Analyse, Optimierung und Evaluierung eines sicheren verteilten Dienstverzeichnisses für drahtlose Sensornetze

Diplomarbeit am Institut für Telematik
Prof. Dr. M. Zitterbart
Fakultät für Informatik
Universität Karlsruhe (TH)

von

cand. inform.
Ingmar Baumgart

Betreuer:
Prof. Dr. M. Zitterbart
Dipl.-Inform. H.-J. Hof

Tag der Anmeldung: 1. November 2004
Tag der Abgabe: 30. April 2005

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

Karlsruhe, den 30. April 2005

Inhaltsverzeichnis

1. Einleitung

In den letzten Jahren hat der technische Fortschritt zunehmend die Integration von Sensoren, Mikroprozessoren, Sendeeinheiten und Energiequellen auf kleinstem Raum ermöglicht. Die dadurch entstandenen Geräte werden Sensorknoten genannt. Durch die Vernetzung solcher Sensorknoten könnten in naher Zukunft auch größere Bereiche in unwegsamem Gelände kostengünstig sensorisch erfasst und beobachtet werden. Zudem eignen sich die Knoten aufgrund ihrer geringen Größe für die Anbringung an Stellen, an denen die Verwendung großer Geräte stören würde. Mögliche Anwendungsbereiche für Sensornetze sind somit unter anderem die Umweltbeobachtung, die Patientenüberwachung im Gesundheitswesen sowie die Koordination und Überwachung von Großveranstaltungen.

In vielen dieser Anwendungsszenarien spielt das Thema Sicherheit eine bedeutende Rolle. Im medizinischen Bereich können Sensorknoten beispielsweise zur Überwachung von Herzschlag und Atmung oder zur Steuerung lebenserhaltender Systeme eingesetzt werden. Für den praktischen Einsatz ist daher eine Absicherung der Kommunikation gegen Manipulation durch Dritte unabdingbar. Um die Privatsphäre der Patienten zu schützen, sollten die Sensordaten zudem auch verschlüsselt werden. Die stark begrenzte Rechen- und Batteriekapazität der Sensorknoten erfordert den Entwurf von energieeffizienten kryptographischen Algorithmen und Protokollen, die direkt auf die Eigenschaften von Sensornetzen zugeschnitten sind.

In dienstorientierten Sensornetzen bieten die einzelnen Sensoren und Aktoren verschiedene Dienste an. Durch die Verknüpfung der Dienste können komplexere Dienste generiert und verwendet werden. Auf diese Weise können Sensornetze auch in neuen Anwendungsbereichen wie beispielsweise der Gebäudeautomation oder der Altenpflege eingesetzt werden.

1.1 Zielsetzung der Arbeit

Zielsetzung der Diplomarbeit ist die Analyse, Optimierung und Evaluierung eines sicheren verteilten Dienstverzeichnisses für drahtlose Sensornetze. Das Dienstverzeichnis ist ein Bestandteil der „Karlsruher Sensornetz Plattform“ [HuHZ04], einer flexiblen, modularen Architektur für dienstorientierte Sensornetze.

Das zu untersuchende Dienstverzeichnis „Secure Content Addressable Networks Version 2“ (SCANv2)[HoBZ04] beruht auf einem Overlay-Netzwerk mit zusätzlichen Sicherheitsmechanismen. Die Evaluierung des Dienstverzeichnisses soll mit Hilfe einer Implementierung in einem Simulator erfolgen.

1.2 Gliederung der Arbeit

Im zweiten Kapitel werden die Grundlagen für diese Diplomarbeit erläutert. Dabei wird kurz auf die Themen Sensornetze, Dienstverzeichnisse sowie Angreifermodelle und Kryptographie eingegangen.

Im dritten Kapitel wird das Konzept von SCANv2 vorgestellt und die Sicherheit des Konzepts auf Grundlage der zuvor vorgestellten Angreifermodelle untersucht.

Eine optimierte Version, SCANv3, die einige der Schwachstellen von SCANv2 beseitigt, wird im vierten Kapitel vorgestellt.

Im fünften Kapitel werden Besonderheiten der Implementierung im Simulator beschrieben.

Abschließend wird die Implementierung im sechsten Kapitel anhand der Simulationsergebnisse analysiert und bewertet.

2. Grundlagen

Im Folgenden werden die zum Verständnis der Arbeit notwendigen Grundlagen erläutert. Nach der Klärung des Begriffs „Sensornetze“ wird kurz auf das Thema Dienstverzeichnisse eingegangen. Im Anschluss werden die erforderlichen Grundlagen aus dem Bereich Sicherheit vermittelt. Einen Schwerpunkt bilden hier verschiedene Angriffsszenarien.

2.1 Sensornetze

Sensorknoten sind kleine, kostengünstige Geräte, die aus Sensoren, Aktoren, Prozessoren und Speicher bestehen. Eine weit verbreitete Sensornetz-Plattform ist der an der Universität Berkeley entwickelte MICA Mote [HiCu02]. Eine moderne Variante dieses Motes verwendet einen ATmega 128L 8-bit Mikroprozessor, besitzt 512kB Flashspeicher und kann Daten mit bis zu 250kbps über Distanzen von mehreren Metern übertragen. An der FU Berlin wurde im Rahmen des Scatterweb-Projekts ebenfalls ein Sensorknotenmodul (ESB) entwickelt [ScLR]. Eine Besonderheit des ESB ist die Möglichkeit den Knoten mit Hilfe einer Solarzelle zu betreiben.

