WAN) bezeichnet, untersucht. Hierbei wurde die in Abbildung 98, rechts dargestellte Netwerktopologie, Datenübertragungsraten und Verzögerungszeiten verwendet. In Abschnitt 3.3 wurde dargelegt, dass das im Rahmen dieser Arbeit entwickelte Schlüsselmanagement die spezifischen Anforderungen, die beim Auslandseinsatz von Streitkräften im Rahmen von Konfliktverhütung und Krisenbewältigung auftreten, erfüllen soll. Bei derartigen Einsätzen findet neben der Kommunikation im Einsatzgebiet auch eine Kommunikation über Weitverkehrsnetz mit den Kommandostellen der Heimat statt. Mit dem in [Ami02] definierten Netzwerktopologie werden deshalb der Einsatz eines Schlüsselmanagement in einer derartigen Kommunikationsinfrastruktur untersucht.

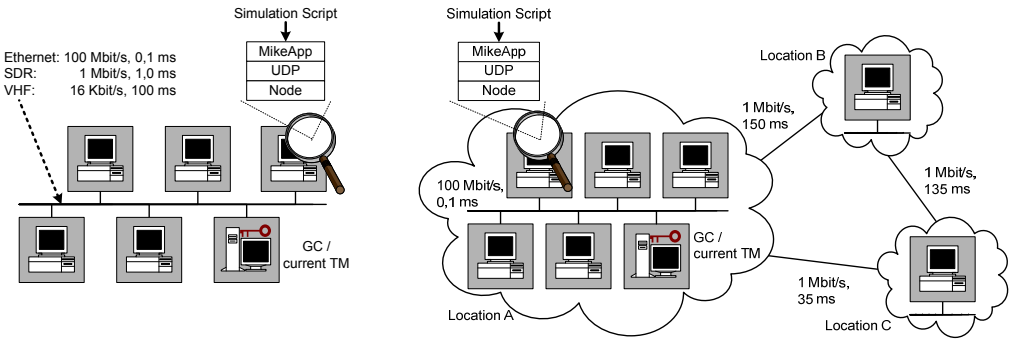

Abbildung 98: Netzwerktopologien bei der Effizienzanalyse durch Simulation

Für diese Untersuchung wird das in Abschnitt 6.3 modellierte synthetische Nutzerverhalten zur Erzeugung einer Last verwendet. Da hierbei keine Kollisionen auf dem Netzwerk

auftreten, kann die Kommunikationsinfrastruktur durch die Verwendung des ns-2-Konzepts `Link` abstrahiert werden. Hierbei werden Verbindungen durch eine Datenübertragungsrate, eine Signallaufzeit sowie eine Warteschlange nach dem Prinzip FIFO abstrahiert. Weiterhin wird angenommen, dass die Verbindungen fehlerfrei sind.

8.2.5 Ergebnisse der Effizienzanalyse durch Simulation

Begonnen wird die Darstellung der Ergebnisse der Effizienzanalyse durch Simulation mit den Ergebnissen der Analyse des Modus Key Agreement in Netzwerktopologien, die im vorherigen Abschnitt vorgestellt wurden. Anschießend werden die Ergebnisse für den Betriebsmodus Key Distribution präsentiert.

Die gesamte Zeitdauer für den Gruppenbeitritt bzw. Gruppenaustritt (schwarze Quadrate, JOIN/LEAVE delay) und die Schlüsselwechselzeitdauer (grüne Dreiecke, p3UpdateDistribute+p3TMDistribute delay) im Modus Key Agreement ist sowohl für den Nachrichtenaustausch mittels Ethernet (oben) als auch für VHF (unten) in Abbildung 99 dargestellt. Zusätzlich ist die Zeit für die Übertragung und Verarbeitung der Nachricht `p3UpdateDistribute` (rote Kreise) in der Abbildung enthalten. Die beim Schlüsselwechsel übertragene Datenmenge ergibt sich aus der Summe der mit den beiden Nachrichten `p3TMDistribute` und `p3UpdateDistribute` übertragenen Daten. In Abbildung 99 (oben) sind die Nachrichtengrößen mit blauen Kreuzen bzw. Sternen ebenfalls eingetragen.

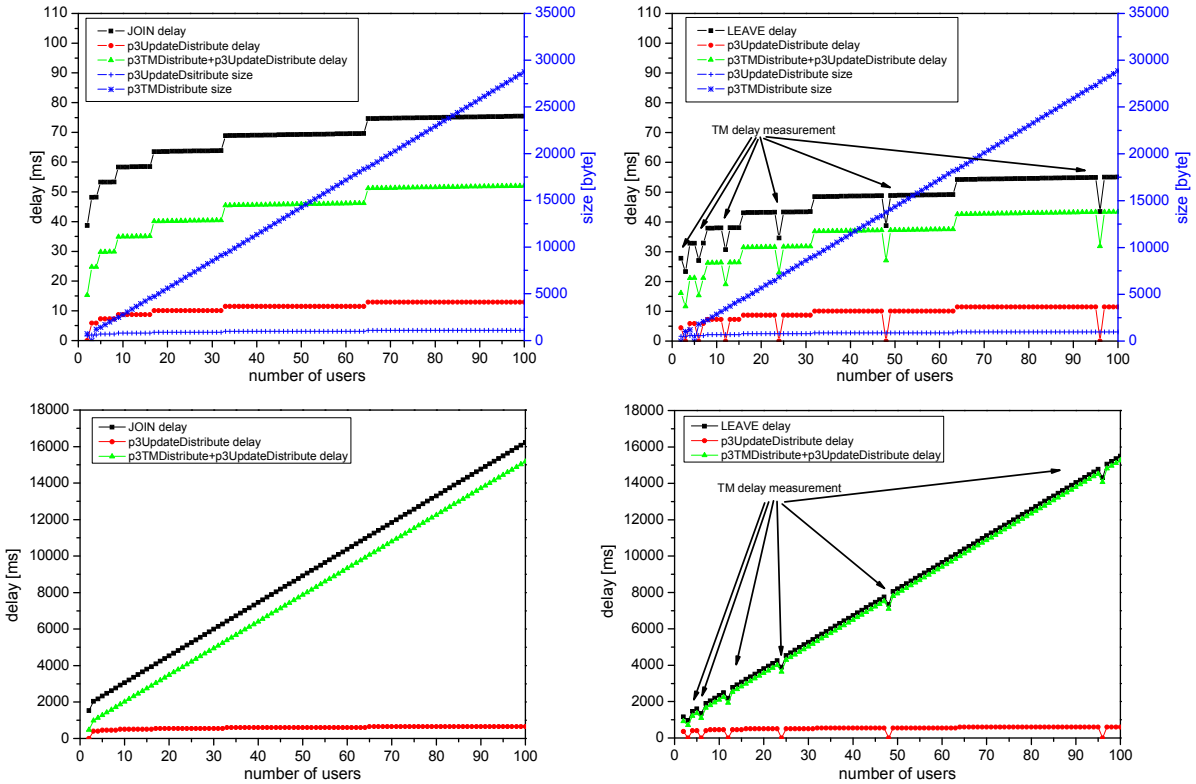

Abbildung 99: Simulierte Zeitdauer für den Gruppenbeitritt (links) und den Gruppenaustritt (rechts) im Modus Key Agreement bei Ethernet (oben) und VHF (unten)

Zuerst werden die durch die Simulation ermittelten Zeiten für den Schlüsselwechsel erläutert. Diese sind im Modus Key Agreement durch die Übertragung und Verarbeitung der beiden Nachrichten `p3TMDistribute` und `p3UpdateDistribute` bestimmt (vgl. Abschnitt 6.1). Die mit den beiden Nachrichten übertragene Datenmenge für den Schlüsselwechsel steigt bei linear zunehmender Teilnehmerzahl ebenfalls linear. Wird das Übertragungsmedium Ethernet zur Übermittlung der Managementnachrichten verwendet, zeigt die Schlüsselwechselzeitdauer nach der Teilnehmeroperation JOIN einen logarithmischen Anstieg mit der Gruppengröße. Wird der Schlüsselbaum jeweils um eine Ebene erweitert, muss der TM eine zusätzliche kryptographische Operation zur Berechnung des Gruppenschlüssels durchführen. Diese zusätzlichen kryptographischen Operationen sind als Sprünge in der Kurve sichtbar und finden jeweils bei einer Potenz des Knotengrads d=2 statt. Bei der Teilnehmeroperation LEAVE kann ein ähnliches Verhalten beobachtet werden. Die nach einer Teilnehmeroperation LEAVE übertragene Datenmenge für den Schlüsselwechsel steigt bei linear steigender Teilnehmerzahl ebenfalls linear und nicht wie erwartet logarithmisch (vgl. Tabelle 16). Grund hierfür ist, dass in der Implementierung, mit der die Messungen durchgeführt wurden, nach jeder Teilnehmeroperation der vollständige Schlüsselbaum übertragen wird. In einer realen Implementierung würde man nach einer Teilnehmeroperation LEAVE nur die Änderungen im Schlüsselbaum übertragen, um so nur noch logarithmischen Übertragungsaufwand zu haben. Untersucht man die Zeiten für die Übertragung und Verarbeitung der Nachricht `p3UpdateDistribute` nach einer Teilnehmeroperation LEAVE genauer, so stellt man fest, dass diese bei manchen Teilnehmerzahlen null beträgt (vgl. Abbildung 99, rechts). Ein MIKE-Prozess ermittelt für die Übertragungs- und Verarbeitungszeit der Nachricht `p3UpdateDistribute` den Wert null, falls er Absender der Nachricht, d.h. TM, ist. Sind die Prozesse des Schlüsselmanagements im Betriebsmodus Key Agreement über VHF verbunden anstatt über Ethernet, bestimmt die Übertragungsdauer der Datenmenge für den Schlüsselwechsel dessen Zeitdauer. Da diese Datenmenge linear steigt, falls die Nutzeranzahl linear zunimmt, vergrößert sich die Schlüsselwechseldauer ebenfalls linear und die Anzahl der kryptographischen Operationen spielt keine Rolle.

Nachfolgend wird die gesamte Zeitdauer für die Durchführung einer Teilnehmeroperation JOIN bzw. LEAVE im Betriebsmodus Key Agreement betrachtet. Der Vergleich zwischen der gesamten Zeitdauer für den Gruppenbeitritt und den Gruppenaustritt zeigt, dass die Gruppenbeitrittsdauer größer als die Gruppenaustrittsdauer ist. Dies wird durch den 3-Wege-Anmeldevorgang zum Schutz gegen Angriffe durch wiederholtes Senden und die Initialisierung des Schlüsselwechsels mit der Nachricht `p2UpdateDistribute` hervorgerufen.

In Abbildung 100 ist die Zeitdauer der Blockierung des Schlüsselmanagements bei der Teilnehmeroperation JOIN dargestellt. Die Systemblockierung resultiert aus dem Wechsel des TM und ergibt sich aus der Übertragungs- und Verarbeitungszeit der Nachricht `p3TMDistribute` (vgl. Abschnitt 6.1), d.h. aus der Differenz der in Abbildung 99 mit grünen Dreiecken und roten Kreisen dargestellten Kurven. Erfolgt der Nachrichtenaustausch des Schlüsselmanagements über VHF, so vergrößert sich die Zeit, in der das System keine Nutzeranfragen beantworten kann, bei linear steigender Nutzerzahl ebenfalls linear. In diesem

Fall bestimmt die Übertragungsdauer der Nachricht `p3TMDistribute` die Dauer der Systemblockierung. Steht dem Schlüsselmanagement eine ausreichende Datenübertragungskapazität zur Verfügung, bestimmt die Aktualisierung des Schlüsselbaums die Blockierungszeit. In diesem Fall steigt die Systemblockierungszeitdauer logarithmisch, da die Anzahl der kryptographischen Operationen logarithmisch wächst. Wird im Modus Key Agreement die in Abschnitt 7.3 vorgeschlagene Sammelverarbeitung von Nutzeranfragen durchgeführt, so sind mehrere Wechsel des Transaction Managers zur Beantwortung einer Nutzeranfrage notwendig. Deshalb findet durch die Sammelverarbeitung zwar eine effiziente Verarbeitung von Nutzeranfragen statt, allerdings steigt auch die Zeit der Systemblockierung.

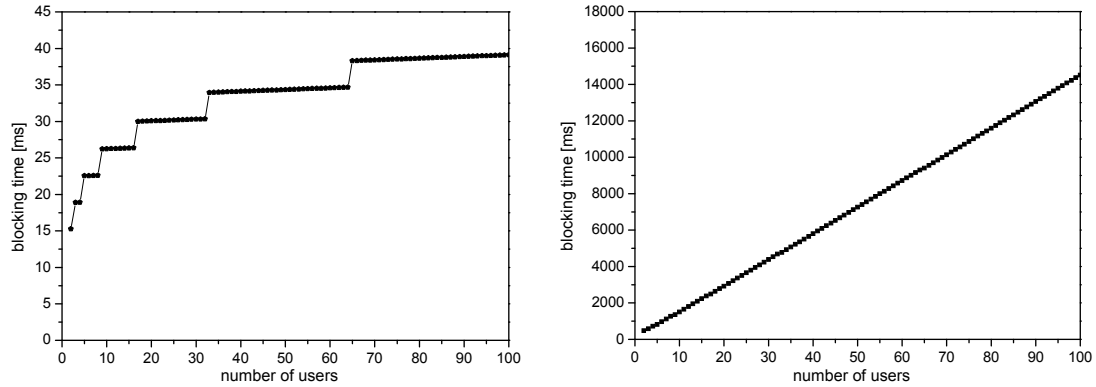

Abbildung 100: Simulierte Blockierungszeitdauer durch den Wechsel des Transaction Managers im Modus Key Agreement bei Ethernet (links) und VHF (rechts)

Motivation für die Einführung des Betriebsmodus Key Agreement in das Konzept MIKE war die Schlüsselbereitstellung ohne Single Point of Failure. Die Analyse der Effizienz des Betriebsmodus Key Agreement zeigt, dass auf Grund der beim Schlüsselwechsel übertragenen Datenmenge, der zur Gruppenschlüsselberechnung benötigten Rechenleistung und der mit steigender Teilnehmerzahl zunehmenden Systemblockierung insbesondere in Netzwerken mit geringer Datenübertragungsrate ein alternatives effizienteres Schlüsselbereitstellungsverfahren vorhanden sein muss. Deshalb wurde in das Konzept MIKE der Betriebsmodus Key Distribution integriert.

In Abbildung 101 ist die gesamte Zeit für die Durchführung eines Gruppenbeitritts bzw. Gruppenaustritts (schwarze Quadrate, JOIN/LEAVE delay), die Schlüsselwechselzeitdauer (rote Kreise, p3UpdateDistribute delay) sowie die Datenmenge für den Schlüsselwechsel (blaue Striche) im Modus Key Distribution bei Ethernet (oben) und VHF (unten) dargestellt. Der Abbildung 101 (oben) kann man entnehmen, dass bei linear steigender Teilnehmerzahl die Größe der Nachricht `p3UpdateDistribute`, d.h. die übertragene Datenmenge für den Schlüsselwechsel, logarithmisch steigt. Sprünge finden bei einer Gruppengröße statt, die einer Potenz des Schlüsselbaumgrads d=3 entsprechen. In diesem Fall wird der Schlüsselbaum um eine Ebene erweitert und die Datenmenge für den Schlüsselwechsel vergrößert sich. Die Schlüsselwechselzeitdauer im Modus Key Distribution ist durch den Austausch und die Verarbeitung der Nachricht `p3UpdateDistribute` bestimmt (vgl. Abschnitt 6.1). Beim Einsatz des Schlüsselmanagements in Kombination mit dem Übertragungsmedium Ethernet bewirkt der logarithmische Anstieg der Datenmenge für den Schlüsselwechsel auf Grund der großen verfügbaren Datenübertragungsrate des Übertragungsmediums nur eine sehr geringe

Erhöhung der Schlüsselwechselzeitdauer. Kommunizieren die MIKE-Prozesse hingegen über VHF, findet eine deutliche Beeinflussung der Schlüsselwechselzeitdauer durch die Größe der Datenmenge des Schlüsselwechsels statt. Diese weist dann an den Stellen Sprünge auf, an denen die übertragene Datenmenge ebenfalls sprunghaft steigt. Auf Grund der geringen benötigten Rechenleistung der kryptographischen Operationen zeigt die Verarbeitungszeit der Nachricht `p3UpdateDistribute` keinen messbaren Einfluss auf die Schlüsselwechseldauer.

Analysiert man die Gesamtzeit für die Durchführung einer Teilnehmeroperation, so stellt man auch beim Modus Key Distribution fest, dass diese für die Operation JOIN größer ist als für die Teilnehmeroperation LEAVE. Ursache hierfür ist der beim Gruppenbeitritt durchgeführte 3-Wege-Anmeldevorgang zum Schutz gegen Angriffe durch wiederholtes Senden. Weiterhin wird im Rahmen des Gruppenbeitritts die Nachricht `p2UpdateDistribute` ausgetauscht, die den Schlüsselwechsel initialisiert. Im linken Teil der Abbildung 101 erkennt man, dass die Gesamtdauer für die Teilnehmeroperation JOIN fast linear mit der Gruppengröße ansteigt. Dieser Anstieg wird durch die Übertragung der Informationen zur Konfiguration des IPSec-Nutzdatenschutzes hervorgerufen. Wird das Schlüsselmanagement in Kombination mit VHF eingesetzt, führt die Übertragung dieser Information zu einer erheblichen Verzögerung der gesamten Bearbeitungszeit der Operation JOIN. Die IPSec-Konfigurationsinformationen wurden in dem in [Har03] spezifizierten Format übertragen. Dadurch enthalten sie erhebliche Redundanzen. Durch die Verwendung eines effizienteren Formats zur Übertragung der IPSec-Konfiguration kann die Zeitdauer für den Gruppenbeitritt vermindert werden.

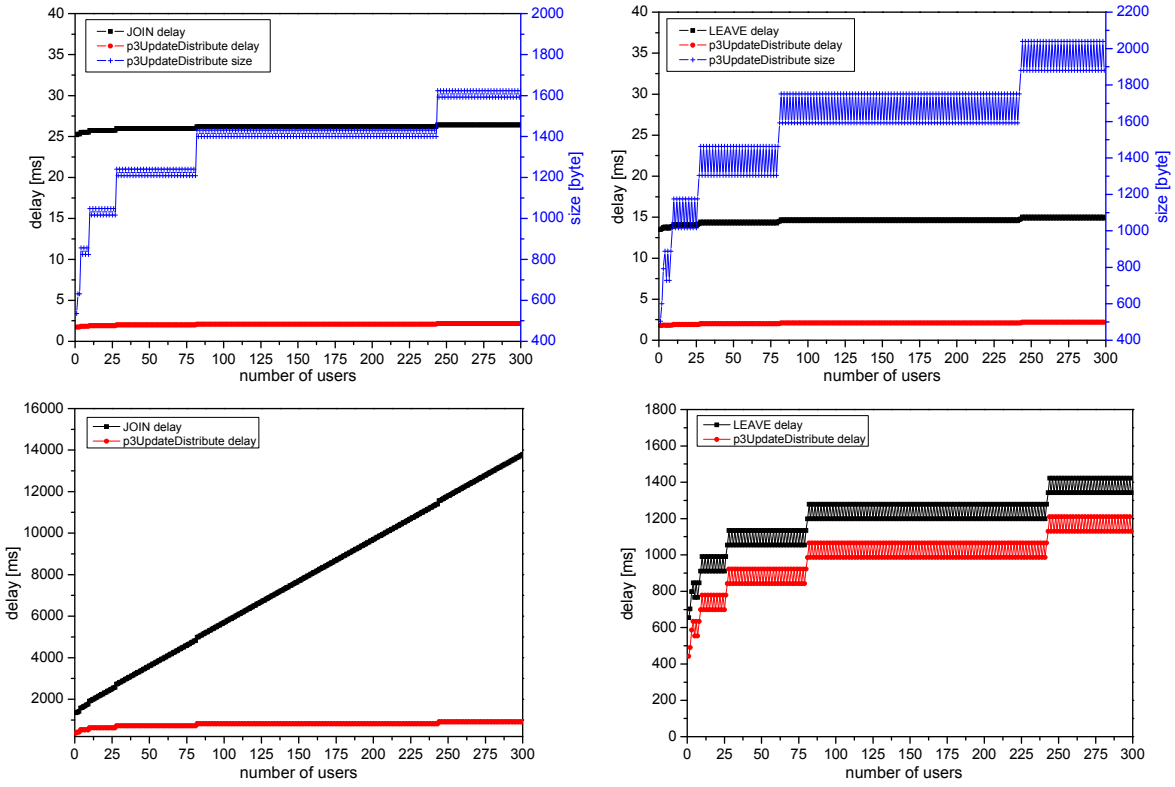

Abbildung 101: Simulierte Zeitdauer für den Gruppenbeitritt (links) und den Gruppenaustritt (rechts) im Modus Key Distribution bei Ethernet (oben) und VHF (unten)

Um die Einsatzfähigkeit des Schlüsselmanagements MIKE in heterogenen Netzwerken zu bewerten, ist die Schlüsselwechselzeitdauer in der günstigsten bzw. ungünstigsten anzunehmenden Kommunikationsinfrastruktur zu vergleichen. Soll der Gruppenschlüssel zum Schutz einer paketbasierten Sprachkommunikation verwendet werden, so muss die Differenz der Schlüsselwechselzeitdauer beim Betrieb im günstigsten (Ethernet) bzw. ungünstigsten Infrastruktur (VHF) kleiner als 600 ms sein. Dieser Grenzwert lässt sich für die in [Col99] spezifizierten Vocoder abschätzen. Derartige Vocoder kodieren digitalisierte Sprache mit einer Rate von 1,2 Kbit/s und besitzen eine Fehlertoleranz von drei aufeinander folgenden Sprachpaketen. Bei einer größeren Differenz zwischen der Schlüsselwechseldauer können Sprachpakete auf Grund eines unterschiedlichen Gruppenschlüssels nicht richtig interpretiert werden. In diesem Fall treten Störungen der Sprachübertragung auf. Die Durchführung eines Schlüsselwechsels im Betriebsmodus Key Agreement in heterogenen Netzwerken ist ohne Störung der Kommunikation nicht möglich. Im Betriebsmodus Key Distribution kann die Teilnehmeroperation JOIN bis zu einer Gruppengröße von 27 Teilnehmern und die Teilnehmeroperation LEAVE bis zu einer Gruppengröße von 8 Teilnehmern durchgeführt werden, ohne dass ein Nutzer eine Störung der Anwendung paketbasierte Sprachkommunikation feststellen kann. Die störungsfreie Durchführung einer Teilnehmeroperation JOIN ist in größeren Gruppen möglich, da zur Aufnahme eines neuen Nutzers im Betriebsmodus Key Distribution eine geringere Datenmenge für den Schlüsselwechsel übertragen werden muss.

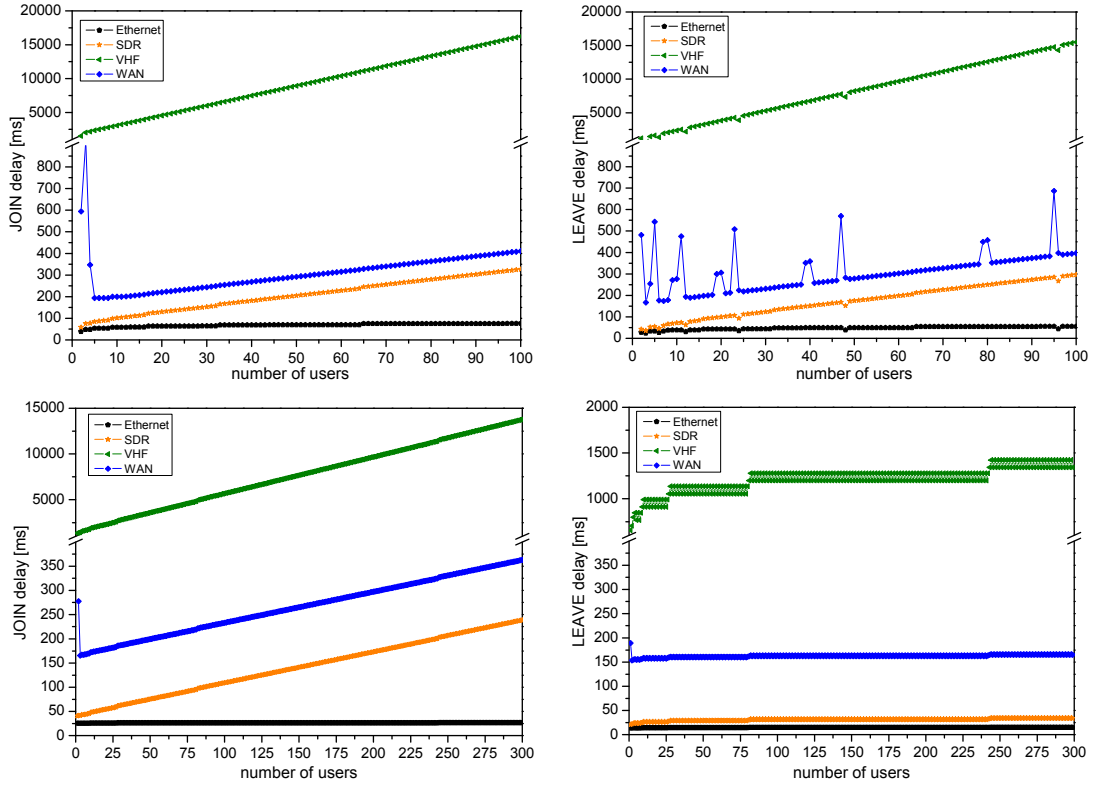

Abbildung 102: Simulierte Zeitdauer für den Gruppenbeitritt (links) bzw. den Gruppenaustritt (rechts) im Modus Key Agreement (oben) und im Modus Key Distribution (unten) bei unterschiedlicher Kommunikationsinfrastruktur

Einen Überblick über die Zeiten für den Gruppenbeitritt bzw. den Gruppenaustritt in den untersuchten Kommunikationsinfrastrukturen vermittelt Abbildung 102. Nachfolgend werden die Verzögerungszeiten des Betriebsmodus Key Agreement beim Betrieb im WAN detaillierter erläutert. In Abbildung 102, oben ist die simulierte Zeitdauer für die Durchführung einer Teilnehmeroperation JOIN bzw. LEAVE durch blaue Quadrate dargestellt. Die Verzögerungszeiten wurden am Standort B ermittelt (vgl. Abbildung 98). Extremwerte in den Verzögerungszeiten treten im Modus Key Agreement auf, wenn ein Nutzer der Standorte B bzw. C Transaction Manager wird. Dieses ist beim Gruppenbeitritt der Nutzer der Standorte B bzw. C der Fall. Außerdem tritt der Fall auf, wenn ein den Nutzern der Standort B bzw. C im Schlüsselbaum benachbarter Nutzer die Gruppe verlässt. Diese Extremwerte können aber verhindert werden, in dem zusätzlich der in Abschnitt 7.5 definierte Algorithmus zur ressourcengesteuerten Auswahl des TM verwendet wird. Die simulierte Zeitdauer für die Durchführung einer Teilnehmeroperation LEAVE ist in Abbildung 103, rechts durch schwarze Quadrate (LEAVE delay) dargestellt. Weiterhin ist in der Abbildung die Schlüsselwechseldauer (grüne Dreiecke, p3TMDisrtribute+p3UpdateDistribute delay) zu entnehmen.

Wie bereits erläutert, war die Schlüsselbereitstellung ohne Single Point of Failure die Motivation für die Einführung des Betriebsmodus Key Agreement in das Konzept MIKE. Der Nachteil des Betriebsmodus besteht darin, dass er bei steigender Gruppengröße nicht skaliert, d.h. der Aufwand für den Schlüsselwechsel steigt linear mit der Gruppengröße. Aus diesem Grund wurde der zweite Betriebsmodus in das Schlüsselmanagement integriert. Auf ihn kann umgeschaltet werden, so dass der Aufwand für den Schlüsselwechsel nur noch logarithmisch mit der Gruppengröße steigt. In Abbildung 103, links ist die gesamte Zeit für die Durchführung eines Gruppenbeitritts (schwarze Quadrate, JOIN delay), die Schlüsselwechselzeitdauer (grüne Dreiecke bzw. rote Kreise) dargestellt. Die Betriebsmodusumschaltung erfolgte bei einer Gruppengröße von 20 Teilnehmern. Der Abbildung entnimmt man, dass durch Betriebsmodusumschaltung das Konzept MIKE auch in großen Gruppen eine Schlüsselbereitstellung ermöglicht.

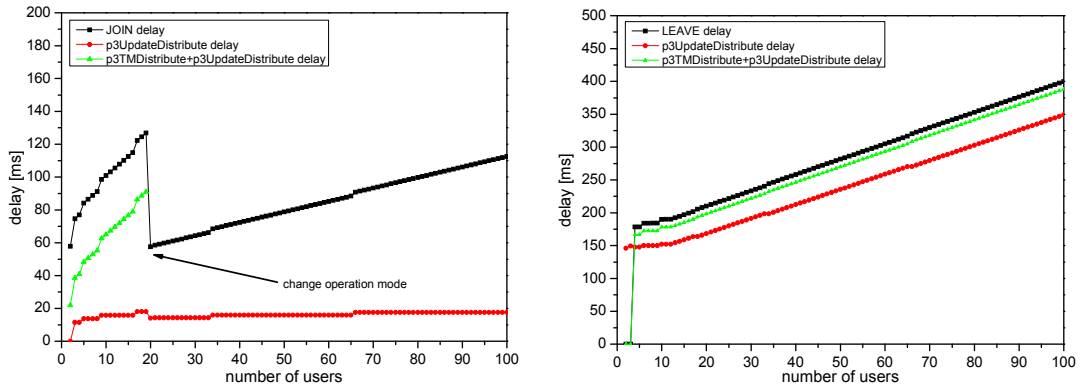

Abbildung 103: Simulierte Zeitdauer für den Gruppenbeitritt mit Betriebsmoduswechsel bei SDR (links) und für den Gruppenaustritt mit einer ressourcengesteuerten Auswahl des TM im WAN (rechts)

8.2.6 Messverfahren der experimentellen Effizienzanalyse im praktischen Einsatz

In diesem Abschnitt wird das Messverfahren für die experimentelle Effizienzanalyse im praktischen Einsatz vorgestellt. Eine Messung in einem verteilten System kann auf zwei verschiedene Arten durchgeführt werden:

- Passiv
  Bei dieser Art der Messung werden die Messwerte durch die Auswertung des Netzwerkverkehrs ermittelt.
- Aktiv
  Die Messwerte werden durch die Auswertung speziell für die Messung erzeugter Nachrichten bzw. Protokollelemente ermittelt.

Zur Bestimmung der in Abschnitt 6.1 definierten Messwerte wird eine aktive Messung durchgeführt. Grund hierfür ist die Granularität, die eine aktive Messung bietet. Durch eine aktive Messmethode ist es möglich, Start und Ende des Messzeitraums sehr flexibel festzulegen.

Für eine aktive Messung im praktischen Einsatz muss ein Messsystem in die Implementierung MIKE integriert werden. Dieses bestimmt Paketgrößen und Einwegverzögerung gemäß den in Abschnitt 6.1 definierten Metriken. Zur Messung der Einwegverzögerung wertet das Messsystem ein zusätzlich in die ausgetauschten Nachrichten eingeführtes Protokollelement aus (Abbildung 104). Format und Identifizierungsnummer sind gemäß RFC 2408 festgelegt. Als Zeitstempel werden die vier Bytes der Coordinated Universal Time (UTC) in Mikrosekunden verwendet. Mit dem Messsystem können Verzögerungszeiten bis zu einer halben Stunde gemessen werden. Allerdings werden durch die aktive Messung die Messergebnisse beeinflusst. Diese wird im Rahmen der Messarchitekturvorstellung in Abschnitt 8.2.7 abgeschätzt.

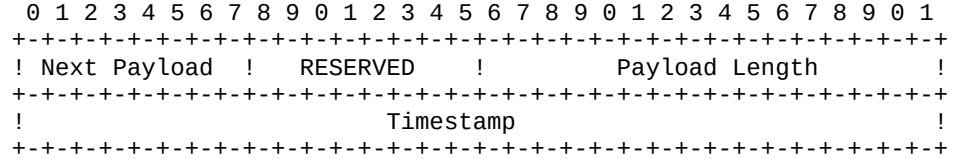

Abbildung 104: Protokollelement zur Übertragung eines Zeitstempels

8.2.7 Messaufbau der experimentellen Effizienzanalyse im praktischen Einsatz

Für die Analyse im praktischen Einsatz wurden Messungen gemäß der in Abschnitt 6.1 definierten Metrik in einem drahtgebundenen Netzwerk durchgeführt. Das Testnetzwerk basiert auf dem Übertragungsmedium Ethernet. Der eingesetzte Messaufbau, d.h. die Messarchitektur sowie die Topologie des Testnetzwerks sind in Abbildung 105 dargestellt und werden im Folgenden erläutert. Zur Erzeugung einer Last werden die in Abschnitt 6.3 modellierten drei Arten des Nutzerverhaltens verwendet. Die vom Nutzerverhalten vorgesehene Anzahl an Rechnern stand für die Messungen nicht zur Verfügung. Aus diesem Grund wird ein Rechner als Lastcenter eingesetzt und repräsentiert eine größere Anzahl von Rechnern mit MIKE-Prozessen [Gin06]. Damit der Rechner diese Aufgabe übernehmen kann,

werden der Netzwerkschnittstelle die IP-Adressen aller MIKE-Prozesse, die das Lastcenter repräsentiert, zugewiesen und die erforderliche Anzahl von MIKE-Prozessen ausgeführt. Auf Grund der hohen Rechenleistung beim gleichzeitigen Empfang von Multicast-Nachrichten der MIKE-Prozesse des Lastcenters können auf diesem keine brauchbaren Messungen durchgeführt werden. Die Messungen übernehmen deshalb die als Monitore bezeichneten Rechner. Auf diesen wird jeweils nur ein MIKE-Prozess ausgeführt. Als Monitore werden fünf Laptops mit dem Betriebssystem Linux 2.6.17.8 verwendet. Diese verfügen über Prozessoren mit einer Taktfrequenz von 1.2 GHz bis 2.0 GHz. Bei den Messungen unterscheidet sich der Messaufbau für die beiden Betriebsmodi. Im Modus Key Agreement muss außerdem sichergestellt werden, dass der Wechsel des TM zwischen zwei Rechnern erfolgt, auf denen nur ein MIKE-Prozess ausgeführt wird (Abbildung 105, rechts). Ein Wechsel des TMs in das Lastcenter bzw. innerhalb des Lastcenters darf nicht erfolgen, da dies zu einer erheblichen Verfälschung der Messung führen würde. Auf Grund der Tatsache, dass dies nur für das synthetische Nutzerverhalten sichergestellt werden kann, werden im Modus Key Agreement nur Messungen unter dessen Einfluss durchgeführt. Für die Messungen im Modus Key Distribution wird ein Rechner als GC konfiguriert (Abbildung 105, links).

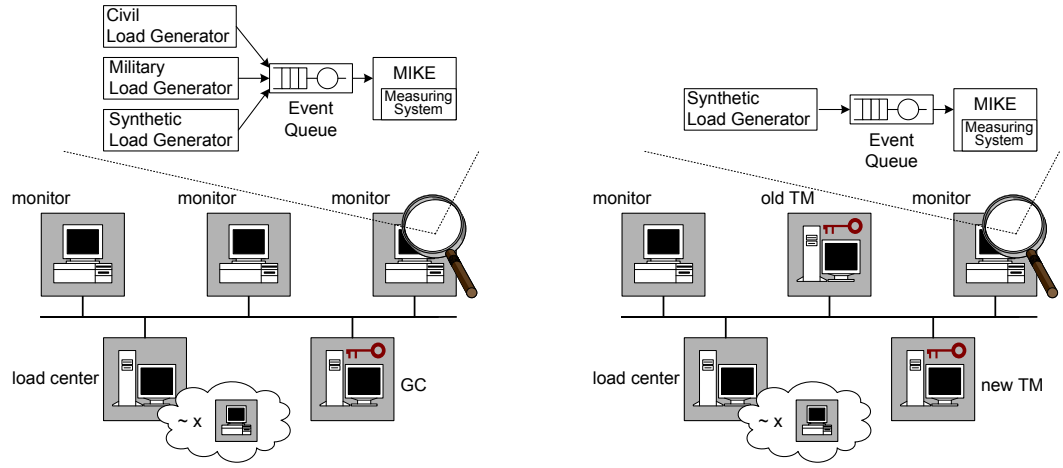

Abbildung 105: Messaufbau bei der Effizienzanalyse für den Modus Key Distribution (links) und Key Agreement (rechts) im praktischen Einsatz

Zur Durchführung der Messungen ist auch die Messung der Einwegverzögerung von Nachrichten erforderlich. Um diese ermitteln zu können, werden die Uhren der an der Messung beteiligten Rechner synchronisiert. Die bei dieser Messung auftretenden Messfehler werden im Folgenden abgeschätzt:

- Messfehler durch die ungenaue Zeitsynchronisation
  Eine wichtige Voraussetzung zur Messung von Einwegverzögerungen ist die Zeitsynchronisation der beteiligten Rechner. Im idealen Fall sollte der Unterschied zwischen den zwei an der Messung beteiligten Rechneruhren Clock-1 und Clock-2 null sein. Tatsächlich gilt aber für die zum selben Zeitpunkt gemessenen Zeitwerte der beiden Uhren $t_{Clock\text{-}2}=t_{Clock\text{-}1}+\sigma*t_{Clock\text{-}1}+\omega_{t'}$. Wobei $\omega_{t'}$ die Abweichung der Uhren zum Zeitpunkt t' und σ den Skew bezeichnet. Der Skew einer Uhr drückt den Frequenzunterschied zur fiktiven Uhr der tatsächlichen Zeit nach der Coordinated Universal Time (UTC) aus. Seine Änderung im Laufe der Zeit wird Drift genannt. Zur Zeitsynchronisation wird das

Network Time Protocol eingesetzt [Mil92]. Für die im Ethernet hiermit erzielte Genauigkeit der Zeitsynchronisation wurde 1ms ermittelt [Smo02]. Messungen in dem Testnetzwerk zeigen, dass die Genauigkeit besser als 1ms ist [Gin06].