Für viele Anwendungen müssen Sensorknoten jedoch noch wesentlich kleiner[1] und kostengünstiger werden. Das *Smart Dust*-Projekt [WLLP01] der Universität Berkeley beschäftigt sich mit der Fragestellung mit welchen Verfahren diese Miniatursensorknoten hergestellt werden können und welche technologischen Probleme noch zu lösen sind.

Mit Hilfe der Sensoren können die Knoten verschiedene Umweltparameter messen und auf den gemessenen Werten Berechnungen durchführen. Durch die Vernetzung hunderter oder tausender solcher Knoten entstehen so genannte drahtlose Sensornetze [MaRö03].

Durch die Verknüpfung der Daten vieler unterschiedlicher Sensoren können Umweltphänomene, wie beispielsweise die Bewegung eines Objekts, erfasst und beobachtet werden. Falls zusätzlich einige der Knoten mit Aktoren ausgestattet sind, können die gewonnenen Sensordaten auch zu Steuerungszwecken verwendet werden. Mögliche

[1] in der Größenordnung weniger Kubikmillimeter

Anwendungsbereiche für Sensornetze finden sich in der Medizin, dem Umweltschutz, dem Katastrophenschutz sowie in der Automatisierungstechnik. Ein Beispiel für den erfolgreichen Einsatz eines Sensornetzes in der Zoologie ist das Great Duck Island Projekt. Dort wurde der Lebensraum von Seevögeln mit Hilfe von Sensorknoten beobachtet und das Brutverhalten der Tiere auf einer kleinen Insel studiert [MCPS+02]. Da die Tiere sehr scheu sind und die Anwesenheit von Menschen auf der Insel großen Einfluss auf das Verhalten der Vögel gehabt hätte, wurde dort eindrucksvoll gezeigt, welches Potential Sensornetze bereits in diesem frühen Stadium besitzen.

Aufgrund der geringen Größe der Module sind die Ressourcen eines Sensorknotens stark begrenzt. Insbesondere die knappen Energieressourcen erfordern energieeffiziente Hardwarekomponenten sowie entsprechend ausgelegte Betriebssysteme und Kommunikationsprotokolle. Um die Lebenszeit des gesamten Sensornetzes zu verlängern, sollte der Energieverbrauch möglichst so zwischen den einzelnen Knoten aufgeteilt werden, dass ein frühzeitiger Ausfall einzelner Knoten vermieden wird. Da der zur Verfügung stehende Speicher der Sensorknoten ebenfalls stark limitiert ist, müssen die anfallenden Daten, die ein Sensorknoten speichern muss, ebenfalls auf ein Minimum reduziert werden.

Eine weitere Herausforderung stellt die unvorhersehbare Dynamik der Netze dar. Knoten können aufgrund von Umwelteinflüssen oder erschöpfter Batterien ausfallen. Um die Genauigkeit der Messdaten zu verbessern oder um ausgefallene Knoten zu ersetzen, können jederzeit neue Knoten in der Netz eingebracht werden. Des Weiteren sind Szenarien mit mobilen Sensorknoten denkbar, die zu ständigen Änderungen in der Netzwerktopologie führen. Da sich die Anzahl der Knoten, die ein Sensornetz bilden, je nach Anwendungszenario um mehrere Größenordnungen unterscheiden kann und Netze mit tausenden Knoten denkbar sind, müssen alle verwendeten Verfahren und Kommunikationsprotokolle auf Skalierbarkeit ausgelegt sein.

In sensiblen Anwendungsbereichen wie in der Medizin- oder Militärtechnik ist ein zuverlässiges Funktionieren des Sensornetzes unabdingbar. Daher ist ein robustes Verhalten gegenüber Ausfällen oder Störungen einzelner Knoten erforderlich. Hier spielt auch die Robustheit gegenüber Angriffen bösartiger Knoten eine bedeutende Rolle. Dieses Ziel kann nur mit Hilfe ausgeklügelter, ressourcensparender Sicherheitsmechanismen erreicht werden (siehe Abschnitt 2.4).

Aufgrund der denkbaren Einsatzszenarien für Sensornetze müssen diese ohne zentrale Infrastruktur auskommen. Dies hat zur Folge, dass Sensornetze ähnlich wie Adhoc-Netze somit selbstkonfigurierend, selbstorganisierend sowie selbstheilend sein müssen.

Bei Sensornetzen kann zwischen zwei Netzwerkstrukturen unterschieden werden: Beim klassischen **datenorientierten** Sensornetz sendet eine große Anzahl Sensorknoten ihre Messdaten zu einer gemeinsamen Senke. Auf dem Weg von der Quelle zur Senke können die Daten bereits von Zwischenknoten aggregiert werden. Dadurch wird das Datenvolumen reduziert und notwendige Kommunikation energieeffizienter gestaltet. Abbildung 2.1 zeigt eine solche Baumtopologie mit einer Basisstation.

In **dienstorientierten** Sensornetzen bieten die einzelnen Sensoren und Aktoren bestimmte Dienste an, z.B. „Temperatursensor in Raum x“ oder „Ventilsteuerung für y“. Im Gegensatz zu einem datenorientierten Sensornetz erfolgt der Datenfluss

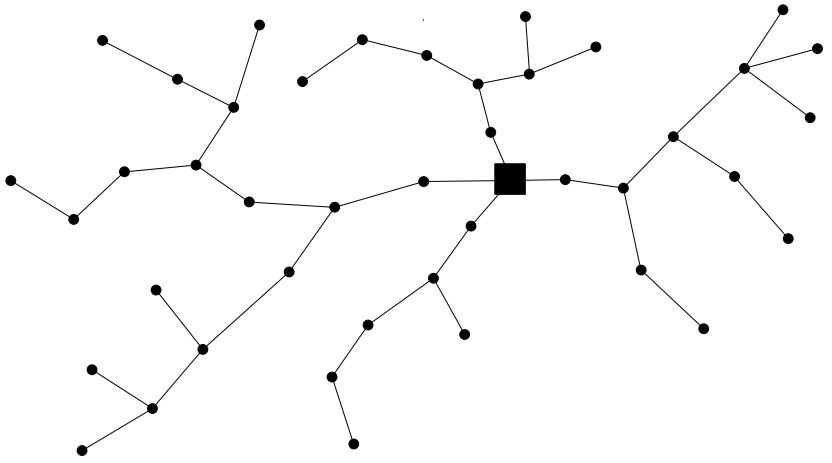

Abbildung 2.1: Struktur eines datenorientierten Sensornetzes mit einer Basisstation

hier in der Regel zwischen mehreren Sensoren und Aktoren. Durch die Verknüpfung mehrerer einfacher Dienste kann so dynamisch ein neuer komplexerer Dienst entstehen. Dieser Ansatz bietet ein hohes Maß an Flexibilität für Netze mit häufig wechselnden Sensorknoten. Ein so generierter Dienst kann hierbei auch die Form eines datenorientierten Sensornetzes besitzen (siehe Abbildung 2.2). Da die Wahl eines Kommunikationspartners dynamisch aufgrund der angebotenen Dienste erfolgt, muss ein Mechanismus geschaffen werden, mit dessen Hilfe Dienste angeboten und aufgefunden werden können. Üblicherweise werden hierfür so genannte Dienstverzeichnisse verwendet. In Abschnitt 2.2 werden einige Ansätze für Dienstverzeichnisse kurz vorgestellt.

In datenorientierten Sensornetzen spielt die Identität einzelner Sensorknoten in der Regel keine große Rolle. Die Adressierung kann auch hier über die angebotenen Dienste erfolgen. Beim Ausfall eines Knotens kann dieser dann gegebenenfalls einfach durch einen anderen Knoten, der den gleichen Dienst anbietet, ersetzt werden.

2.2 Dienstverzeichnisse

Dienstverzeichnisse werden zum Veröffentlichen, Speichern und Auffinden von Diensten in Kommunikationsnetzen verwendet. Durch deren Verwendung kann der Administrationsaufwand, der üblicherweise beim Hinzufügen neuer Dienste entsteht, reduziert und eine dynamische Konfiguration ermöglicht werden. Insbesondere in Netzen mit hoher Mobilität und ständig wechselnden Diensten ist eine statische Konfiguration der Endgeräte nicht mehr praktikabel. Dienstverzeichnisprotokolle bieten dem Benutzer üblicherweise die Möglichkeit nach einzelnen Diensten zu suchen, alle verfügbaren Dienste aufzulisten, einen Dienst aufgrund bestimmter Parameter auszuwählen, sowie in einigen Fällen einen Mechanismus, um einen gefundenen Dienst zu nutzen.

Der klassische Ansatz ein Dienstverzeichnis zu realisieren, besteht in der Verwendung eines zentralen Verzeichnisservers. Ein bekanntes Beispiel für einen weitgehend zentrales Verzeichnis ist der Internetdienst DNS zum Auflösen von Rechnernamen. Da Sensornetze selbstorganisierend und robust gegenüber dem Ausfall einzelner Knoten sein sollen und in der Regel keine Infrastruktur zur Verfügung steht, scheidet hier

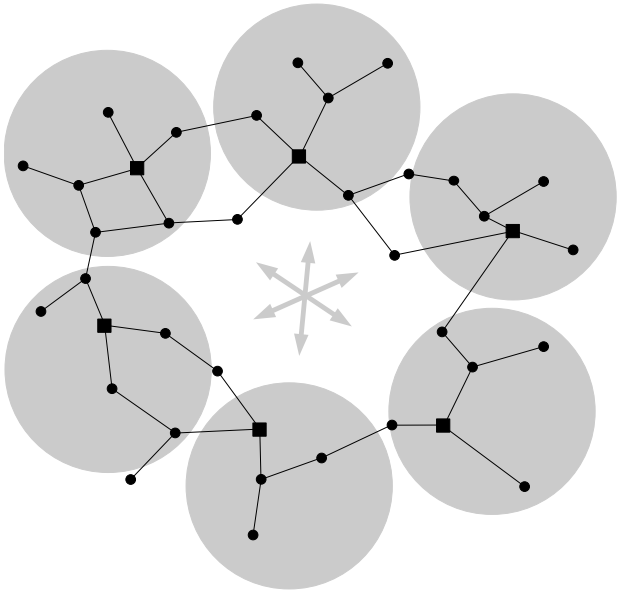

Abbildung 2.2: Struktur eines dienstorientierten Sensornetzes, bei welchem einzelne Dienste durch datenorientierte Sensornetze gebildet werden