- Messfehler bei der Bestimmung der Zeitwerte
  Theoretisch können mit dem Aufruf `gettimeofday()` Zeitwerte mit einer Genauigkeit im Mikrosekundenbereich ermittelt werden. Allerdings ist es möglich, dass vor Beginn der Zeitmessung der MIKE-Prozess durch die Prozessverwaltung des Betriebssystems unterbrochen wird. Durch eine geringe Auslastung der Rechner wird dieser Einfluss gering gehalten.
- Messfehler durch die Übertragung zusätzlicher Protokollelemente
  Durch die aktive Messung werden den Nachrichten zusätzliche Protokollelemente hinzugefügt. Zu Messzwecken werden maximal zwei Zeitstempel, d.h. 128 Bit, übertragen. Eine Berechnung der Verzögerungszeit im Ethernet durch die zusätzlichen Daten ergibt 0,001ms.

Der Messfehler durch die ungenaue Zeitsynchronisation besitzt den größten Einfluss auf die Messwerte. Es wird die Genauigkeit der Messung deshalb insgesamt auf 1ms abgeschätzt.

8.2.8 Ergebnisse der experimentellen Effizienzanalyse im praktischen Einsatz

Die auf einem Monitor gemessene gesamte Zeitdauer für die Durchführung der Teilnehmeroperation JOIN bzw. LEAVE (schwarze Quadrate, JOIN/LEAVE delay) und die Schlüsselwechselzeitdauer (grüne Dreiecke, p3TMDistribute+p3UpdateDistribute delay) im Modus Key Agreement ist in Abbildung 106 dargestellt. Zusätzlich wurde in der Abbildung die Übertragungs- und Verarbeitungsdauer der Nachricht `p3UpdateDistribute` dargestellt (rote Kreise). Die dargestellte Messung wurde unter dem Einfluss des synthetischen Nutzerverhaltens durchgeführt.

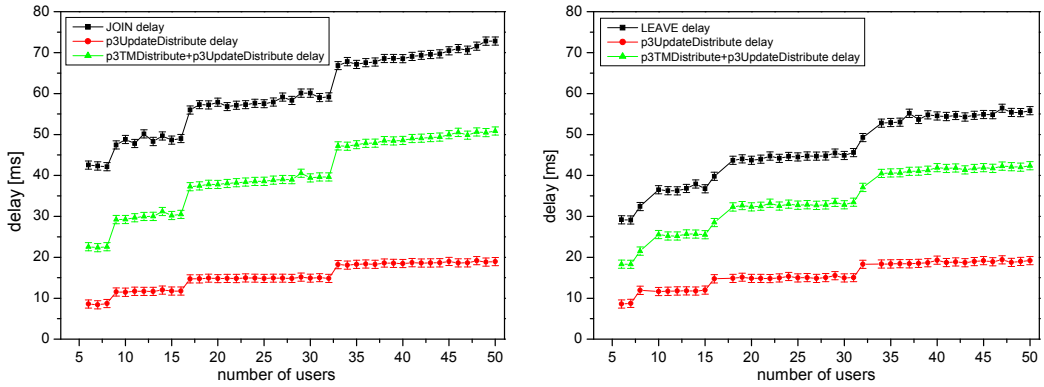

Abbildung 106: Gemessene Zeitdauer für den Gruppenbeitritt (links) und den Gruppenaustritt (rechts) unter dem Einfluss des synthetischen Nutzerverhaltens im Modus Key Agreement

Ein Vergleich zwischen den im praktischen Einsatz gewonnenen Verzögerungszeiten und den durch Simulation ermittelten Zeiten zeigt, dass beide den gleichen Verlauf aufweisen. Allerdings werden bei der Simulation bei größeren Teilnehmerzahlen geringere Werte für die gesamte Zeitdauer einer Nutzeranfrage bzw. die Schlüsselwechseldauer ermittelt. Ursache hierfür ist, dass in der Implementierung, mit der die Messungen durchgeführt werden, beim

Schlüsselwechsel der Schlüsselbaum vollständig gelöscht und danach aus den erhaltenen Informationen wieder neu aufgebaut wird. In einer realen Implementierung würde man den Schlüsselbaum nur entsprechend der erhaltenen Informationen verändern. Die hierfür benötigte Rechenzeit wird bei der Simulation nicht berücksichtigt und führt bei großen Gruppen zu einer Erhöhung der Schlüsselwechseldauer und damit auch zu einer Erhöhung der gesamten Dauer für die Bearbeitung einer Nutzeranfrage.

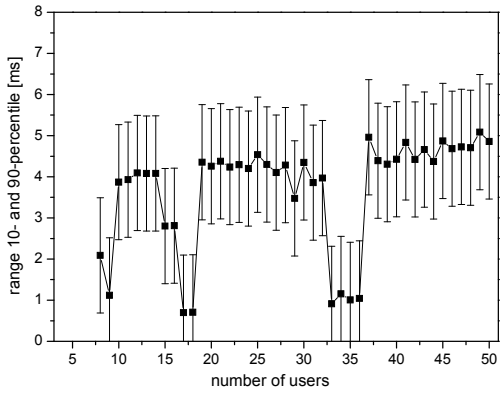

Abbildung 107: Gemessene Streuung der Schlüsselwechselzeitdauer beim Gruppenbeitritt im Modus Key Agreement

Das bei den Messungen eingesetzte Testnetz ist bezüglich der verfügbaren Rechenleistung heterogen. Auf Grund der hohen benötigten Rechenleistung des Betriebsmodus Key Agreement hat die unterschiedliche verfügbare Rechenleistung der Komponenten des Testnetzes einen Einfluss auf die Streuung der Schlüsselwechselzeitdauer (Abbildung 107). Diese Streuung ist von den im Testnetzwerk vorhandenen Rechnern abhängig. Weiterhin erkennt man in Abbildung 107, dass beim Einfügen einer neuen Ebene in den Schlüsselbaum die Streuung der Schlüsselwechselzeitdauer sinkt. Ursache hierfür ist, dass beim Einfügen einer neuen Ebene Nutzer mit der größeren Rechenleistung bereits auf die neu eingefügte Ebene verschoben wurden, während den übrigen Nutzern noch eine Position auf der tieferen Ebene zugeordnet ist. Somit müssen die Teilnehmer mit der geringeren Rechenleistung eine kryptographische Operation weniger durchführen und die Streuung der Schlüsselwechseldauer sinkt.

Die auf einem Monitor gemessene gesamte Zeitdauer für die Teilnehmeroperation JOIN bzw. LEAVE (schwarze Quadrate, JOIN delay) sowie die Schlüsselwechselzeitdauer (rote Kreise, p3UpdateDistribute delay) unter dem Einfluss des synthetischen Nutzerverhaltens im Modus Key Distribution ist in Abbildung 108 dargestellt. Die gemessenen Verzögerungszeiten stimmen innerhalb der Fehlergrenzen mit den in der Simulation ermittelten überein.

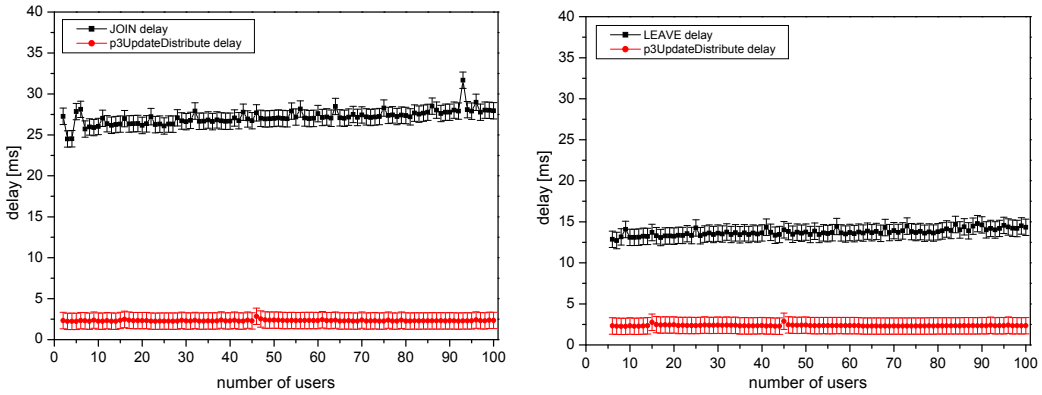

Abbildung 108: Gemessene Zeitdauer für den Gruppenbeitritt (links) und den Gruppenaustritt (rechts) unter dem Einfluss des synthetischen Nutzerverhaltens im Modus Key Distribution

Das bei den Messungen eingesetzte Testnetz ist, wie bereits erwähnt, bezüglich der verfügbaren Rechenleistung heterogen. Wegen der geringen benötigten Rechenleistung des Betriebsmodus Key Distribution bewirkt diese Heterogenität aber nur eine geringe Streuung der Schlüsselwechselzeitdauer. Auf Grund der angenommen Messgenauigkeit kann nur 2,5 ms als obere Grenze für die Streuung der Schlüsselwechselzeitdauer angegeben werden (Abbildung 109). Diese Streuung ist von den im Testnetzwerk verfügbaren Rechnern abhängig.

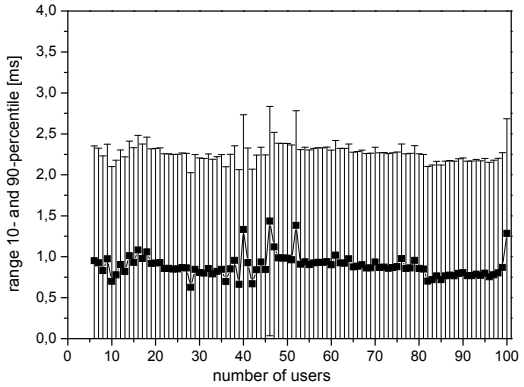

Abbildung 109: Gemessene Streuung der Schlüsselwechselzeitdauer beim Gruppenbeitritt im Modus Key Distribution

In Abbildung 110 ist die auf einem Monitor gemessene gesamte Bearbeitungszeit für eine Teilnehmeroperation JOIN bzw. LEAVE (links) sowie die Schlüsselwechselzeitdauer (rechts) im Modus Key Distribution unter dem Einfluss des zivilen Nutzerverhaltens Nr. 2 dargestellt. Zur Verkürzung der Messzeitdauer wurde die Gruppe mit der durchschnittlichen Teilnehmeranzahl initialisiert. Anschließend werden 48 Stunden lang Teilnehmeroperationen durchgeführt. Nach dieser Zeit befinden sich keine der bei der Initialisierung eingefügten Teilnehmer mehr in der Gruppe. Mit dem auf diese Art eingeschwungenen System wird nun die Messung durchgeführt. Bei dem untersuchten Nutzerverhalten erfordert ein Schlüsselwechsel im Mittel 1,5 ms. Die Nutzer benötigen im Mittel 35 ms für den Gruppenbeitritt und 15 ms für den Gruppenaustritt. Bei dem zivilen Nutzerverhalten ist das Schlüsselmanagement mit einer kontinuierlichen Rate an Nutzeranfragen konfrontiert. Die

selten auftretenden Wiederholungen zeigen, dass das Schlüsselmanagement durch die eintreffenden Nutzeranfragen nicht überlastet wird.

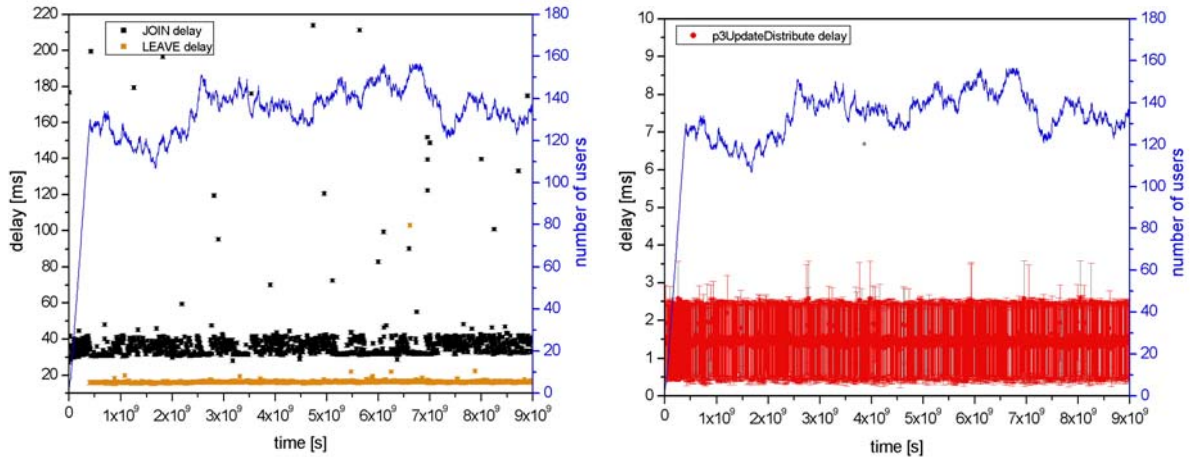

Abbildung 110: Gemessene Zeitdauer für den Gruppenbeitritt bzw. Gruppenaustritt (links) sowie die Schlüsselwechselzeitdauer (rechts) im Modus Key Distribution unter dem Einfluss des zivilen Nutzerverhaltens Nr. 2

Das Ergebnis der Messung des Betriebsmodus Key Distribution unter dem Einfluss des militärischen Nutzerverhaltens Nr. 2 ist in Abbildung 111 dargestellt. Die Beitritts- bzw. Austrittsanfragen wurden einzeln verarbeitet. Zur Verkürzung der Durchführungsdauer der Messung wurden die Zeiträume, in denen keine Änderung der Gruppenzusammensetzung stattfindet, verkürzt. Bei dem untersuchten Nutzerverhalten erfolgt ein Schlüsselwechsel im Mittel in 1,5 ms. Die Nutzer benötigen im Mittel 35 ms für den Gruppenbeitritt und 15 ms für den Gruppenaustritt.

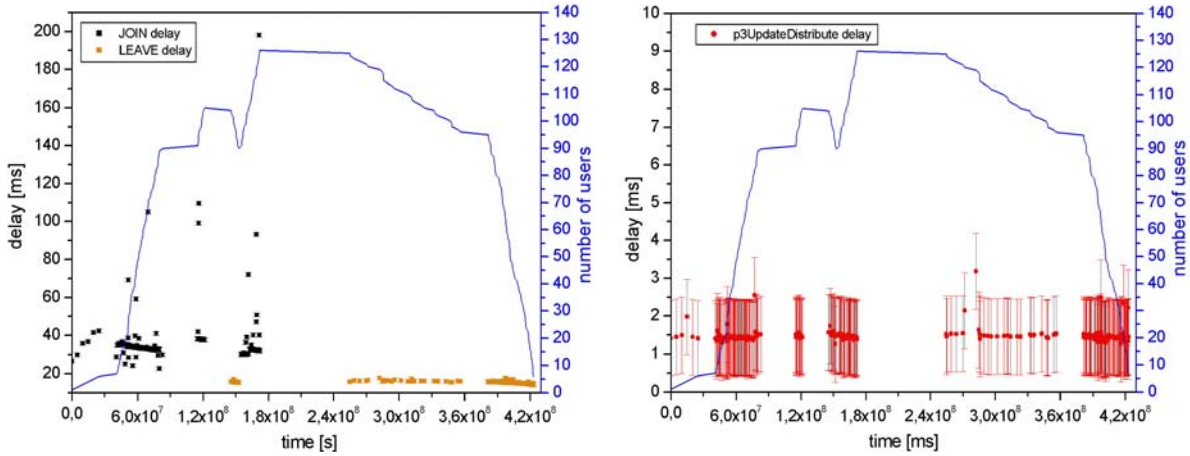

Abbildung 111: Gemessene Zeitdauer für den Gruppenbeitritt bzw. Gruppenaustritt (links) und die Schlüsselwechselzeitdauer (rechts) unter dem Einfluss des militärischen Nutzerverhaltens Nr. 2 im Modus Key Distribution

Die Schlüsselwechseldauer beim zivilen bzw. militärischen und synthetischen Nutzerverhalten unterscheidet sich nicht. Messungen unter dem Einfluss des synthetischen Nutzerhaltens sind deshalb geeignet zur Effizienzanalyse eine Schlüsselmanagements. Treten allerdings bei einem Nutzerverhalten viele Anfragen in einem kurzen Zeitintervall auf, z.B. im Rahmen des militärischen Nutzerverhaltens beim Beitritt der Nutzer der

Fahrzeugpatrouille, finden Wiederholungen der Beitrittsanfrage statt. Diese bewirkt einen erheblichen Anstieg der Gesamtdauer für den Gruppenbeitritt bzw. Gruppenaustritt. Abschließend ist anzumerken, dass die bei der experimentellen Effizienzanalyse verwendeten Metriken, d.h. insbesondere die Schlüsselwechselzeitdauer, erheblich von der Qualität der Implementierung beeinflusst wird. Eine schnelle Schlüsselbereitstellung durch das Konzept MIKE ist durch weitere Verbesserungen der Implementierung möglich. Beispiel für Verbesserungsmöglichkeiten sind Reduktion der Zugriffszeit auf den Baum sowie effizientere Übertragung einer Positionsveränderung von Nutzern im Schlüsselbaum. Weiterhin wird im Betriebsmodus Key Agreement nach jeder Teilnehmeroperation der Schlüsselbaum vollständig gelöscht und aus den erhaltenen Informationen ein neuer Schlüsselbaum aufgebaut, anstatt nur entsprechende Veränderungen vorzunehmen. Außerdem wird bei der Teilnehmeroperation LEAVE unnötigerweise der vollständige Schlüsselbaum übertragen.