die Verwendung einer zentralen Instanz jedoch aus. Des Weiteren muss ein Dienstverzeichnis für Sensornetze für den Einsatz in Multi-Hop-Netzen ausgelegt sein und möglichst sparsam mit den beschränkten Ressourcen der Sensorknoten umgehen. Sensornetze sind aufgrund des Einsatzes drahtloser Telekommunikationstechnologien und deren oftmals erforderlichen Platzierung im öffentlichen Raum der Gefahr eines Angriffs ausgesetzt (siehe Abschnitt 2.4). Daher sollte ein Dienstverzeichnis für Sensornetze mit entsprechenden Sicherheitsmechanismen ausgestattet sein.

In [BeRe00] werden einige gängige Dienstverzeichnisse miteinander verglichen. Das von der IETF spezifizierte *Service Location Protocol (SLP)*[Gutt99] ist ein sprachunabhängiges Protokoll für den Einsatz in IP-Netzen. Dienste werden von so genannten *Service-Agents (SA)* publiziert, worauf diese von *User-Agents (UA)* abgefragt werden können. Das Protokoll sieht optional die Verwendung zentraler *Directory-Agents (DA)* vor, die Dienstbeschreibungen von Service-Agents speichern und Dienstanfragen von User-Agents beantworten. Wird SLP in dezentralen Netzen verwendet, verbreiten die Service-Agents ihre Dienstbeschreibungen periodisch mittels Multicast an alle vorhandenen User-Agents. Alternativ können die User-Agents mit einer *Service Request* Anfrage an die SLP-Multicastadresse selbst gezielt bestimmte Dienste anfordern. SLP bietet keine Sicherheitsmechanismen um das Verzeichnis vor Angreifern zu schützen, sondern verlässt sich auf Sicherheitsmaßnahmen der IP-Schicht wie zum Beispiel IPSec.

Bluetooth sieht die Verwendung des *Bluetooth Service Discovery Protocol (SDP)* [SIG99] vor, um Dienste in der Umgebung eines Bluetooth-Geräts aufzufinden. Das Protokoll wurde für den Einsatz in Bluetooth-Piconetzen entworfen und bietet keine Unterstützung für Multi-Hop-Netze.

Microsofts *Universal Plug and Play (UPnP)*[PlFo03] wurde für den Einsatz in kleinen Büro- und Heimnetzen entworfen und verwendet das *Simple Service Discovery Protocol (SSDP)* um Dienste aufzufinden. Dienstbeschreibungen werden in einer

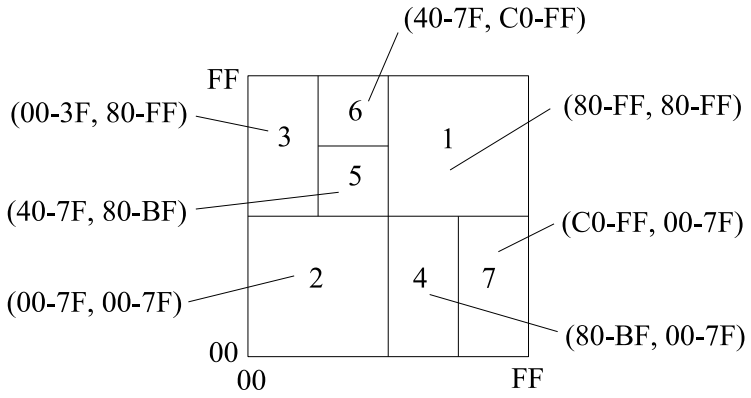

Abbildung 2.3: Aufgefalteter 2-dimensionaler CAN-Koordinatenraum mit 7 Knoten

XML-Beschreibungssprache formuliert und können per HTTP abgefragt werden. Neu hinzukommende Knoten verbreiten eine kurze Ankündigung ihres Dienstes per HTTPMU (HTTP Multicast over UDP). Mit Hilfe von SOAP[2] können auf den beitretenden Knoten Parameter geändert oder erweiterte Informationen über angebotene Dienste angefordert werden.

Ein Protokoll, das speziell für den Einsatz in Ad-hoc-Netzen entworfen wurde, ist das *Secure Service Discovery Protocol (SPDP)*[AlCa03]. Das Auffinden von Diensten erfolgt durch Fluten von *Service Requests*. Um den Kommunikationsaufwand zu reduzieren, können Dienstbeschreibungen in einem lokalen Cache zwischengespeichert werden. Bestandteil des Protokolls ist eine Public-Key-Infrastruktur auf Basis eines Reputationssystems, die für Authentifizierung und Autorisierung der Knoten verwendet wird. Integrität und Vertraulichkeit der Daten werden mittels IPsec sichergestellt.

Alle genannten Protokolle eignen sich nicht für die Verwendung in Sensornetzen, da sie zentrale Komponenten verwenden, keine Unterstützung für Multi-Hop-Netze bieten oder das Dienstverzeichnis nicht ausreichend vor Angriffen geschützt wird. Das Protokoll SPDP erfüllt zwar die genannten Anforderungen, eignet sich aber ebenfalls nicht für die Verwendung in Sensornetzen, da der vorgeschlagene Authentifizierungsmechanismus asymmetrische Kryptoverfahren verwendet (siehe Abschnitt 2.4.2.2).