8.3 Vergleich des Konzepts MIKE mit existieren Schlüsselmanagementkonzepten

Mit dem Konzept MIKE wurde ein verteiltes Verfahren zur Bereitstellung von Gruppenschlüsseln entworfen und als Betriebsmodus Key Agreement bezeichnet. In das Konzept MIKE wurde der zweite Betriebsmodus Key Distribution integriert, um eine Schlüsselbereitstellung auch dann zu ermöglichen, wenn der Betriebsmodus Key Agreement nicht mehr effizient genug arbeitet. Aus diesem Grund bildet der Vergleich des MIKE-Betriebsmodus Key Agreement mit anderen verteilten Schlüsselmanagementverfahren den Schwerpunkt dieses Abschnitts. Mittels einer einfachen Analyse wurde in Abschnitt 3.4.6 das Verfahren Tree-based Group-Diffie-Hellman (TGDH) als effizientes verteiltes Verfahren zur Bereitstellung eines Gruppenschlüssels identifiziert. In diesem Abschnitt wird ein detaillierter Vergleich mit diesem Verfahren durchgeführt. Wie ebenfalls in Abschnitt 3.4.6 gezeigt wurde, besteht der Nachteil des Verfahrens darin, dass zum Betrieb ein Gruppenkommunikationssystem (GCS) notwendig ist (vgl. Abschnitt 2.2.4). Aus diesem Grund wird zusätzlich ein Vergleich mit dem Verfahren Ingemarsson-Trang-Wong Group-Diffie-Hellman (ITW) durchgeführt. Dieses Verfahren wurde als Vertreter eines verteilten Verfahrens zur Gruppenschlüsselbereitstellung, das kein Gruppenkommunikationssystem benötigt, ausgewählt.

8.3.1 Vergleich der Verlässlichkeit

Verteilte Verfahren zur Bereitstellung von Gruppenschlüsseln sind bei Prozessfehlern der Fehlerklasse Zusammenbruchfehler reparabel. In diesem Abschnitt wird die Reparaturzeit der Verfahren ITW, TGDH und des Betriebsmodus Key Agreement in einer Gruppe mit U Teilnehmern nach einem solchen Fehler verglichen. Wird nach der Zeit $t_{Diagnosis}$ ein Prozessfehler festgestellt, müssen die Prozesse eine Einigung über die Teilnehmer der Gruppe erzielen. Hierzu kann z.B. das Totem-M-Protokoll (vgl. Abschnitt 2.2.4) eingesetzt werden. Da bei dem Verfahren ITW die Prozesse in einer festen Reihe angeordnet sind, ist bei diesem Verfahren alternativ der Einsatz des Ring-Algorithmus möglich. Die Zeitdauer, die für eine Einigung über die Mitgliedschaft in der Gruppe benötigt wird, wird mit t_{Member} bezeichnet. Bei allen drei Verfahren wird, nachdem ein Fehler erkannt wurde, der fehlerhafte Prozess bzw.

Nutzer durch Schlüsselwechsel von der Gruppe ausgeschlossen. Wird dieser in der Zeitdauer t_{Update} durchgeführt, gilt für die Reparaturzeit t_{TTR} der Verfahren:

$$t_{TTR} = t_{Diagnosis} + t_{Member} + t_{Update}$$

Schwerpunkt der weiteren Betrachtungen ist die Zeit $t_{Diagnosis}$, die für die Diagnose eines Fehlers benötigt wird. Diese hat einen entscheidenden Einfluss auf die Reparaturzeit t_{TTR}, wird vorausgesetzt, dass alle drei Schlüsselbereitstellungsverfahren den gleichen Mechanismus zur Einigung über die Teilnehmer der Gruppe verwenden. Bei der Analyse der Zeit zur Fehlererkennung wird für den Betriebsmodus Key Agreement der Fall untersucht, bei dem der TM ausgefallen ist. Für das Verfahren TGDH wird angenommen, dass der für den Betrieb erforderliche zuverlässige, geordnete Multicast-Dienst durch das Totem-SR-Protokoll (vgl. Abschnitt 2.2.4) sichergestellt wird. Tabelle 28 gibt einen Überblick über die Zeitdauer für die Diagnose eines Prozessfehlers. In der Tabelle wird mit $t_{maxMsgTimeout}$ die Zeit für die Durchführung der maximalen Anzahl an Übertragungswiederholungen bezeichnet. Wie erwartet, wird beim Verfahren TGDH ein Prozessfehler auf Grund des Gruppenkommunikationssystems am schnellsten bemerkt. Im ungünsten Fall wird ein Umlauf des Regular-Token benötigt, d.h. U-1 Übertragungen des Regular-Token $t_{R\text{-}Token}$, um einen Prozessfehler zu bemerken. Im günstigen Fall wird ein Prozessfehler schon nach einer Übertragung des Regular-Token bemerkt. Bei dem Verfahren ITW und Key Agreement dauert die Fehlererkennung wesentlich länger. Diese erkennen im ungünstigen Fall einen Prozessfehler nach dem Ablauf der Gültigkeitsdauer des Gruppenschlüssels $t_{KeyTimeout}$. Üblicherweise wird die Gültigkeitsdauer eines Gruppenschlüssels in der Größenordung von Stunden gewählt. Der ungünstige Fall tritt ein, falls keine Teilnehmeroperation stattfindet. Ein Nachteil bei dem Verfahren Key Agreement besteht darin, dass ein Nutzer nach der Überschreitung der Zeitschranke $t_{KeyTimeout}$ zunächst eine erneute Gruppenanmeldung versucht, ehe mit der Reparatur des Schlüsselmanagementsystems begonnen wird. Werden von den Verfahren ITW und Key Agreement Nutzeranfragen bearbeitet, tritt der günstige Fall ein und Prozessfehler werden schneller erkannt.

Verfahren	Günstiger Fall	Ungünstiger Fall
ITW	$t_{maxMsgTimeout(p3UD)}$	$t_{KeyTimeout}$
TGDH	$t_{maxMsgTimeout(R\text{-}Token)}$	$(U\text{-}1)\cdot t_{R\text{-}Token}+t_{maxMsgTimeout(R\text{-}Token)}$
Key Agreement	$t_{MsgTimeout(p3TD)}+t_{maxMsgTimeout(p1JR)}$	$t_{KeyTimeout}+t_{maxMsgTimeout(p1JR)}$

p3TD=p3TMDistribute p3UD=p3UpdateDistribute p1JR=p1JoinRequest R-Token=Regular-Token

Tabelle 28: Zeitdauer für die Diagnose eines Prozessfehlers

8.3.2 Theoretischer Vergleich der Effizienz

Der theoretische Vergleich der Effizienz des Schlüsselwechsels wird unter dem Aspekt Vergleich der Komplexität des Betriebsmodus Key Agreement mit den aus der Literatur bekannten Verfahren ITW und TGDH durchgeführt. Das heißt die Abhängigkeit der in Abschnitt 6.2 definierten und als einfachere Metriken bezeichneten Messwerte von der Teilnehmeranzahl wird analysiert. Hierbei wird der Anmelde- bzw. Abmeldevorgang nicht betrachtet. Diese Untersuchungen basieren auf der in Tabelle 11 und Tabelle 16 zusammengefassten Komplexität bezüglich der Teilnehmerzahl.

In Tabelle 29 ist die Komplexität des Schlüsselwechsels für die Verfahren ITW, TGDH und Key Agreement im ungünstigen Fall zusammengefasst. Den in dieser Tabelle enthaltenen Abschätzungen liegt die Annahme zugrunde, dass nach der Teilnehmeroperation JOIN bzw. LEAVE/EJECT die Gruppe aus U Teilnehmern besteht. Bei der Teilnehmeroperation MERGE wird der Zusammenschluss von zwei Teilgruppen U′ und U″ zu einer Gruppe der Größe U‴=U′+U″ betrachtet. Weiterhin wird für die Verfahren Key Agreement und TGDH angenommen, dass der TM bzw. der Sponsor, der die Teilnehmeroperation durchführt, die Position $v_{\ell,p}$ im Schlüsselbaum mit der Pfadlänge $k=\|path(v_{\ell,p})\|$ besitzt. Außerdem wird für die Teilnehmeroperation PARTITION bezüglich der Verfahren ITW und TGDH nur der Austritt von Teilnehmern mit einem gemeinsamen Wurzelknoten betrachtet. Andernfalls wird, wie in Abschnitt 4.4.1 erläutert, angenommen, dass jeder Nutzer die Teilnehmeroperation LEAVE durchführt. Die Komplexität des Betriebsmodus Key Agreement und des Verfahrens TGDH ist bezüglich der Anzahl der durchgeführten kryptographischen Operationen identisch und steigt logarithmisch bei linear steigender Teilnehmerzahl. Im Gegensatz dazu steigt beim Verfahren ITW die Anzahl der kryptographischen Operationen linear. Das TGDH-Verfahren benötigt für die Durchführung der Teilnehmeroperation LEAVE bzw. PARTITION eine Nachricht. Im MIKE-Modus Key Agreement sind hierzu zwei Nachrichten erforderlich. Mit einer wird der TM für die Operation festgelegt. Die zweite Nachricht enthält die Informationen zur Aktualisierung des Schlüsselbaums. Bei dem Verfahren ITW steigt die Anzahl der übertragenen Nachrichten bei linear steigender Teilnehmerzahl quadratisch. Im Hinblick auf die Anzahl der ausgetauschten Hilfsschlüssel muss im Modus Key Agreement die gleiche bzw. eine geringere Zahl ausgetauscht werden. Ausnahme bildet die Operation MERGE. Bei dieser müssen 2·U′-1 Hilfsschlüssel zusätzlich übertragen werden, falls U′ die Teilnehmeranzahl der bestehenden Gruppe ist.

Verfahren	Teilnehmeroperation	Anzahl der Nachrichten	Anzahl der Exponentiationen	Anzahl der Hilfsschlüssel
MIKE, Betriebsmodus Key Agreement	JOIN	2	2·k-1	2·U-1
	MERGE	2	2·k-1	2 (2·U′-1)+2·U″-1
	LEAVE/EJECT	2	2·k-1	k-1
	PARTITION	2	2·k-1	k-1
TGDH	JOIN	2	2·k-1	2·U-1
	MERGE	2	2·k-1	2·U′-1+2·U″-1
	LEAVE/EJECT	1	2·k-1	2·U-1
	PARTITION	1	2·k-1	2·U-1
ITW	JOIN	U·(U-1)	U-2	U·(U-1)
	MERGE	U‴·(U‴-1)	U‴-2	U‴·(U‴-1)
	LEAVE/EJECT	U·(U-1)	U-2	U·(U-1)
	PARTITION	U·(U-1)	U-2	U·(U-1)

Tabelle 29: Komplexität der Verfahren ITW, TGDH und des Modus Key Agreement

Beim Verfahren ITW enthält jede im Rahmen des Schlüsselwechsels ausgetauschte Nachricht einen Hilfsschlüssel. Somit steigt die Anzahl der übertragenen Hilfsschlüssel ebenfalls quadratisch. Die durchgeführte Analyse bezüglich der ausgetauschten Nachrichten ist nur

bedingt aussagekräftig, weil das vom Protokoll TGDH benötigte Gruppenkommunikationssystem Quittierungen für jede übertragene Nachricht durchführt.

8.3.3 Simulationsumgebung für den Effizienzvergleich

Zum Vergleich der Effizienz wurden in den Simulator ns-2 die Verfahren TGDH und ITW integriert. Allerdings wurden einige Modifikationen an dem in Abschnitt 3.4.5 dargestellten Verfahrensablauf vorgenommen. Bei den Verfahren TGDH und ITW wird zur Übermittlung der Gruppenbeitrittsanfrage nur eine Nachricht verwendet. Deshalb bieten die Verfahren bei der Gruppenanmeldung keinen Schutz gegen Angriffe durch wiederholtes Senden, d.h. kein störsicheres Verfahren zur Zugangskontrolle (vgl. Bewertung in Tabelle 14). Um diesen geforderten Schutz (Forderung 2, Tabelle 6) zu gewährleisten, müssen bei der Gruppenanmeldung drei Nachrichten ausgetauscht werden. Für den Vergleich wurde dieser Schutz in die Verfahren TGDH und ITW integriert. Der im Simulator realisierte Protokollablauf für die beiden Verfahren ist in Abbildung 112 graphisch dargestellt. Gemäß Forderung 8, Tabelle 6 wird das Schlüsselmanagement benötigt, um einen Nutzdatenschutz mittels IPSec zu ermöglichen. Hierzu sind pro Nutzer 45 Byte Konfigurationsinformation notwendig, die vom Schlüsselmanagement übertragen werden müssen. Die Übertragung dieser Informationen wurde beim Entwurf der Verfahren ITW und TGDH nicht berücksichtigt (vgl. Bewertung in Tabelle 14). Die Übertragung der IPSec-Konfiguration wird ebenfalls in die Verfahren integriert.

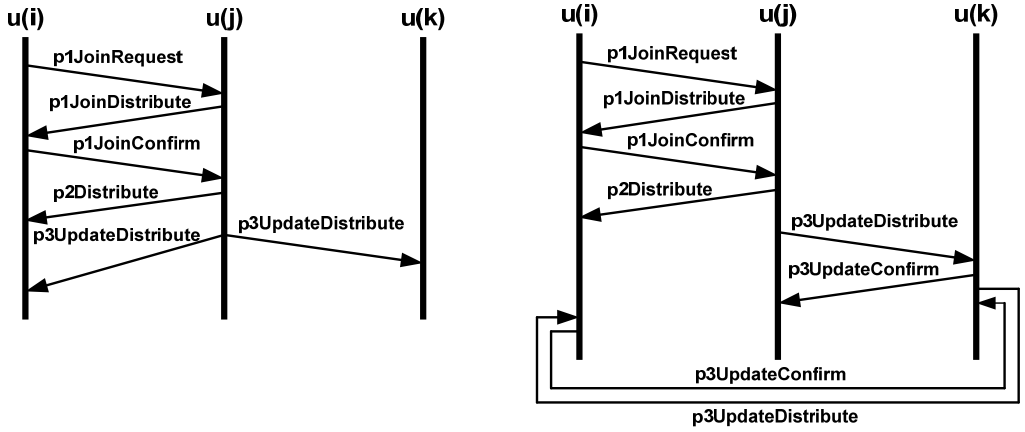

Abbildung 112: Protokoll beim Gruppenbeitritt der Verfahren TGDH (links) und ITW (rechts)

Bei der Simulation wird die in Abschnitt 8.2.3 dargestellte Simulationsumgebung verwendet, d.h. die Verarbeitungszeit von rechenleistungsintensiven kryptographischen Operationen bei der Ermittlung des Gruppenschlüssels wird durch Warteschlangen abstrahiert. Zur Gewährleistung der Authentizität der Schlüsselmanagementnachrichten werden sowohl von den Verfahren TGDH und ITW als auch vom Betriebsmodus Key Agreement digitale Signaturen verwendet. Ein wichtiger Aspekt bei deren Verifikation ist die Überprüfung der Gültigkeit des Authentisierungsschlüssels durch eine Überprüfung des Nutzerzertifikats. Dadurch, dass nur Nachrichten mit einem gültigen Signaturschlüssel akzeptiert werden, wird ein erneuter Gruppenbeitritt von Nutzern, die ausgeschlossen wurden, verhindert. Bei der Simulation wird die Berechnung und Überprüfung von digitalen Signaturen und die

Überprüfung der Gültigkeit des Signaturschlüssels ebenfalls durch Warteschlangen abstrahiert.

Wie bereits erwähnt, wird im Gegensatz zu dem Verfahren ITW und dem Betriebsmodus Key Agreement für den Betrieb des Schlüsselmanagements TGDH ein Gruppenkommunikationssystem benötigt, das die Semantik Virtual Synchrony (vgl. Abschnitt 2.2.4) gewährleistet. Ein abstrahiertes Gruppenkommunikationssystem wurde deshalb ebenfalls in den Simulator integriert (Abbildung 113). Ein Gruppenkommunikationssystem (GCS) gewährleistet den Gruppenmitgliedschaftsdienst und den zuverlässigen, geordneten Multicast-Dienst. Für die Simulation wird angenommen, dass das Gruppenkommunikationssystem Totem (vgl. Abschnitt 2.2.4) eingesetzt wird. Der Mitgliedschaftsdienst eines GCS ermittelt die derzeitgen Teilnehmer der Gruppe. Dieser wird vor jeder Teilnehmeroperation des Schlüsselmanagementsystems ausgeführt. Sein Zeitbedarf $\Delta t_{Membership}$ wird mit der nachfolgenden Formel abgeschätzt und als zusätzlicher Offset berücksichtigt (vgl. Abschnitt 3.4.6):

$$\Delta t_{Membership} = 2 \cdot U \cdot t_{Join} + 2 \cdot U \cdot (t_{C\text{-}Token} + t_{C-Token\text{-}Ack})$$

In der vorherigen Formel wird die in Tabelle 13 zusammen gefasste Notation verwendet. Durch eine entsprechende Steuerung der Simulation wird Sending View Delivery (vgl. Abschnitt 2.2.4) derart gewährleistet, dass in einer Gruppensicht alle Nachrichten ausgeliefert werden und der Recovery-Mechanismus des Gruppenkommunikationssystems nicht benötigt wird. Der zuverlässige, total geordnete Multicast-Dienst wird durch ein in einem logischen Ring zirkulierendes Token realisiert.

Zum Schluss des Abschnitts sei nochmals erwähnt, dass das Verfahren ITW auf Grund seines Protokollablaufs und das Verfahren TGDH wegen des benötigten Gruppenkommunikationssystems nicht einsetzbar sind, wenn sich ein Teil der Nutzer im Zustand EMCON befindet.

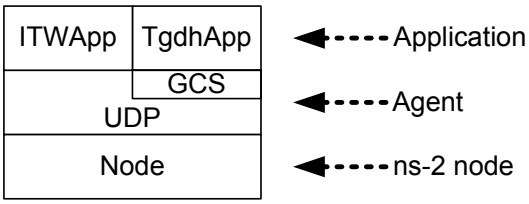

Abbildung 113: Integration des Gruppenkommunikationssystems in den Simulator ns-2

8.3.4 Ergebnisse des Effizienzvergleichs durch Simulation

Zum Effizienzvergleich durch Simulation in diesem Abschnitt wird als Metrik die Datenmenge für den Schlüsselwechsel sowie die gesamte Zeitdauer für den Gruppenbeitritt und Gruppenaustritt verwendet (vgl. Abschnitt 6.1). Der Effizienzvergleich wird für die in Abbildung 98 dargestellten Kommunikationsinfrastrukturen durchgeführt. Auf Grund der für den praktischen Einsatz sehr hohen Verzögerungszeiten bei VHF wird sich jedoch auf die Infrastrukturen Ethernet, SDR und WAN beschränkt.

In Abbildung 114 ist die übertragene Datenmenge beim Schlüsselwechsel infolge eines Gruppenbeitritts für alle drei Verfahren dargestellt. Diese steigt bei linear steigender

Teilnehmerzahl ebenfalls linear. Sowohl bei dem Verfahren TGDH als auch beim Betriebsmodus Key Agreement wird beim Schlüsselwechsel der gesamte Schlüsselbaum übertragen. Das Verfahren TGDH weist allerdings eine geringere übertragene Datenmenge auf, da beim Betriebsmodus Key Agreement der gesamte Schlüsselbaum mit zwei Nachrichten übertragen wird. In Abbildung 114 ist für das Verfahren ITW die übertragene Datenmenge beim Schlüsselwechsel pro Nutzer dargestellt. Die Darstellung der gesamten Datenmenge für den Schlüsselwechsel würde eine quadratische Abhängigkeit von der Nutzerzahl zeigen. In Abbildung 114 ist die übertragene Datenmenge des Gruppenkommunikationssystems nicht berücksichtigt.

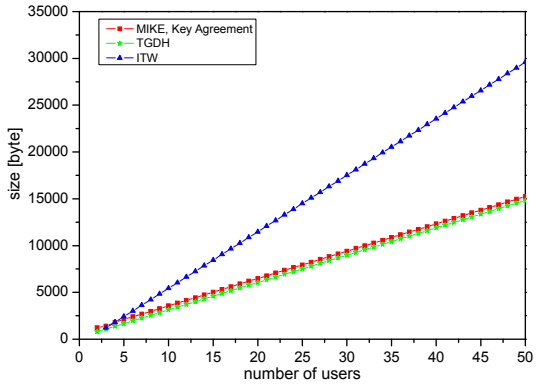

Abbildung 114: Übertragene Datenmenge beim Gruppenbeitritt für die Verfahren ITW, TGDH und für den Modus Key Agreement

Der Zeitdauervergleich für den Gruppenbeitritt und Gruppenaustritt wird mit den im Ethernet ermittelten Zeiten begonnen. Ein Vergleich der in Abbildung 115 dargestellten Zeiten der Verfahren ITW (blaue Dreiecke), TGDH (grüne Sterne) und des Modus Key Agreement (rote Kreise) zeigt, dass durch den Betriebsmodus Key Agreement nur eine geringe Verbesserung der gesamten Zeitdauer für den Gruppenbeitritt bzw. Gruppenaustritt im Vergleich zum Verfahren TGDH erzielt wird. In kleinen Gruppen benötigt das Verfahren TGDH nach einer Teilnehmeroperation sogar weniger Zeit zur Bereitstellung eines Gruppenschlüssels. Minimalwerte bei der Zeit für die Schlüsselbereitstellung durch das Verfahren TGDH treten auf, falls der Sponsor die Zeit ermittelt. Das Verfahren ITW benötigt eine erheblich längere Zeit zur Durchführung eines Schlüsselwechsels als die beiden anderen Verfahren. Die Ursache hierfür besteht darin, dass eine Vielzahl von Nachrichten übertragen wird. Diese besitzen zwar nur eine geringe Größe, müssen aber zur Gewährleistung der Authentizität bzw. Integrität alle mit einer digitalen Signatur geschützt werden. Deren Erzeugung bzw. Verifikation führt zu dem erheblichen Zeitbedarf des Protokolls. Weiterhin entsteht durch den sequenziellen Protokollablauf eine große Streuung der Zeiten für den Gruppenbeitritt bzw. Austritt. Es sei noch angemerkt, dass die für das Verfahren ITW gemessenen Zeiten von dem Nutzer ermittelt wurden, der den Schlüsselwechsel initiiert. In Abbildung 115 ist zusätzlich der Anteil des Gruppenmitgliedschaftsdiensts an der Zeitdauer für die Schlüsselbereitstellung nach einer Teilnehmeroperation beim Verfahren TGDH dargestellt (schwarze Quadrate).

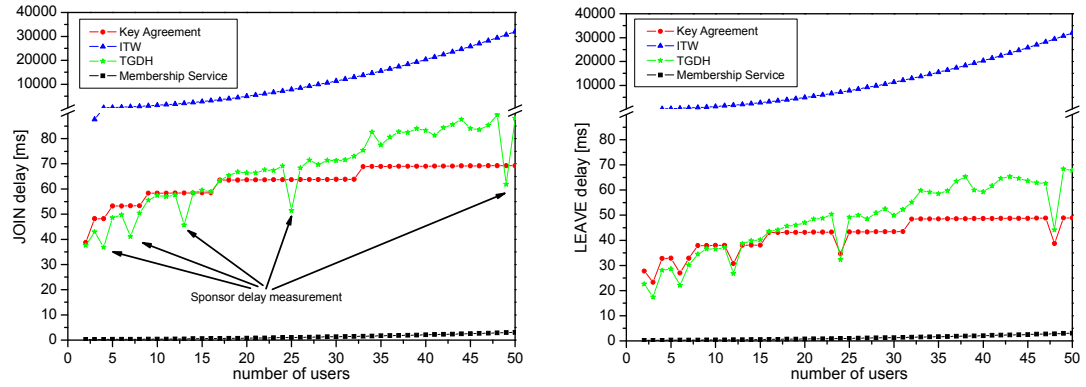

Abbildung 115: Simulierte Zeitdauer für den Gruppenbeitritt (links) und den Gruppenaustritt (rechts) bei den Verfahren ITW, TGDH und dem Modus Key Agreement im Ethernet

Im nachfolgenden Abschnitt wird ein Vergleich der Zeitdauer des Verfahrens TGDH und des Betriebsmodus Key Agreement beim Betrieb in der Kommunikationsinfrastruktur SDR durchgeführt. Das Verfahren ITW wird nicht weiter betrachtet, da sich dessen Ineffizienz schon im Ethernet gezeigt hat. In Abbildung 116 ist die simulierte Zeitdauer für die Durchführung einer Teilnehmeroperation JOIN bzw. LEAVE des Verfahrens TGDH (grüne Sterne) und des Modus Key Agreement (rote Kreise) dargestellt. Die Abbildung verdeutlicht die bereits in Abschnitt 3.4.5 prognostizierte Schwäche von TGDH in einer derartigen Infrastruktur auf Grund des benötigten Gruppenkommunikationssystems. Die Skalierbarkeit des Modus Key Agreement kann durch Umschalten auf den zweiten Betriebsmodus verbessert werden. (vgl. Abbildung 103, links)

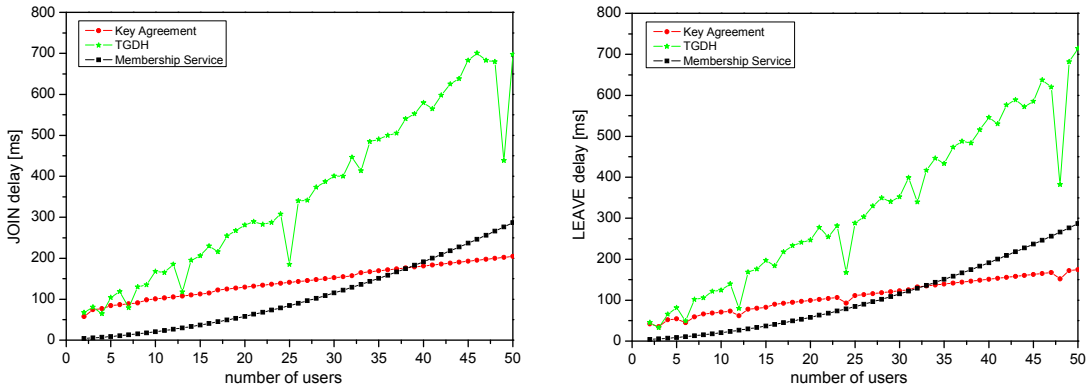

Abbildung 116: Simulierte Zeitdauer für den Gruppenbeitritt (links) und den Gruppenaustritt (rechts) bei dem Verfahren TGDH und dem Modus Key Agreement über SDR

In Abbildung 117 ist die simulierte Zeitdauer für den Gruppenbeitritt und den Gruppenaustritt des Verfahrens TGDH (grüne Sterne) und Key Agreement (rote Kreise) beim Betrieb im WAN dargestellt. Die dargestellten Verzögerungszeiten wurden am Standort B ermittelt (vgl. Abbildung 98). Man erkennt beim Vergleich die deutlich verminderten Zeiten für das Verfahren Key Agreement. Die dabei auftretenden Extremwerte der Verzögerungszeiten können aber verhindert werden, indem zusätzlich der in Abschnitt 7.5 dargestellte Algorithmus zur ressourcengesteuerten Auswahl des TM verwendet wird (vgl. Abbildung 103).

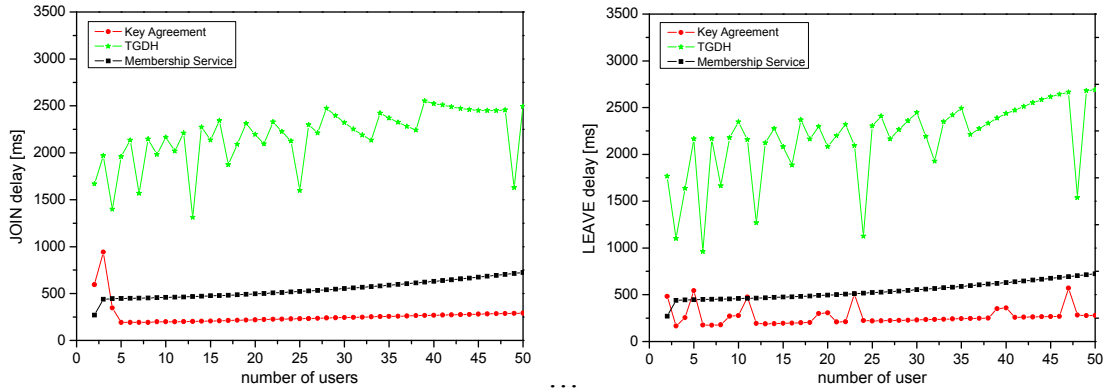

Abbildung 117: Simulierte Zeitdauer für Gruppenbeitritt (links) und Gruppenaustritt (rechts) bei dem Verfahren TGDH und dem Modus Key Agreement im WAN

8.3.5 Experimentelle Effizienzanalyse in der Literatur

In diesem Abschnitt werden die in der Literatur dokumentierten Effizienzanalysen von mit den Modi Key Agreement und Key Distribution vergleichbaren Verfahren zusammengefasst und mit den in dieser Arbeit entstandenen Ergebnissen verglichen. Begonnen wird mit den in der Literatur dokumentierten und mit dem Modus Key Agreement vergleichbaren Verfahren.

Die in Abschnitt 3.4.5 beschriebenen verteilten Verfahren zur Gruppenschlüsselbereitstellung Burmester-Desmedt Group Diffie-Hellman (BD), Group Diffie-Hellman (GDH), Skinny Tree Protocol (STR) und Tree-based Group Diffie-Hellman (TGDH) wurden in Kombination mit dem Gruppenkommunikationssystem Spread [Sta98] einer Effizienzanalyse unterzogen [Ami02]. Alle untersuchten Verfahren basieren auf der iterativen Anwendung des DH-Algorithmus. Bei den Untersuchungen wird der DH-Algorithmus mit einer Schlüssellänge von 512 Bit eingesetzt. Zur Gewährleistung der Authentizität der für die Schlüsselverwaltung ausgetauschten Nachrichten werden diese mit einer digitalen Signatur versehen. Zur Signaturerzeugung wird der asymmetrische Algorithmus RSA mit einer Schlüssellänge von 1024 Bit genutzt. Bei der Effizienzanalyse wurde der sukzessive Beitritt von Nutzern zu einer Gruppe und anschließend der sukzessive Austritt der Teilnehmer in umgekehrter Reihenfolge als Last verwendet. Dieses entspricht dem in Abschnitt 6.3 vorgestellten synthetischen Nutzerverhalten, welches auch bei der Analyse des Schlüsselmanagements MIKE eingesetzt wird. Als Metrik bei den Untersuchungen wurde die gesamte Zeitdauer für den Gruppenbeitritt bzw. –austritt verwendet (vgl. Abschnitt 6.1).

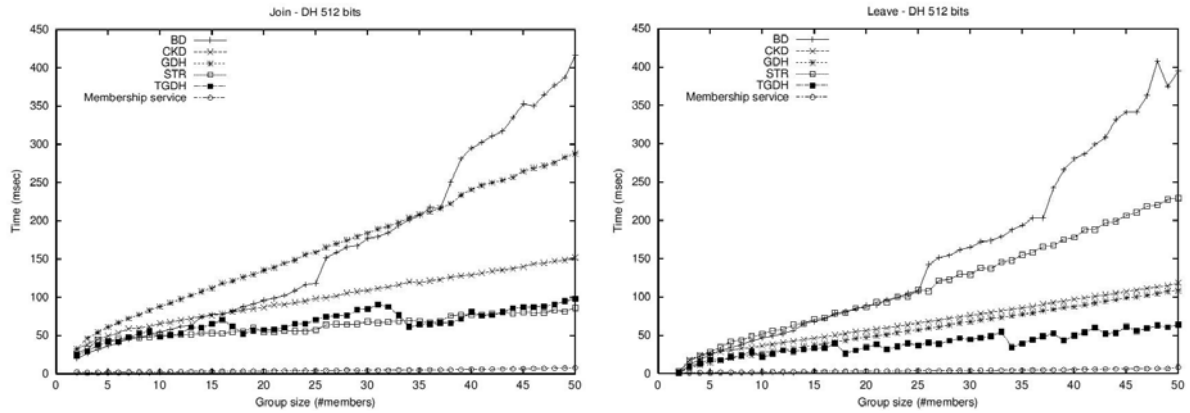

Abbildung 118: Gemessene Verzögerungszeiten für den Gruppenbeitritt (links) und Gruppenaustritt (rechts) für das Verfahren TGDH im Ethernet [Ami02]

Zunächst wurde die Effizienzanalyse in einer Kommunikationsinfrastruktur durchgeführt, bei der Nutzer über Ethernet verbunden sind. Die Ergebnisse dieser Effizienzanalyse sind in Abbildung 118 getrennt für den sukzessiven Gruppenbeitritt bzw. -austritt dargestellt. Zusätzlich wurde die Effizienz der Verfahren beim Betrieb über Weitverkehrsnetze untersucht. Hierbei wurde die in Abbildung 98, rechts dargestellte Netwerktopologie verwendet. Die Ergebnisse sind in Abbildung 119 getrennt für den sukzessiven Gruppenbeitritt bzw. -austritt dargestellt. Aus der Abbildung 118 und Abbildung 119 entnimmt man die bessere Leistungsfähigkeit des Verfahrens TGDH gegenüber den anderen Verfahren.

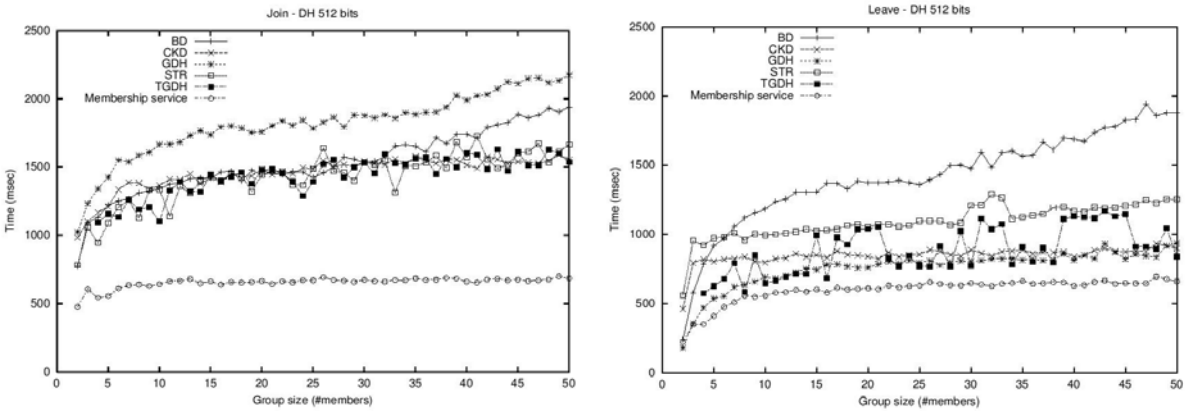

Abbildung 119: Gemessene Verzögerungszeiten für den Gruppenbeitritt (links) und Gruppenaustritt (rechts) beim Betrieb von TGDH im WAN [Ami02]

Im Folgenden werden die in den beiden vorherigen Abbildungen darstellten Messergebnisse mit den Simulationsergebnissen in Abbildung 115 bzw. Abbildung 117 verglichen. Bei der Simulation des Verfahrens TGDH wurde für die Schlüsselbereitstellung eine höhere Zeitdauer ermittelt. Eine Ursache hierfür besteht darin, dass bei den praktischen Messungen in [Ami02] zur Übermittlung der Gruppenbeitrittsanfrage nur eine Nachricht verwendet wird. Um ein störsicheres Verfahren zur Zugriffskontrolle zu ermöglichen, wurden in der Simulation zur Gruppenanmeldung drei Nachrichten ausgetauscht. Zur Gewährleistung der Authentizität der Schlüsselmanagementnachrichten werden digitale Signaturen verwendet. Ein wichtiger Aspekt bei deren Verifikation ist die Überprüfung der Gültigkeit des

Authentisierungsschlüssels durch eine Überprüfung des Nutzerzertifikats. Dies wird bei der in [Ami02] vorgestellten Analyse des Verfahrens TGDH ebenfalls nicht berücksichtigt und ist eine weitere Ursache für die Erhöhung der bei der Simulation ermittelten Zeiten. Wird ein Schlüsselmanagement benötigt, um einen Nutzdatenschutz mittels IPSec zu ermöglichen, müssen beim Schlüsselwechsel pro Nutzer 45 Byte Konfigurationsinformation übermittelt werden. Die Übertragung dieser Informationen bei der Simulation ist der dritte Grund für die größeren Verzögerungszeiten.

Bei der Simulation wird im Gegensatz zu den Messungen in [Ami02] für die Teilnehmeroperation LEAVE ein ähnliches Verhalten beobachtet wie für die Teilnehmeroperation JOIN. Grund hierfür ist, dass in der Implementierung des Verfahrens TGDH, mit der die Simulationen durchgeführt wurden, nach jeder Teilnehmeroperation der vollständige Schlüsselbaum übertragen wird. Bei der Implementierung, mit der die Messungen in [Ami02] durchgeführt wurden, werden nach einer Teilnehmeroperation LEAVE nur die Änderungen im Schlüsselbaum übertragen, um so nur noch logarithmischen Übertragungsaufwand zu bekommen. Die in Abbildung 115 und in Abbildung 118 dargestellten Zeiten des Mitgliedschaftsdiensts zeigen, dass dieser im Ethernet nur einen geringen Einfluss auf die Schlüsselbereitstellungszeit nach einer Teilnehmeroperation hat. Die Abbildung 117 hingegen verdeutlicht, dass bei der Simulation der Mitgliedschaftsdienst im WAN eine linear steigende Verzögerungszeit bei der Schlüsselbereitstellung durch das Verfahren TGDH verursacht. In der Messung wird dieses Verhalten nicht festgestellt, da eine Gruppe von bis zu 50 Teilnehmern untersucht wurde, die nur über 14 Rechner verfügt.

Der Vergleich der Simulationsergebnisse mit den eigenen sowie den in [Ami02] erzielten praktischen Ergebnissen zeigt, dass die im Rahmen der Arbeit erstellte Simulationsumgebung geeignet ist, eine Beurteilung von Verfahren zur Gruppenschlüsselbereitstellung durchzuführen.