2.3 CAN - ein inhaltsadressierbares Netzwerk

Verteilte Hashtabellen (distributed hash tables, kurz DHTs) wie sie zum Beispiel durch Chord [SMLNK+03], Pastry [RoDr01] oder Tapestry [ZhKJ01] realisiert werden, sind in den letzten Jahren zum intensiven Forschungsgegenstand geworden. In modernen Peer-to-Peer Systemen werden immer öfter DHTs zur Verwaltung von Overlaynetzen eingesetzt. *Overlaynetze* sind logisch Netze, die die Infrastruktur eines darunterliegenden Netzes, des so genannten *Underlays*, verwenden. Um Datensätze der Form *(Schlüssel, Wert)* verteilt in einem Netzwerk abzulegen und aufzufinden, werden die Schlüssel mit Hilfe einer Hashfunktion auf kurze Hashwerte abgebildet. Der Wertebereich der Hashfunktion wird dabei als *Koordinatenraum* der DHT

[2]Simple Object Access Protocol

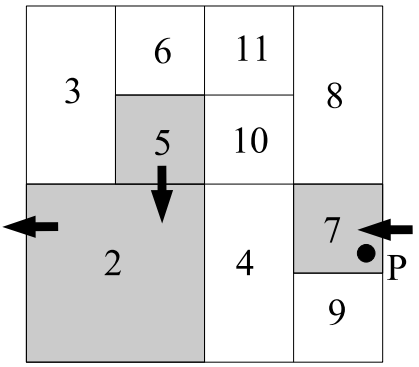

Abbildung 2.4: Routing im CAN-Overlay von Knoten 5 zu Knoten 7

bezeichnet. Jedem Overlay-Knoten wird ein Teil dieses Raums zugeteilt. Der Teilnehmer ist dann für die Speicherung aller Datensätze zuständige, deren Hashwerte der Schlüssel in dessen Wertebereich liegt. Da das dieser Arbeit zugrundeliegende Dienstverzeichnis SCANv2 auf dem System CAN aufsetzt, wird im Folgenden die Funktionsweise dieser DHT kurz beschrieben.

In dem inhaltsadressierbaren Netzwerk CAN [RFHK+01] (content addressable network) wird der Koordinatenraum auf eine d-dimensionale Torusoberfläche abgebildet, welche in so genannte *Zonen* unterteilt wird. Jede Zone ist dabei genau einem bestimmten Overlayknoten zugeordnet, der als *Zone-Owner* der Zone bezeichnet wird. Abbildung 2.3 zeigt eine mögliche Zonenaufteilung in einem 2-dimensionalen CAN mit 7 Knoten. Zur einfacheren Darstellung ist der Torus aufgefaltet dargestellt. Die Schlüssel der zu verwaltenden Datensätze werden mit Hilfe einer Hashfunktion auf Punkte im Koordinatenraum abgebildet. Der Knoten, in dessen Zone der abgebildete Punkt fällt, ist für die Speicherung des Datensatzes zuständig. Die Overlay-Routingtabelle eines Knotens besteht aus einer Liste aller Knoten, die benachbarte Zonen verwalten, und enthält deren Netzwerkadressen und Zonenkoordinaten. Zwei Zonen sind *benachbart*, falls deren Koordinaten in $d - 1$ Dimensionen überlappen und in einer Dimension aneinander angrenzen.

Das Auffinden eines Datensatzes erfolgt durch schrittweises Weiterleiten einer Suchanfrage an denjenigen Nachbarknoten, dessen Zonenkoordinaten dem Schlüsselhashwert am nächsten liegen, bis die Anfrage schließlich den Zielknoten erreicht. In Abbildung 2.4 ist ein Beispiel für das Routing im CAN-Overlay dargestellt. In diesem Beispiel möchte Knoten 5 einen Datensatz auffinden. Der Hashwert des Datensatzschlüssels ergibt den Punkt P. Da die benachbarte Zone von Knoten 2 dem Zielpunkt P am nächsten liegt, wird die Anfrage zunächst an diesen Knoten gesendet. Die Zone von Knoten 2 ist aufgrund der Torus-Struktur des Overlays direkt mit der Zielzone, die von Knoten 7 verwaltet wird, benachbart. Somit kann Knoten 2 die Anfrage direkt an Knoten 7 weiterleiten, der daraufhin den bei ihm gespeicherten Datensatz an Knoten 5 liefert.

Möchte ein neuer Knoten dem Overlay beitreten, muss diesem ein eigener Teil des Koordinatenraums zugewiesen werden. Dazu halbiert ein bereits vorhandener Knoten seine Zone und überlässt dem neuen Knoten eine der beiden Hälften. Um eine gleichmäßige Verteilung der Zonengrößen zu erreichen wird die zu teilende Zone durch zufällige Wahl eines *Join-Points* im Koordinatenraum bestimmt. Eine gleich-