Nachfolgend werden die in der Literatur dokumentierten Effizienzanalysen von mit dem Modus Key Distribution vergleichbaren Verfahren zusammengefasst und mit den in dieser Arbeit entstandenen Ergebnissen verglichen. Eine Leistungsbewertung des Konzepts Group Domain in Kombination mit dem in [Won98] spezifizierten Verfahren zur Gruppenschlüsselbereitstellung wurde bisher nicht durchgeführt. Allerdings wurde das in [Won98] definierte Verfahren mit einem proprietären Protokoll untersucht. Bei der Analyse übernahm ein Rechner die Aufgaben des Schlüsselverwalters. Auf einem anderen Rechner wurde ein Simulator gestartet, der die Nutzer simulierte. Die Rechner standen über ein Netzwerk basierend auf dem Übertragungsmedium Ethernet in Verbindung. Die Kommunikation erfolgte über UDP. Der Nutzersimulator initialisierte den Schlüsselverwalter mit der gewünschten Teilnehmerzahl. Nach Erreichen dieser Gruppengröße wurden 1000 zufällig verteilte Teilnehmeroperationen JOIN bzw. LEAVE vom Nutzersimulator durchgeführt. Gemessen wurde die mittlere benötigte Zeitdauer des Schlüsselservers für die Durchführung einer Teilnehmeroperation (Abbildung 120). Bei der Messung wurde ein Schlüsselbaum vom Grad d=4 eingesetzt.

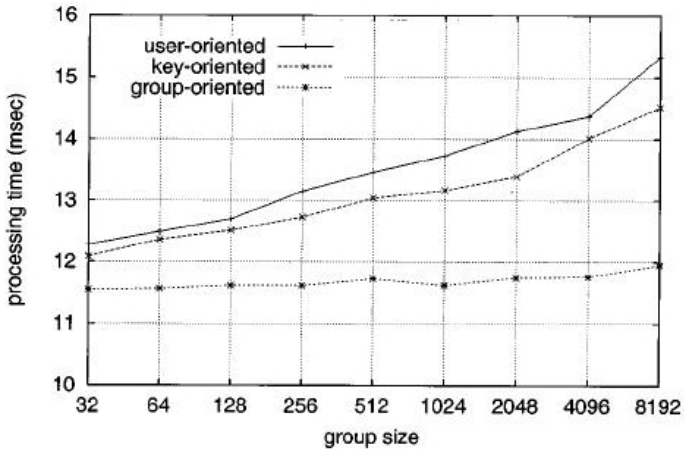

Abbildung 120: Mittlere Zeitdauer für die Schlüsselbereitstellung durch den Schlüsselserver [Won98]

Ziel der in diesem Abschnitt vorgestellten Untersuchung ist, zu messen in welchem Zeitraum ein Schlüsselserver die Informationen zum Schlüsselwechsel erzeugen kann. Die Übermittlung dieser Informationen wird bei der Untersuchung nicht berücksichtigt. Die in [Won98] durchgeführte Analyse ist daher nur bedingt aussagekräftig, um die praktische Nutzbarkeit des Verfahrens zu zeigen. Deshalb wurde mit der Untersuchung des Modus Key Distribution im Rahmen dieser Arbeit erstmals eine Analyse eines Verfahrens zur Gruppenschlüsselbereitstellung für IPSec im praktischen Einsatz durchgeführt.

8.4 Kapitelzusammenfassung

In diesem letzten Kapitel wurden die Untersuchungen zur Verifikation der Trägfähigkeit des Konzeptes MIKE vorgestellt. Hierzu wurden die zwei Bewertungskriterien Verlässlichkeit und Effizienz eingesetzt. Die Analysen zur Verlässlichkeit und Effizienz des Schlüsselwechsels wurden mittels Simulation und einer experimentellen Implementierung des Konzepts durchgeführt. Die Funktionsfähigkeit der Mechanismen zur Kompensation temporärer Kommunikationsstörungen konnten mittels software-implementierter Fehlerinjektion qualitativ nachgewiesen werden. Weiterhin wurde zur Bewertung der Verlässlichkeit die Reparaturzeit beim Zusammenbruchfehler eines Nutzers bzw. des TMs abgeschätzt. Beim Vergleich der beiden Betriebsmodi wurde festgestellt, dass der Modus Key Distribution verfahrensbedingt effizienter bei der Durchführung eines Schlüsselwechsels ist. Allerdings wird im Modus Key Agreement durch eine gleichzeitige Verwaltung des Schlüsselbaums bei allen Teilnehmern eine Reparierbarkeit gewährleistet. Die durchgeführte Effizienzanalyse ergab, dass das als Betriebsmodus Key Agreement entworfene verteilte Schlüsselbereitstellungsverfahren eine schnellere Schlüsselbereitstellung als vergleichbare Konzepte ermöglicht. Weiterhin konnte die Flexibilität des Schlüsselmanagements MIKE nachgewiesen werden, indem gezeigt wurde, dass durch die Verfügbarkeit eines zweiten Betriebsmodus auch in Netzwerken mit sehr geringer Datenübertragungskapazität eine Schlüsselbereitstellung in akzeptabler Zeit möglich ist.

9 Zusammenfassung und Ausblick

Im Rahmen der vorliegenden Arbeit wurde die Gruppenschlüsselbereitstellung sowohl in Dynamic Peer Groups als auch in großen Gruppen untersucht. Insbesondere wurde der Fall betrachtet, wenn Dynamic Peer Groups zu großen Gruppen wachsen. Die Ergebnisse der Untersuchung werden unter den Aspekten Schlüsselmanagementkonzept, Fehlertoleranz, szenariospezifischen Optimierungen und Konzeptevaluierung zusammengefasst. Ein Ausblick beschließt die Arbeit.

Schlüsselmanagementkonzept: Für das automatische Gruppenschlüsselmanagement wurde das Konzept Multicast Internet Key Exchange (MIKE) eingeführt. Skalierbarkeit bezüglich der Gruppengröße ist durch die Realisierung von zwei Betriebsarten erzielt worden. Der auf dem iterativen Diffie-Hellman-Algorithmus basierende Modus Key Agreement zeichnet sich durch Reparierbarkeit im Fehlerfall, z.B. Verbindungsverlust, aus, weil bei diesem auf den Einsatz eines zentralen Prozesses zur Schlüsselbereitstellung verzichtet wird. Stattdessen wird einem Nutzer zur Koordination der Schlüsselbereitstellung dynamisch der als Transaction Manager bezeichnete Status zugewiesen. Der Modus Key Distribution wird zur Schlüsselbereitstellung in großen Gruppen eingesetzt, wenn die Leistungsfähigkeit des anderen Betriebsmodus dafür nicht ausreicht. Durch die Verwendung von Schlüsselbäumen für beide Modi wird ein effizienter schneller Schlüsselwechsel erreicht. Weitere Synergieeffekte der gemeinsamen Betrachtung beider Schlüsselbereitstellungsansätze bestehen in einer einfachen Realisierung beider Ansätze und der Möglichkeit, zwischen diesen umzuschalten. In wissenschaftlicher Hinsicht wird mit dem Konzept gezeigt, dass eine strikte Trennung beider Schlüsselbereitstellungsarten nicht sinnvoll und eine Kombination der beiden Verfahren sogar möglich ist. Dadurch werden die Vorteile beider Verfahren, d.h. die hohe Effizienz des Modus Key Distribution in großen Gruppen und die Reparierbarkeit des Modus Key Agreement genutzt.

Fehlertoleranz: Mechanismen zur Fehlertoleranz des Schlüsselmanagements MIKE wurden vor dem Hintergrund des Fehlermodells Zusammenbruchsfehler, hervorgerufen durch dauerhaften Verbindungsverlust oder Prozessabsturz, entworfen. Um bei diesem Fehlermodell eine Fehlertoleranz des Systems zu erzielen, wurde für den Zusammenbruchsfehler des Transaction Managers im Modus Key Agreement ein Mechanismus definiert, der es den Teilnehmern ermöglicht, sich auf einen neuen Transaction Manager zu einigen. Um das Schlüsselverwaltung MIKE in Kommunikationsnetzwerken mit funkbasierten Verbindungen einsetzen zu können, muss zusätzlich eine Toleranz gegenüber temporären Kommunikationsstörungen, d.h. Paketverlusten, gewährleistet werden. Zur Abschwächung deren Auswirkungen werden vier verschiedene Mechanismen genutzt. Eine Unempfindlichkeit gegen temporäre Kommunikationsfehler beim An- und Abmeldevorgang wird durch Übertragungswiederholungen erzielt. Eine Absicherung der Schlüsselübermittlung beim Schlüsselwechsel erfolgt durch eine Vorwärtsfehlerkorrektur. Die periodische Erneuerung des Gruppenschlüssels, d.h. dessen begrenzte Lebensdauer, in Kombination mit einer Sequenznummerüberwachung wirkt als Keep-Alive-Mechanismus für die Nutzer. Der Mechanismus nimmt an, dass ein Nutzer vorübergehend seine Verbindung verloren hat, wenn er einen Gruppenschlüssel mit ungültiger Sequenznummer erhält. Nach einer erneuten

Gruppenanmeldung erhalten vom temporären Kommunikationsfehler betroffene Teilnehmer wieder den aktuellen Gruppenschlüssel.

Szenariospezifische Optimierungen: Zur weiteren Anpassung des Schlüsselmanagements MIKE an die militärische Verwendung wurde die Sammelverarbeitung von Nutzeranfragen und die nutzerverhaltensbasierte Schlüsselbaumkonstruktion untersucht. Ergebnis der Forschungsarbeiten ist die Übertragung einer geringeren Datenmenge zur Etablierung eines neuen Gruppenschlüssels pro Teilnehmeroperation. Weil die Mechanismen durch eine verbesserte Verarbeitung des Schlüsselbaums realisiert wurden, konnte eine gleichzeitige Optimierung beider Betriebsmodi erzielt werden. Die ressourcengesteuerte Auswahl des Transaction Managers wurde entwickelt, um auch im Modus Key Agreement den Betrieb mit Nutzern, die zeitweise nur über eine Simplexkommunikation verfügen, zu ermöglichen. Allerdings kann der Mechanismus auch dazu verwendet werden, Nutzern, die nur über eine geringe Datenübertragungskapazität und Rechenleistung verfügen, nicht die ressourcenintensive Aufgabe des Transaction Managers zu übertragen.

Konzeptevaluierung: Zur Verifikation der Tragfähigkeit des Konzeptes MIKE wurde dieses evaluiert. Schwerpunkt der durchgeführten Analysen waren Untersuchungen zur Effizienz des Schlüsselwechsels mittels Simulationen und einer experimentellen Implementierung des Konzepts. Die bei der Effizienzanalyse potentiell einsetzbaren Metriken wurden erstmals systematisch analysiert und dann eine geeignete Metrik ausgewählt. Um die Schlüsselverwaltung MIKE bei der Effizienzanalyse mit einer Last zu konfrontieren, wurden drei Generatoren für Nutzerverhalten realisiert. Diese ermöglichen die Bewertung des Schlüsselmanagements in verschiedenen Einsatzfeldern und erlauben Schlussfolgerungen über die Skalierbarkeit des Konzepts. Die durchgeführte Effizienzanalyse ergab, dass das als Betriebsmodus Key Agreement entworfene verteilte Schlüsselbereitstellungsverfahren insbesondere in Netzwerken mit limitierter Datenübertragungskapazität eine schnellere Schlüsselbereitstellung als bestehende Konzepte ermöglicht. Weiterhin konnte die Flexibilität des Konzepts nachgewiesen werden, indem gezeigt wurde, dass durch Umschalten auf den zweiten Betriebsmodus auch in Netzwerken mit geringer Datenübertragungskapazität eine Schlüsselbereitstellung trotz steigender Gruppengröße möglich ist.

Ausblick: In dem vorgestellten Gruppenschlüsselmanagement MIKE muss der Gruppenbeitritt bzw. Gruppenaustritt durch einen Administrator initiiert werden. Durch automatische Entdeckung im Netzwerk verfügbarer Prozesse zum Nutzdatenschutz und eine Weiterleitung dieser Informationen an das Schlüsselmanagement könnte automatisch ein Gruppenbeitritt bzw. –austritt durchgeführt werden. Die derzeit noch benötigten Administratoreingriffe würden dadurch weiter vermindert. Dieser Ansatz zur weiteren Automatisierung des Schutzes von Multicast-Nutzdaten wird bereits im Rahmen eines deutsch-amerikanischen Projekts aufgegriffen.
Bei den bisherigen Untersuchungen wurden die beiden Betriebsmodi immer getrennt betrachtet. Ein Simultanbetrieb beider Betriebsmodi ist bisher noch nicht vorgesehen. Dies bewirkt, dass ein Schlüsselmanagementsystem im Betriebsmodus Key Distribution keinen gemeinsamen Gruppenschlüssel mit einem Schlüsselmanagementsystem im Modus Key Agreement vereinbaren kann. Die Möglichkeit, einen gemeinsamen Gruppenschlüssel für Domänen mit unterschiedlichen Schlüsselmanagementverfahren bereitzustellen, bildet den

praktischen Hintergrund dieser Forschungsaktivität. Technisch ist hierzu das Problem der Kopplung zweier Schlüsselbäume zu lösen.
Im Rahmen dieser Arbeit wurden betriebsmodusspezifische Optimierungen nicht betrachtet. In Kapitel 7 wurden nur solche Effizienzsteigerungen betrachtet, die beide Betriebsmodi verbessern. Sowohl für den Betriebsmodus Key Agreement, z.B. [Lia06], als auch den Modus Key Distribution, z.B. [Kwa06], existieren betriebsmodusspezifische Optimierungskonzepte. Derartige Konzepte können zur weiteren Effizienzverbesserung in das Schlüsselmanagement MIKE integriert werden. Bei der Integration ist zu untersuchen, ob eine Veränderung der Schlüsselbaumdefinition in Abschnitt 3.1 notwendig ist und das Umschalten zwischen den Betriebsmodi weiterhin möglich ist.
In Abschnitt 7.4 wurde die nutzerverhaltensbasierte Schlüsselbaumkonstruktion zur Effizienzsteigerung der Gruppenschlüsselbereitstellung für beide Betriebsmodi vorgeschlagen. Hierbei wurde ausgenutzt, dass in dem Einsatzbereich des Schlüsselmanagements die Nutzer ein bekanntes Verhaltensmuster aufweisen. Eine Optimierung der Schlüsselbaumkonstruktion hinsichtlich anderer Kriterien wäre ebenfalls denkbar. Sind zum Beispiel durch ein Global Positioning System (GPS) die Positionen der Nutzer bekannt, können Nutzer eines Gebiets in einen Teilbaum des Schlüsselbaums eingefügt werden. Der Ausschluss eines oder mehrerer Nutzer einer geographischen Region wäre dann sehr effizient möglich.
Die Operation EJECT wird verwendet, um einen kompromittierten Nutzer auszuschließen. Um zu verhindern, dass ein derartiger Nutzer wieder der Gruppe beitritt, muss eine entsprechende Zertifikatswiderrufsliste manuell generiert und verteilt werden. Soll dieses automatisch erfolgen, ist die Group Policy Database (vgl. Abschnitt 4.5) an ein Policy Based Network Management (PBNM) anzuschließen. Ein derartiges System ist zum Beispiel in [Zeb06] beschrieben. Der Widerruf eines Zertifikats, d.h. Erzeugung und Verteilung einer Zertifikatswiderrufsliste, benötigt erhebliche Zeit. Mechanismen zur Überbrückung dieser Zeitspanne könnten dann ebenfalls von einem Policy Based Network Management bereitgestellt werden. Weiterhin könnte ein PBNM benötigte Zertifikate automatisch von einem Verzeichnisdienst laden. Zertifikate werden zum Beispiel von einem Verzeichnisdienst gemäß dem X.500-Standard [Itu93] zur Verfügung gestellt.
Die Sichere Inter-Netzwerk Architektur (SINA) [Bun06] ist eine vom Bundesamt für Sicherheit in der Informationstechnik (BSI) entwickelte IT-Architektur zur Übertragung von hoch schützenswerten Informationen in unsicheren Netzen. Durch die Kombination von Thin-Client/Server- und Virtual-Private-Network-Technologie können mit SINA flexible, hochsichere Systeme realisiert werden. SINA kann bisher nur zum Schutz von Systemen mit Punkt-zu-Punkt-Kommunikation eingesetzt werden. Nach einer Integration des Schlüsselmanagements MIKE wäre es zusätzlich möglich, SINA auch bei Systemen, die die hoch schützenswerten Informationen mittels Multicast übertragen, einzusetzen. Ein weiteres potentielles Einsatzgebiet des Schlüsselmanagements MIKE ist die Möglichkeit, es als Baustein des Sicherheitsmoduls eines Software Defined Radio zu verwenden. Dies sind durch Software konfigurierbare Funksysteme. In Rahmen einer Studie [Bro06] mit Industriebeteiligung wurde die Einsetzbarkeit des Konzepts MIKE bereits nachgewiesen.