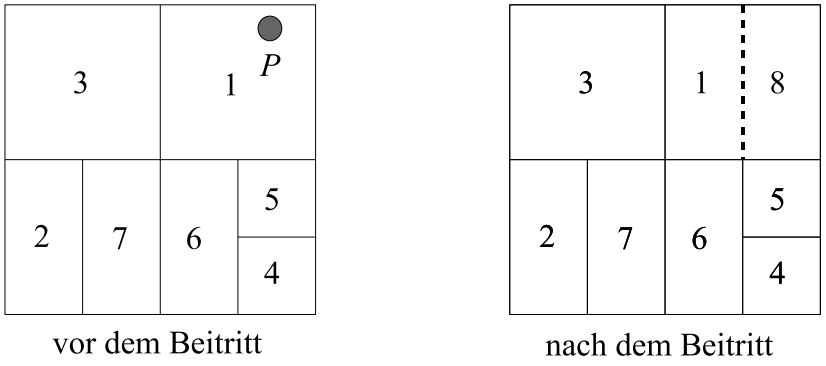

Abbildung 2.5: Beitritt von Knoten 8 zum CAN-Overlay mit Join-Point P

mäßige Verteilung der Zonengrößen ist wichtig um die Anzahl der Overlay-Nachbarn eines Knotens nicht unnötig zu erhöhen und eine gleichmäßige Verteilung der Kommunikationslast zu erreichen. Der beitretende Knoten sendet seinen Beitrittswunsch über das Overlay in Richtung des gewählten Join-Points. Der Zone-Owner, in dessen Zone sich der Join-Point befindet, teilt daraufhin seine Zone und informiert den beitretenden Knoten über dessen neue Overlaynachbarn und Zonenkoordinaten. Ein Beispiel für einen Knotenbeitritt ist in Abbildung 2.5 zu sehen. Der beitretende Knoten 8 sendet dazu eine Overlay-Nachricht an den Punkt P. Knoten 1, in dessen Zone der Punkt P liegt, halbiert daraufhin seine Zone und gibt eine der Zonenhälften an Knoten 8 ab.

Die Zahl der Nachbarn eines Knotens und damit die Größe seiner Routingtabelle ist in einem CAN weitgehend unabhängig von der Anzahl der Knoten im Netzwerk. Sofern die Teilung der Zonen gleichverteilt erfolgt, hängt diese nur von der Dimensionalität d des Torus ab und beträgt konstant $2d$. Aufgrund der Topologie eines CAN existieren zwischen zwei Punkte bis zu d verschiedene Pfade mit annähernd optimaler Pfadlänge. Im Falle von Knotenausfällen stehen somit alternative Wege zum Ziel zur Verfügung. Die durchschnittliche Pfadlänge h zwischen zwei Knoten hängt zusätzlich von der Gesamtzahl der Knoten N ab und kann mit der folgenden Gleichung abgeschätzt [RFHK+01] werden:

$$h = \frac{d}{4} N^{\frac{1}{d}} \tag{2.1}$$

In Abbildung 2.6 ist dieser Zusammenhang nochmals graphisch dargestellt.

Um die Struktur des Overlays auch im Falle von Knotenausfällen konsistent zu halten, sendet ein Knoten periodisch *Update*-Nachrichten an seine Nachbarn. Falls diese Nachrichten ausbleiben, wird dies als Zeichen für einen Knotenausfall gewertet. In diesem Fall wird die Zone des ausgefallenen Knoten vorübergehend von einem seiner Nachbarknoten mitverwaltet. Um die daraus resultierende zunehmende Fragmentierung des Koordinatenraums zu bereinigen, schlagen die Autoren in [RFHK+01] vor gelegentlich Zonen durch geschickte Neuzuweisung wiederzuvereinigen.

Die Performanz eines CAN kann durch eine Reihe von Maßnahmen verbessert werden. Durch die Verwendung mehrerer, verschiedener Hashfunktionen kann ein Datensatz auf verschiedenen Knoten gespeichert werden. Dadurch wird die Last besser