10 Abkürzungsverzeichnis

A

ABAM On-Demand Associativity-Based Multicast
AES Advanced Encryption Standard
AH Authentication Header
AKD Area Key Distributor
ASM Any Source Multicast

B

BD Burmester-Desmedt Group Diffie-Hellman
BE Broadcast Encryption
bk Blind Key
BSI Bundesamt für Sicherheit in der Informationstechnik

C

CAMP Core-Assisted Mesh Protocol
CBT Core Based Tree Protocol
CCNP Cryptographic Context Negotiation Protocol
CCNT Cryptographic Context Negotiation Template
CFT Centralized Flat Table
CKA Conference Key Agreement
CST Cipher Sequences Technique

D

DCCM Dynamic Cryptographic Context Negotiation Management
DEP Dual Encryption Protocol
DFT Distributed Flat Table
DH Diffie-Hellman
DiRK Distributed Registration and Key distribution
DKD Domain Key Distributor
DLKH Distributed Logical Key Hierarchy
DOFT Distributed One-way Function Tree
DoS Denial of Service
DPG Dynamic Peer Group
DVMRP Distance Vector Multicast Routing Protocol

E

ELK Efficient Large Group Key
EMCON Emission Control
EMSS Efficient Multicast Stream Signature
ESP Encapsulating Security Payload

EVS Extendend Virtual Synchrony

F

FEC Forward Error Correction
FIFO First-In-First-Out

G

GC Group Controller
GCKS Group Controller/Key Server
GCS Group Communication System, Group Communication System
GDH Group Diffie-Hellman
GDOI Group Domain of Interpretation
GKMNM Group Key Management with Network Mobility
GKMP Group Key Management Protocol
GM Group Manager
GO Group Owner
Gothic Group Access Control Architecture
GPS Global Positioning System
GSA Group Security Association
GSAKMP Group Secure Association Key Management Protocol
GSC Group Security Controller
GSI Group Security Intermediarie
GSM Gobal System for Mobile Communication
GSPTv1 Group Security Policy Token Version 1

H

HIP Hierarchical Multicast Routing
HTC Hierarchical a-ary Tree with Clustering

I

ICMPv6 Internet Control Message Protocol Version 6
IDP IPSec Discovery Protocol
IETF Internet Engineering Task Force
IGKMP Inter-Domain Group Key Management
IGMP Internet Group Management Protocol
IKE Internet Key Exchange
IPDL Ismene Policy Description Language
IPSec IP Security
IPv4 Internet Protocol Version 4
IPv6 Internet Protocol Version 6
ISAKMP Internet Security Association and Key Management Protocol
ITW Ingemarsson-Tang-Wong Group Diffie-Hellman

K

KD Key Download
KEK Key Encryption Key
KHIP Keyed Hierarchical Multicast Routing
KT Key Tree

L

LG Large Group
LKH Logical Key Hierarchy
LSM Link State Multicast

M

M Member
MAC kryptographischer Message Authentication Code
MANET Mobiles Ad hoc Netzwerk
MAODV Multicast Ad hoc On-Demand Distance Vector Routing
MD5 Message Digest Algorithm 5
MESP Multicast Encapsulating Security Payload
MGCS Mobile Group Controller Scheme
MIKE Multicast Internet Key Exchange
MLD Multicast Listener Discovery
MOLSR Multicast Optimized Link State Routing
MOSPF Multicast Open Shortest Path First

N

NTP Network Time Protocol

O

OCBT Ordered Core Based Tree Protocol
OFCT One-way Function Chain Tree
OFT One-way Function Tree
OSI Model Open Systems Interconnection Model
OSPF Open Shortest Path First

P

PBNM Policy Based Network Management
PGK Pre-distributed Group Key
PIM Protocol Independent Multicast
PKI Public Key Infrastructure
PSS Pre-Positioned Secret Sharing

R

RFC Request for Comments

RP Rendezvous Point
RPM Reverse Path Multicast
RSA Rivest Shamir Adleman

S

SA Security Association
SAD Security Association Database
SAK SA Key Encryption Key
SAKM Scalable and Adaptive Key Management Scheme
SAT SA Traffic Encryption Key
SDR Software Defined Radio
SEQ Sequence Number
SGKDP Synchronized Group Key Distribution Protocol
SHA-1 Secure Hash Algorithm 1
SIM-KM Scalable Infrastructure for Multicast Key Management
SINA Sichere Inter-Netzwerk Architektur
SMKD Scalable Multicast Key Distribution
SPBM Scalable Position-Based Multicast
SPD Security Policy Database
SPoF Single Point of Failure
SRTP Secure Real-time Transport Protocol
SSKEK Shared Secret Key Encryption Key
SSM Source Specific Multicast
STL Sub Tree Leader
STR Skinny Tree Protokoll
STW Steiner-Tsudik-Waidner Group Diffie-Hellman

T

TDES Triple Data Encryption Standard
TEK Traffic Encryption Key
TESLA Timed Efficient Stream Loss-tolerant Authentication
TGDH Tree-based Group-Diffie-Hellman
TM Transaction Manager
TMD TM Download
Totem-M Totem-Membership
Totem-SR Totem-Single-Ring
TTR Time To Repair

U

UDP User Datagram Protocol
UTC Coordinated Universal Time, Coordinated Universal Time

V

VoIP Voice over IP
VS Virtual Synchrony
VTKD Virtual Token-based Key Distribution

W

WAN Wide Area Network

11 Literaturverzeichnis

[Ada05] A. Adams, J. Nicholas, W. Siadak, Request for Comments 3973: Protocol Independent Multicast - Dense Mode (PIM-DM) Specification, Internet Engineering Task Force, 2005

[Aga94] D. A. Agarwal, Totem: A Reliable Ordered Delivery Protocol for Interconnected Local-Area Networks, PhD Thesis, University of California, Santa Barbara, 1994

[Alm96] K. C. Almeroth, M. H. Ammar, Collecting and Modeling the Join/Leave Behavior of Multicast Group Members in the Mbone, Proceedings of the Symposium on High Performance Distributed Computing, 1996, page 209

[Ami02] Y. Amir, Y. Kim, C. Nita-Rotaru, G. Tsudik, On the Performance of Group Key Agreement Protocols, 22nd International IEEE Conference on Distributed Computing Systems, 2002

[Atk98] S. Kent, R. Atkinson, Request for Comments 2406: IP Encapsulating Security Payload (ESP), Internet Engineering Task Force, 1998

[Ami92] Y. Amir, D. Dolev, S. Kramer, D. Malki, Transis: A communication sub-system for high availability, 22nd International Symposium on Fault-Tolerant Computing Systems, 1992, pages 76–84

[Aur03] T. Aurisch, C. Karg, A Daemon for Multicast Internet Key Exchange, Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN '03), 2003, pages 368 ff.

[Aur04] T. Aurisch, Using key trees for securing military multicast communication, Unclassified Proceedings of the IEEE Milcom 2004, 2004

[Aur05] T. Aurisch, Optimization technique for military multicast key management, Unclassified Proceedings of the IEEE Milcom 2005, 2005

[Bal01] A. Ballardie, Request for Comments 2201: Core Based Trees (CBT) Multicast Routing Architecture, Internet Engineering Task Force, 1997

[Bal02] A. Ballardie, Request for Comments 2189: Core Based Trees (CBT version 2) Multicast Routing -- Protocol Specification --, Internet Engineering Task Force, 1997

[Bal96] A. Ballardie, Request for Comments 1949: Scalable Multicast Key Distribution, Internet Engineering Task Force, 1996

[Bal99] D. M. Balenson, D. K. Branstad, D. A. McGrew, J. W. Turner, M. Heyman, Cryptographic Context Negotiation Template, Dynamic Cryptographic Context Management (DCCM) Report #2 Version 2, TIS Labs at Network Associates, (Cryptographic Technologies Group), 1999

[Bau03] M. Baugher, R. Canetti, P. Cheng, P. Rohatgi, MESP: A Multicast Framework for the IPsec ESP, INTERNET-DRAFT draft-ietf-msec-mesp-01.txt, Internet Engineering Task Force, 2003

[Bec98] C. Becker, U. Wille, Communication complexity of group key distribution, Proceedings of the 5th ACM Conference on Computer and Communications Security, 1998, pages 1-6

[Ber01] M.Y. Li, R. Poovendran, C. Berenstein, Optimization of key storage for secure multicast, Proceedings of the Conference on Information Science and Systems, 2001, pages 771-774

[Bir94] K. P. Birman, R. V. Renesse, Reliable Distributed Computing with the Isis Toolkit, IEEE Computer Society Press, 1994

[Bou04] F. Boutin, M. Hascoet, Cluster Validity Indices for Graph Partitioning, 8th IEEE International Conference on Information Visualisation (IV'04), 2004, pages 376-381

[Boy97] C. Boyd, On key agreement and conference key agreement, Australasian Conference on Information Security and Privacy, 1997, pages 294–302

[Bri99] B. Briscoe, MARKS: Zero side-effect multicast key management using arbitrarily revealed key sequences, Proceedings of the First International Workshop on Networked Group Communication, 1999

[Bun04] Bundesministerium der Verteidigung, Grundzüge der Konzeption der Bundeswehr, 2004

[Bun06] Bundesamt für Sicherheit in der Informationstechnik, Systembeschreibung Sichere Inter-Netzwerk Architektur (SINA), http://www.bsi.de/fachthem/sina/sysbesch/sysbesch.htm, 2006

[Bur94] M. Burmester, Y. Desmedt, A secure and efficient conference key distribution system, Advances in Cryptology – Eurocrypt'94, 1994, pages 119-129

[Cai00] T. Hardjono and B. Cain, Key Establishment for IGMP Authentication in IP Multicast, IEEE ECUMN, 2000

[Can00] R. Canetti, B. Pinkas, A taxonomy of multicast security issues, INTERNET-DRAFT draft-irtf-smug-taxonomy-01.txt, Internet Engineering Task Force, 2000

[Can99] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, Multicast security: A taxonomy and some efficient constructions, Proceedings of the INFOCOM '99, 1999

[Cha00] T. Chai-Keong, G. Guillermo, B. Santithorn, On-demand associativity-based multicast routing for ad hoc mobile networks (ABAM), Proceedings of the 52nd IEEE VTS Vehicular Technology Conference (VTC), 2000, vol. 3, pages 987-993

[Cha04] Y. Challal, H. Bettahar, A. Bouabdallah, SAKM: A Scalable and Adaptive Key Management Approach for Multicast Communications, ACM SIGCOMM Computer Communications Review, 34(2), 2004, pages 55–70

[Chi01] G. Chaddoud, I. Chrisment, A. Shaff, Dynamic Group Communication Security, Proceedings of the 6th IEEE Symposium on computers and communication, 2001, pages 49-56

[Chi89] G. H. Chiou, W. T. Chen, Secure Broadcast using Secure Lock, IEEE Transactions on Software Engineering, 15(8), 1989, pages 929-934

[Chu02] H. Chu, L. Qiao, K. Nahrstedt, H. Wang, R. Jain, A secure multicast protocol with copyright protection, ACM SIGCOMM Computer Communication Review, 32(2), 2002, pages 42-60

[Col06] A. Colegrove, H. Harney, Group Security Policy Token v1, INTERNET-DRAFT draft-ietf-msec-policy-token-sec-06.txt, Internet Engineering Task Force, 2006

[Col99] J. Collura, US Secure Voice Communications, NATO Ad Hoc Working Group on Narrow Band Voice Coding, European Briefings, 1999

[Con98] A. Conta, S. Deering, Request for Comments 2463: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6), Internet Engineering Task Force, 1998

[Cro95] T. Ballardie, J. Crowcroft, Multicast-specific security threats and counter-measures, Proceedings of the 1995 Symposium on Network and Distributed System Security (SNDSS'95), 1995, pages 2 ff.

[Dec01] B. DeCleene, L. Dondeti, S. Griffin, T. Hardjono, D. Kiwior, J. Kurose, D. Towsley, S. Vasudevan, C. Zhang, Secure group communications for wireless networks, Unclassified Proceedings of the IEEE Milcom 2001, 2001, pages 113–117

[Dee89] S. Deering, Request for Comments 1112: Host Extensions for IP Multicasting, Internet Engineering Task Force, 1989

[Dee99] S. Deering, W. Fenner, Request for Comments 2710: Multicast Listener Discovery (MLD) for IPv6, Internet Engineering Task Force, 1999

[Del06] H. Um, E. J. Delp, A Secure Group Key Management Scheme for Wireless Cellular Networks, Proceedings of the Third International Conference on Information Technology: New Generations (ITNG '06), 2006, pages 414-419

[Dif76] W. Diffie, M. E. Hellman, New directions in cryptography, IEEE Transactions on Information Theory, IT-22, 6, 1976, pages 644-654

[Din00] P. T. Dinsmore, D. M. Balenson, M. Heyman, P. S. Kruus, C. D. Scace, A.T. Sherman, A.T., Policy-based security management for large dynamic groups: An Overview of the DCCM project, Proceedings of the DARPA Information Survivability Conference and Exposition 2000 (DISCEX '00), volume 1, 2000, pages 64-73

[Don99] L. Dondeti, S. Mukherjee, A. Samal, A Distributed Group Key Management Scheme for Secure Many-to-many Communication, Technical Report PINTL-TR-207-99, Department of Computer Science, University of Maryland, 1999

[Don00] L. R. Dondeti, S. Mukherjee, A. Samal, Scalable secure one-to-many group communication using dual encryption, Computer Communications, 23(17), 2000, pages 1681–1701

[Eas05] D. Eastlake 3rd, Request for Comments 4305: Cryptographic Algorithm Implementation Requirements Encapsulating Security Payload (ESP) and Authentication Header (AH), Internet Engineering Task Force, 2005

[Ech98] K. Echtle, J. G. Silva, Fehlerinjektion - ein Mittel zur Bewertung der Maßnahmen gegen Fehler in komplexen Rechensystemen, Informatik Spektrum 21(6), 1998

[Esk06] Crypto++ Library, http://www.eskimo.com/~weidai/cryptlib.html, 2006

[Est98] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei, Request for Comments 2362: Protocol Independent Multicast-Sparse Mode (PIM-SM) Specification, Internet Engineering Task Force, 1998

[Fek01] A. Fekete, N. Lynch, A. Shvartsman, Specifying and using a partitionable group communication service, ACM Transactions on Computer Systems, vol. 19, 2001

[Fen02] B. Fenner, Protocol Independent Multicast-Sparse Mode (PIM-SM) Protocol Specification (Revised), INTERNET-DRAFT draft-ietf-pim-sm-v2-new-05.txt Internet, Engineering Task Force, 2002

[Fen97] W. Fenner, Request for Comments 2236: Internet Group Management Protocol, Internet Engineering Task Force, 1997

[Fer92] D. F. Ferraiolo, D. R. Kuhn, Role Based Access Control, Proceedings of the 15th NIST-NCSC National Computer Security Conference, 1992, pages 554-563

[Fia93] A. Fiat, M. Naor, Broadcast encryption, In Advances in Cryptology - CRYPTO '93, Springer-Verlag, 1994, pages 480-491

[Gam98] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns, Elements of Reusable Object-Oriented Software, Addison-Wesley, 1998

[Gär01] F. C Gärtner, Formale Grundlagen der Fehlertoleranz in verteilten Systemen, Dissertation an der TU Darmstadt Fachbereich Informatik, 2001

[Gar99] J. J. Garcia-Luna-Aceves, E. L. Madruga, The Core Assisted Mesh Protocol, IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks, vol. 17, no. 8, 1999, pages 1380-1394

[Geh06] C. Gehrmann, M. Näslund, ECRYPT Yearly Report on Algorithms and Keysizes, ECRYPT (European Network of Excellence for Cryptology), 2006

[Gin06] T. Ginzler, Implementierung und Bewertung von Schlüsselmanagementverfahren in Rechnernetzen, Diplomarbeit Rheinische Friedrich-Wilhelm-Universität Bonn, Institut für Informatik IV, 2006

[Gol01] P. Golle, N. Modadugu, Authenticating Streamed Data in the Presence of Random Packet Loss, ISOC Network and Distributed System Security Symposium, 2001, pages 13-22

[Har00] T. Hardjono, R. Canetti, M. Baugher, P. Dinsmore, Secure IP Multicast: Problem areas, Framework, and Building Blocks, INTERNET-DRAFT draft-irtf-smug-framework-01.txt, Internet Engineering Task Force, 2000

[Har01] H. Harney, A. Colgrove, P. McDaniel, Principles of Policy in Secure Groups, Internet Society, Proceedings of the Network and Distributed System Security Symposium, 2001, pages 554-563

[Har03] M. Baugher, B. Weis, T. Hardjono, H. Harney, Request for Comments 3547: The Group Domain of Interpretation, Internet Engineering Task Force, 2003

[Har06] H. Harney, U. Meth, A. Colegrove, G. Gross, Request for Comments 4535: GSAKMP: Group Secure Association Key, Internet Engineering Task Force, 2006

[Har97] H. Harney, C. Muckenhirn, Group Key Management Protocol (GKMP) Architecture, Request for Comments 2093, Engineering Task Force, 1997

[Hin03] R. Hinden, S. Deering, Request for Comments 3513: Internet Protocol Version 6 (IPv6) Addressing Architecture, Internet Engineering Task Force, 2003

[Ing82] I. Ingemarson, D. Tang, C. Wong, A Conference Key Distribution System, IEEE Transactions on Information Theory, 28(5), 1982, pages 714–720

[Isi06] The Network Simulator ns-2, http://www.isi.edu/nsnam/ns, 2006

[Itu93] International Telecommunications Union, The Directory - overview of concepts, models and service, International Telecommunications Union X.500 series of Recommendations, 1993

[Jac01] P. Jacquet, P. Minet, A. Laouiti, L. Viennot, T. Clausen, C. Adjih, Multicast Optimized Link State Routing, INTERNET-DRAFT draft-jacquet-olsr-molsr-00.txt, Internet Engineering Task Force, 2001

[Jud02] P. Q. Judge, M. H. Ammar, Gothic: Group Access Control Architecture for Secure Multicast and Anycast, Proceedings of the IEEE INFOCOM, 2002

[Kau05] C. Kaufman, Request for Comments 4306: Internet Key Exchange (IKEv2) Protocol, updates RFC 2407/2408/2409, Internet Engineering Task Force, 2005

[Ken98] S. Kent, R. Atkinson, Request for Comments 2402: IP Authentication Header (AH), Internet Engineering Task Force, 1998

[Kih98] K. P. Kihlstrom, L. E. Moser, P. M. Melliar-Smith, The SecureRing Protocols for Securing Group Communication, Proceedings of the 31th Annual Hawaii International Conference on System Sciences, 1998, pages 317-326

[Kim00] Y. Kim, A. Perrig, G. Tsudik, Simple and fault-tolerant key agreement for dynamic collaborative groups, 7th ACM Conference on Computer and Communications Security, 2000, pages 235-244

[Kwa06] D. Kwak; S. J. Lee; J. W. Kim; E. Jung, An efficient LKH tree balancing algorithm for group key management, IEEE Communications Letters, 10(3), 2006, pages 222-224

[Lev99] P. Levi, U. Rembold, Einführung in die Informatik für Naturwissenschaftler und Ingenieure, Carl Hanser Verlag, 1999

[Lia06] L. Liao, M. Manulis, Tree-Based Group Key Agreement Framework for Mobile Ad-Hoc Networks, Proceedings of the 20th International Conference on Advanced Information Networking and Applications - Volume 2 (AINA'06), 2006, pages 5-9

[Liu05] F. Liu, H. Koenig, Secure and efficient key distribution for collaborative applications, Proceedings of International Conference on Collaborative Computing, Networking, Applications and Worksharing, 2005, pages 11 ff.