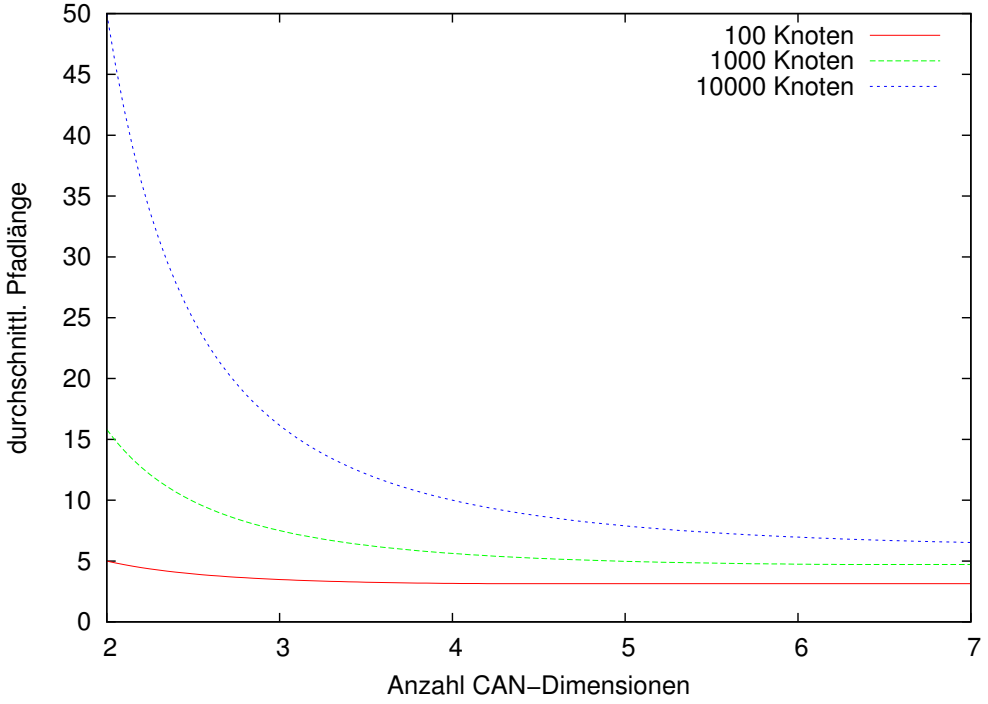

Abbildung 2.6: Durchschnittliche Pfadlänge in einem CAN-Overlay

verteilt und die Robustheit gegenüber Knotenausfällen erhöht. Ein ähnlicher Effekt kann durch das *Überladen* von Zonen erreicht werden. Bei diesem Verfahren wird die gleiche Zone von mehreren Knoten verwaltet, so dass die Verfügbarkeit einer Zone in diesem Fall nicht mehr von der Verfügbarkeit eines einzelnen Knoten abhängt.

2.4 Sicherheit in Sensornetzen

Sensornetze stellen besondere Herausforderungen an den Entwurf einer geeigneten Sicherheitsarchitektur [PeSW04]. Wie bereits erwähnt sind Sensorknoten in ihren Rechen- und Kommunikationsmöglichkeiten stark eingeschränkt und verfügen nur über wenig Speicher. Da Sensorknoten in vielen Fällen an öffentlich zugänglichen Orten eingesetzt werden, besteht zudem die Gefahr, dass die Knoten physischen Angriffen ausgesetzt sind. Zusätzlich besteht wie bei allen drahtlosen Kommunikationstechniken die Gefahr, dass Daten auch von einem weit entfernten Angreifer mittels Richtantennen und empfindlichen Empfängermodulen abgehört und manipuliert werden.

Absolute Sicherheit kann in realen Anwendungen in der Regel nicht garantiert werden, da sich allein die hierzu notwendige Modellierung des Gesamtsystems als zu komplex erweist. Beim Entwurf einer Sensornetzsicherheitsarchitektur muss daher zwischen dem Nutzen (Maß an gewonnener Sicherheit) und den Kosten (Ressourcenverbrauch und Komplexität) abgewogen werden.

Die Bewertung einer Sicherheitsarchitektur für ein konkretes Szenario kann anhand einer Aufstellung geforderter Schutzziele sowie möglicher Angriffsszenarien (siehe Abschnitt 2.4.1) erfolgen. Im Rahmen dieser Arbeit werden die folgenden Schutzziele betrachtet:

- **Integrität:** Der Empfänger einer Nachricht soll überprüfen können, ob die Nachricht verändert wurde.
- **Authentizität:** Der Empfänger soll die Herkunft der Nachricht überprüfen können.
- **Vertraulichkeit:** Der Inhalt der Nachricht soll gegenüber Dritten geheimgehalten werden.
- **Verfügbarkeit:** Die angebotenen Dienste sollen ununterbrochen verfügbar sein.

In der Regel werden diese Ziele durch Verwendung kryptographischer Verfahren erreicht. In Abschnitt 2.4.2 werden die dafür benötigten kryptographischen Grundlagen kurz erläutert.

Ein grundlegendes Problem bei der Verwendung von Verschlüsselungsverfahren ist die Erzeugung und Verteilung des benötigten Schlüsselmaterials (siehe [PeSW04]). Momentan wird davon ausgegangen, dass der Einsatz von Public-Key-Verfahren (siehe Abschnitt 2.4.2.2) in Sensornetzen aufgrund der begrenzten Rechenleistung der Sensorknoten bisher nicht praktikabel ist. Die klassischen Schlüsselaustauschprotokolle wie beispielsweise das Diffie-Hellman-Protokoll [DiHe76], die üblicherweise verwendet werden um spontan zwischen zwei Kommunikationspartnern über einen unsicheren Kanal einen Schlüssel auszutauschen, können daher aufgrund ihres Rechenaufwands in Sensornetzen nicht sinnvoll eingesetzt werden.

Eine in Sensornetzen praktikable Variante besteht im vorherigen Verteilen des Schlüsselmaterials. Den einfachsten Lösungsansatz stellt die Verwendung eines netzwerkweiten, gemeinsamen Schlüssels dar. Jeder Knoten, der im Besitz des Schlüssels ist, gilt als vertrauenswürdig. In diesem Fall kann allerdings die Kompromittierung eines einzelnen Knotens zum Zusammenbruch der gesamten Sicherheitsarchitektur führen. Da die momentan verfügbare Sensorknoten-Hardware keinen besonderen Schutz vor Manipulationen bietet, ist dieser Ansatz für sicherheitskritische Anwendungen nicht geeignet.

Dieses Sicherheitsproblem kann vermieden werden, indem für jedes Knotenpaar ein eigener gemeinsamer Schlüssel verwendet wird. Dieses Verfahren skaliert jedoch nicht, da in einem Netzwerk mit n Knoten jeder Knoten $n - 1$ Schlüssel speichern muss und Sensornetze zum Teil mehrere Tausend Knoten enthalten.