[Mau98] D. Maughan, M. Schertler, M. Schneider, J. Turner, Request for Comments 2408: Internet Security Association and Key Management (ISAKMP), Internet Engineering Task Force, 1998

[McD98] D. McDonald, C. Metz, B. Phan, Request for Comments 2367: PF_KEY Key Management API Version 2, Internet Engineering Task Force, 1998

[Mie01] P. van Mieghem, G. Hooghiemstra, R. van der Hofstad, On the Efficiency of Multicast, IEEE/ACM Transaction on Networking 9, 2001

[Mil92] D. Mills, Request for Comments 1305: Network Time Protocol (Version 3) Specification and Analysis, Internet Engineering Task Force, 1992

[Mit97] S. Mittra, Iolus: a framework for scalable secure multicasting, Proceedings of the ACM conference on Applications, technologies, architectures, and protocols for computer communication (SIGCOMM '97), 1997, pages 277-288

[Mol00] R. Molva, A. Pannetrat, Scalable multicast security with dynamic recipient groups, ACM Transactions on Information and System Security (TISSEC), 3(3), 2000, pages 136-160

[Mos94] L. E. Moser, Y. Amir, P. M. Melliar-Smith, D. A. Agarwal, Extended virtual synchrony, Proceedings of the 14th IEEE International Conference on Distributed Computing Systems, pages 56-65

[Moy94] J. Moy, Request for Comments 1584: Multicast Extensions to OSPF, Internet Engineering Task Force, 1994

[Muk04] R. Mukherjee, J.W. Atwood, SIM-KM: Scalable Infrastructure for Multicast Key Management, Proceedings of the IEEE Local Computer Networks - LCN'04, 2004, pages 335–342

[Nas04] M. Baugher, D. McGrew, M. Naslund, E. Carrara, K. Norrman, Request for Comments 3711: The Secure Real-time Transport Protocol (SRTP), Internet Engineering Task Force, 2004

[Nis01] National Institute of Standards and Technology, Federal Information Processing Standards Publication 197: Specification for the Advanced Encryption Standard (AES), 2001

[Nis95] National Institute of Standards and Technology, Federal Information Processing Standards Publication 180-1: Secure Hash Standard (SHA), 1995

[Opp96] R. Oppliger, A. Albanese, Distributed registration and key distribution (DiRK), Proceedings of the 12th International Conference on Information Security IFIP SEC'96, 1996

[Per00] A. Perrig, R. Canetti, J.D. Tygar, D. Song, Efficient Authentication and Signing of Multicast Streams over Lossy Channels, IEEE Symposium on Security and Privacy, 2000, pages 56-73

[Per01] A. Perrig, R. Canetti, D. Song, J. D. Tygar, Efficient and Secure Source Authentication for Multicast, Network and Distributed System Security Symposium, 2001, pages 35-46

[Pra99] P. McDaniel, A. Prakash, P. Honeyman, Antigone: A Flexible Framework for Secure Group Communication, Proceedings of of the 8th USENIX Security Symposium, Washington D.C., 1999, pages 99-114

[Pra00] P. McDaniel, A. Prakash, Ismene: Provisioning and Policy Reconciliation in Secure Group Communication, Technical Report CSE-TR-438-00, Electrical Engineering and Computer Science, University of Michigan, 2000

[Poo01] R. Poovendran, J. Baras, An Information Theoretic Approach for Design and Analysis of Rooted-Tree-Based Multicast Key Management Schemes, IEEE Trans. Information Theory, Vol. 47(7), 2001, pages 2824-2834

[Raf02] S. Rafaeli, D. Hutchison, Hydra: A Decentralised Group Key Management, WETICE '02: Proceedings of the 11th IEEE International Workshops on Enabling Technologies, 2002, pages 62-67

[Rei94] M. K. Reiter, K. P. Birman, R. van Renesse, A security architecture for fault-tolerant systems, ACM Transactions on Computer Systems (TOCS), 12(4), 1994, pages 340-371

[Ren96] R. V. Renesse, K.Birman, S. Maffeis, Horus: A flexible group communication system, Communications of the ACM, vol. 39, 1996, pages 76–83

[Riv78] R. Rivest, A. Shamir, L. Adleman, A Method for Obtaining Digital Signatures and Public Key Cryptosystems, Communications of the ACM 21(2), 1978, pages 120-126

[Riv92] R. Rivest, Request for Comments 1321: The MD5 message-digest algorithm, Internet Engineering Task Force, 1992

[Riz97] L. Rizzo, Effective erasure codes for reliable computer communication protocols, ACM Computer Communication Review, 27(2), 1997, pages 24-36

[Rod00] O. Rodeh, K. Birman, D. Dolev, Optimized group rekey for group communication systems, Proceedings of ISOC Network and Distributed Systems Security Symposium, 2000

[Roy99] E. M. Royer, C. E. Perkins, Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol, Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), 1999, pages 207 - 218,

[Sel02] A. A. Selçuk, D. Sidhu, Probabilistic optimization techniques for multicast key management, Computer Networks: The International Journal of Computer and Telecommunications Networking, 40(2), 2002, pages 219-234

[Set00] S. Setia, S. Koussih, S. Jajodia, E. Harder, Kronos: A Scalable Group Re-Keying Approach for Secure Multicast, IEEE Symposium on Security and Privacy, 2000, pages 215-228

[She03] A. T. Sherman, D. A. McGrew, Key establishment in large dynamic groups using one-way function trees, IEEE Transactions on Software Engineering, 29(5), 2003, pages 444-458

[Shi96] C. Shields, Ordered core based trees, Master's thesis, University of California - Santa Cruz, 1996

[Shi98] C. Shields, J.J. Garcia-Luna-Aceves, The HIP Protocol for Hierarchical Multicast Routing, Proceedings of Seventeenth Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Puerto Vallarta, Mexico, 1998, pages 257-266

[Shi99] C. Shields, J. J Garcia-Luna-Aceves, KHIP—a scalable protocol for secure multicast routing, Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols For Computer Communication (SIGCOMM '99), 1999, pages 53-64

[Smi98] R. E. Smith, Internet-Kryptographie, Addison-Wesley-Longman Verlag, 1998

[Smo02] V. Smotlacha, One-way Delay Measurement Using NTP, CESNET Technical Report, 2002

[Son01] A. Perrig, D. Song, J. D. Tygar, ELK, a New Protocol for Efficient Large-Group Key Distribution, SP '01: Proceedings of the 2001 IEEE Symposium on Security and Privacy, 2001, pages 247 ff.

[Sta98] Y. Amir, J. Stanton, The Spread wide area group communication system, Tech. Rep. 98-4, John Hopkins University, Center of Networking and Distributed Systems, 1998

[Ste88] D. Steer, L.L. Strawczynski, W. Diffie, M. Weiner, A Secure Audio Teleconference System, CRYPTO'88, 1988

[Ste96] M. Steiner, G. Tsudik, M. Waidner, Diffie-Hellman key distribution extended to group communication, Proceedings of the 3rd ACM Conference on Computer and Communications Security, 1996, pages 31–37

[Sun05] W.H.D Ng, Z. Sun, H. Cruickshank, Group key management with network mobility, 13th IEEE International Conference on Networks2005, Jointly held with the 05 IEEE 7th Malaysia International Conference on Communication, volume 2, 2005, pages 6 ff.

[Tha06] THALES COMMUNICATIONS GmbH, SEM 93E/93/91 - Das Softwaregesteuerte Sprach- und Datenfunksystem, Datenblatt, 2006

[Toe02] J. Tölle, Intrusion Detection durch strukturbasierte Erkennung von Anomalien im Netzwerkverkehr, Dissertation an der Rheinischen Friedrich-Wilhelms-Universität Bonn, GCA-Verlag, 2002

[Tra04] M. Transier, H. Füßler, J. Widmer, M. Mauve, W. Effelsberg, Scalable Position-Based Multicast for Mobile Ad-hoc Networks, Proceedings of the 1st International Workshop on

Broadband Wireless Multimedia: Algorithms, Architectures and Applications (BroadWim), 2004

[Tro06] QT, http://www.trolltech.de, 2006

[Wai88] D. Waitzman, C. Partridge, S. E. Deering, Request for Comments 1075: Distance Vector Multicast Routing Protocol, Internet Engineering Task Force, 1988

[Wal99] M. Waldvogel, G. Caronni, D. Sun, N. Weiler, B. Plattner, The VersaKey Framework: Versatile Group Key Management, IEEE Journal on Selected Areas in Communications (Special Issues on Middleware), 17(8), 1999, pages 1614–1631

[Wan05] W. Trappe, Y. Wang, K. J. R. Liu, Resource-aware conference key establishment for heterogeneous networks, IEEE/ACM Transactions on Networking, 13(1), 2005, pages=134-146

[Wei00] L. Wei, Authenticating PIM version 2 messages, INTERNET-DRAFT draft-ietf-pim-v2-auth-01.txt, Internet Engineering Task Force, 2000

[Wei06] B. Weis, Request for Comments 4359: The Use of RSA/SHA-1 Signatures within Encapsulating Payload (ESP) and Authentication Header (AH), Internet Engineering Task Force, 2006

[Won98] C. K. Wong, M. G. Gouda, S. S. Lam, Secure group communications using key graphs, Proceedings of the ACM SIGCOMM '98, 1998, pages 68-79

[Won99] C. K. Wong, S. S. Lam, Digital signatures for flows and multicasts, IEEE/ACM Transactions on Networking (TON), 7(4), 1999, pages 502-513

[Yan01] X. S. Li, Y. R. Yang, M. G. Gouda, S. S. Lam, Batch Rekeying for Secure Group Communications, Proceedings of the 10th International World Wide Web Conference, 2001, pages 68-79

[Zeb06] G. M. Pérez, A. F. G. Skarmeta, S. Zeber, J. Spagnolo, T. Symchych, Dynamic Policy-based Network Management for a Secure Coalition Environment, IEEE Communications Magazine, 2006, pages 58-64

Zusätzlich zu den oben aufgelisteten Publikationen wurde die nachfolgende Literatur mit eingeschränkter Verfügbarkeit verwendet:

[Bro06] C. Broecker, Software Design Description IFM-E, Rohde&Schwarz, 2006

[Cit05] G. Dürr, Abschätzung der Häufigkeit von Schlüsselwechseln im militärischen Einsatz, Arbeitspapier FGAN/FKIE-KOM, 2005

[Ebe03] M. Eberhard, K. Billmann, T. Karzelek, D. Rupprecht, H. Seifert, M. Zenz, Studie Mobile Kommunikation im Heer, ZASBw - DezStudPlBw, 2003

[Lan02] W. Langer, Prioritätengesteuerte Sprachkommunikation über paketbasierte Netzwerke, FKIE-Bericht Nr. 50, 2002

[Sei06] H. Seifert et al., Task 2 Final Report, Interoperable Networks for Secure Communications II, 2006

12 Anhang

Im Folgenden sind die Zustandsautomaten der beiden Betriebsmodi des Schlüsselmanagements MIKE als Graph dargestellt. Hierbei ist jeder Zustand durch ein Knoten gekennzeichnet. Die Kanten beschreiben die Zustandsübergänge. Von jedem Knoten gehen so viel gerichtete Kanten aus, wie es Eingaben gibt. Eine Kante endet bei demjenigen Knoten, in dem der Automat beim Lesen der Eingabe als nächstes übergeht. Eine Kante ist durch Eingabe und Ausgabe gekennzeichnet. Eingabe und Ausgabe werden durch einen Schrägstrich getrennt. Zur Vereinfachung sind manche Kanten mit mehreren Eingaben bzw. Ausgaben beschriftet. Weiterhin ist in dem Zustandsautomaten des Modus Key Agreement der Mechanismus Neuwahl des TM (vgl. Abschnitt 4.6.1) nicht enthalten.

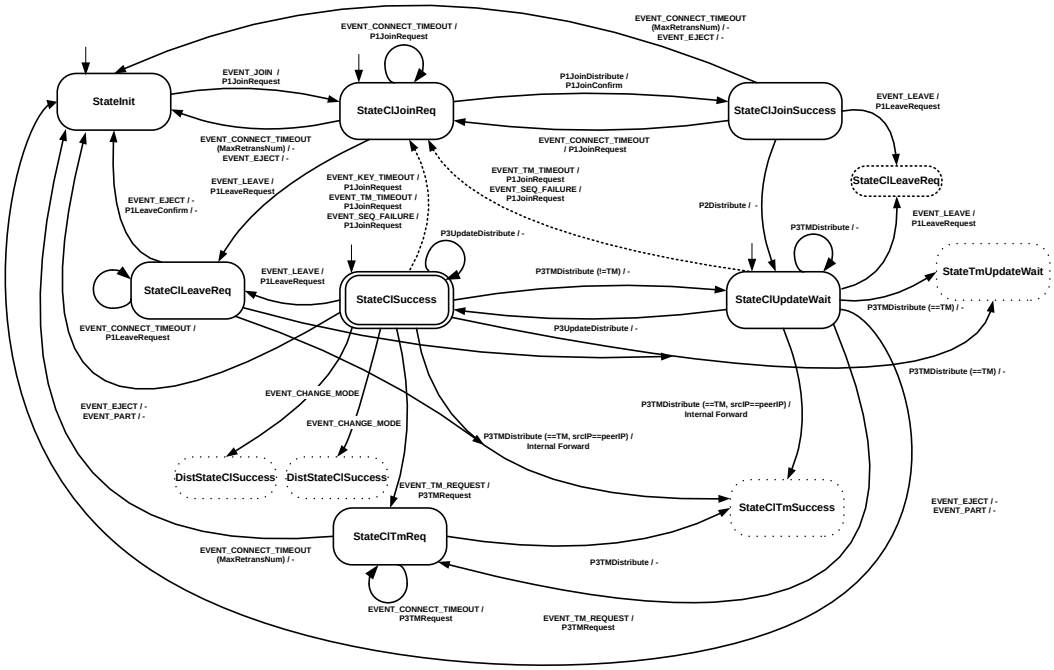

Abbildung 121: Zustandsautomat eines Nutzers im Modus Key Agreement

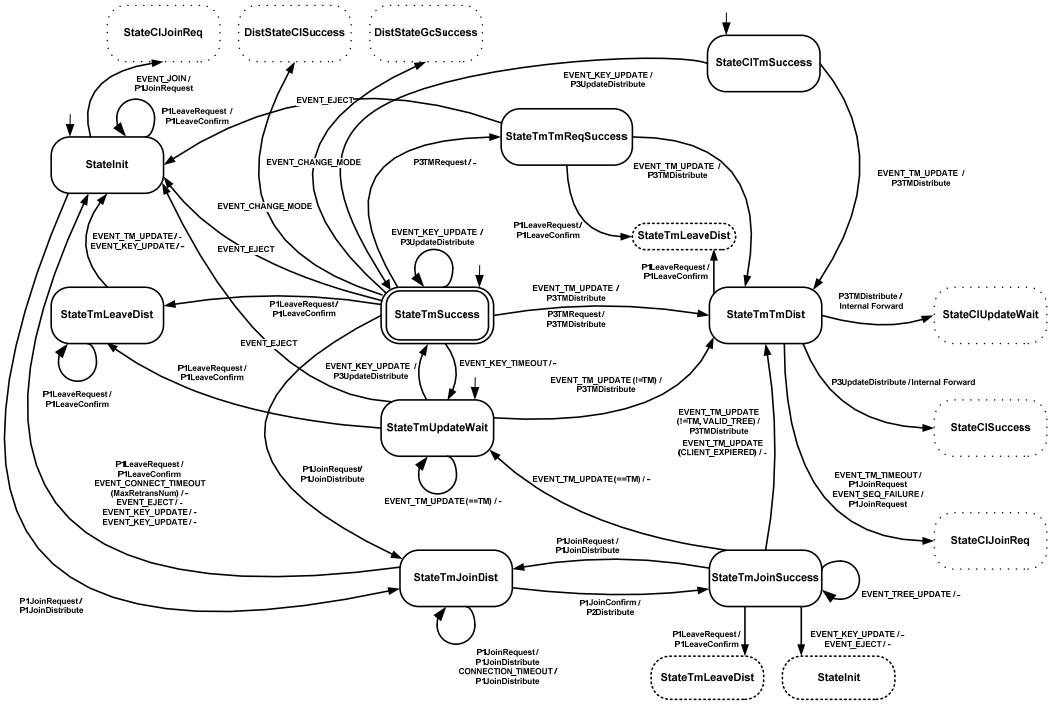

Abbildung 122: Zustandsautomat des TM im Modus Key Agreement

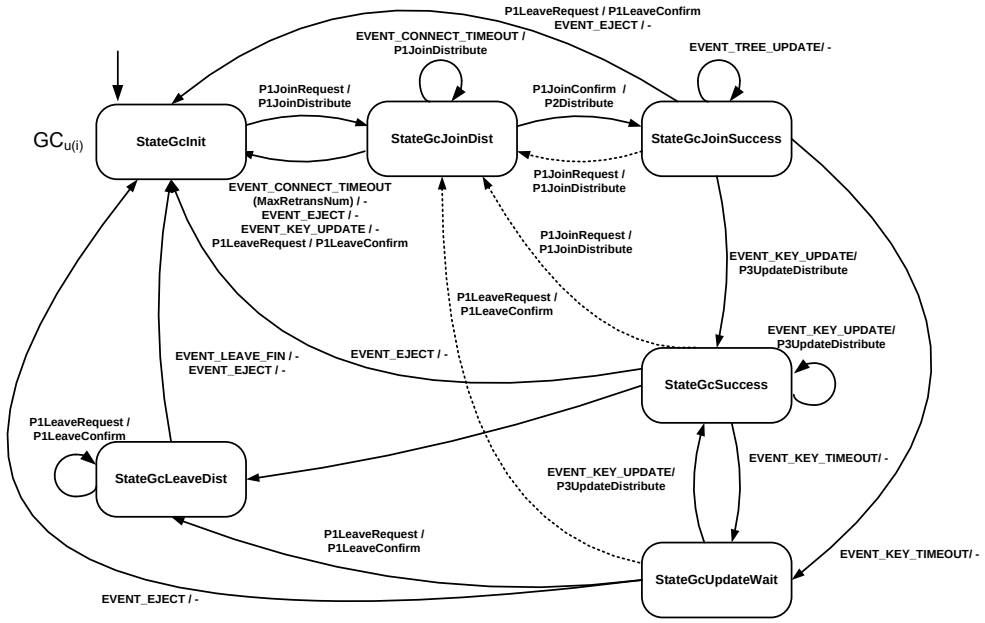

Abbildung 123: Zustandsautomat des GC im Modus Key Distribution

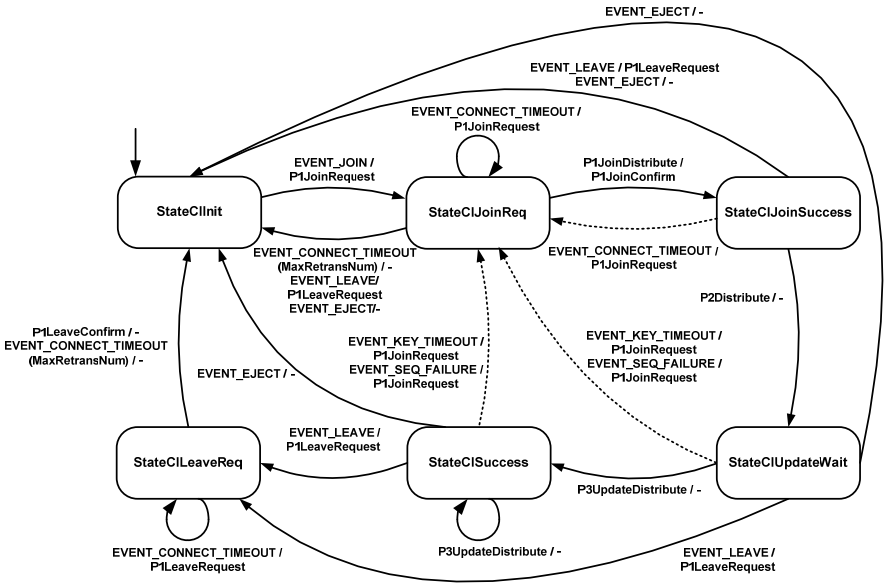

Abbildung 124: Zustandsautomat eines Nutzers im Modus Key Distribution