Sofern eine vertrauenswürdige Basisstation vorhanden ist, kann auch diese zur Schlüsselerzeugung verwendet werden. Jeder Knoten benötigt in diesem Fall nur einen gemeinsamen Schlüssel mit der Basisstation. Falls die Basisstation ausfällt oder von einem Angreifer manipuliert wird, bricht allerdings ebenfalls das komplette System zusammen. Des weiteren kann die Verwendung einer zentralen Basisstation zur ungleichmäßigen Verteilung der Kommunikationslast führen und Sensorknoten in der Nähe der Basisstation übermäßig belasten.

Ein viel versprechender neuer Ansatz sind so genannte *random-key predistribution* Protokolle [EsGl02]. Jedem Sensornetzknoten wird dort eine zufällig Untermenge an Schlüsseln aus einem großen vorgenerierten Schlüsselvorrat zugewiesen. Sofern zwei Knoten, die miteinander kommunizieren möchten, bereits einen gemeinsamen

Schlüssel besitzen, wird dieser zur Verschlüsselung der Kommunikation verwendet. Falls dies nicht der Fall ist, muss mit Hilfe ausgewählter[3] Nachbarknoten ein neuer Sitzungsschlüssel erzeugt werden.

Aufgrund der beschränkten Ressourcen der Sensorknoten, stellt sich die Frage, inwieweit die Verwendung kryptographischer Verfahren ohne entsprechende Hardwareunterstützung überhaupt praktikabel ist. Messungen anhand der TinySec-Implementierung [KaSW04] der Universität Berkeley haben ergeben, dass ein Großteil der Leistungseinbußen auf zunehmende Paketgrößen zurückzuführen sind. Die kryptographischen Berechnungen hatten dagegen kaum Einfluss auf die Verzögerung oder den Durchsatz. Um an aussagekräftige Ergebnisse zu gelangen, wurde TinySec implementiert und der Stromverbrauch der eingesetzten Mica2-Sensorknoten bei Verwendung von TinySec gemessen. Bestandteil der Architektur ist die Sicherstellung von Integrität und Vertraulichkeit durch Einsatz symmetrischer Kryptoverfahren auf Sicherungsschicht. Insgesamt wurden durch die Verschlüsselung der Datenpakete nur ein zusätzlicher Ressourcenverbrauch von $5\% - 10\%$ gemessen.

2.4.1 Angriffsszenarien

Im Folgenden wird ein Überblick über mögliche Angriffsszenarien auf Sensornetze und DHTs gegeben. Die Machbarkeit der verschiedenen Angriffe hängt dabei von den Möglichkeiten und dem Wissen der Angreifer ab. Beispielsweise kann bei den Angreifern zwischen Insidern und Outsidern unterschieden werden. Ein Outsider hat keinen besonderen Zugang zum Sensornetz und hat kein Vorwissen über verwendete Schlüssel oder Ähnliches. Ein Insiderangriff erfolgt dagegen durch ein vertrauenswürdiges Mitglied des Sensornetzes. Ein Outsider kann durch Kompromittierung eines Knotens oder durch Zugriff auf geheimes Schlüsselmaterial die Stellung eines Insiders erlangen.

Des Weiteren kann die Leistungsfähigkeit der Knoten, die vom Angreifer eingesetzt werden, eine Rolle spielen: Mit einem Laptop können viele Angriffe realisiert werden, die mit einem Sensorknoten aufgrund der beschränken Ressourcen (insbesondere Rechenkapazität und Sendeleistung) nicht möglich sind.

Die folgende nach Schichten gegliederte Aufstellung möglicher Angriffe auf Sensornetze orientiert sich weitgehend an [KaWa03] und [WoSt02].

2.4.1.1 Angriffe auf den unteren Schichten

Sensorknoten, die sich an leicht zugänglichen Orten befinden, können durch einen Angreifer, der die Funktionalität des Sensornetzes beeinträchtigen möchte, einfach zerstört werden. Sofern viele Knoten betroffen sind, kann dieser Angriff jedoch zumindest erkannt werden. Eine Variante dieses Angriffs, die wesentlich schwerer zu erkennen ist, ist das Umplatzieren von Sensorknoten: Obwohl die Knoten in diesem Fall weiterhin Sensordaten liefern, spiegeln diese nicht mehr die physikalischen Gegebenheiten des Ortes, der überwacht werden sollte, wieder.

Aufgrund der Tatsachen, dass Sensorknoten möglichst klein und kostengünstig sein sollen, kann nicht davon ausgehen werden, dass Knoten in naher Zukunft manipulationssicher hergestellt werden können. Ein Angreifer kann daher die Firmware

[3] Knoten, mit denen bereits ein gemeinsamer Schlüssel geteilt wird

entwendeter Knoten um Schadensfunktionen erweitern sowie geheimes Schlüsselmaterial extrahieren.

Sobald ein Außenstehender Zugriff auf die geheimen Schlüssel erlangt oder die Firmware eines Knotens für seine Zwecke manipuliert hat, kann er bösartige Knoten einschleusen, die von den anderen Knoten als vertrauenswürdig angesehen werden. In einem solchen Fall können die oben angeführten Schutzziele nicht mehr uneingeschränkt erreicht werden. Allerdings kann versucht werden den durch den bösartigen Knoten entstandenen Schaden lokal zu begrenzen.

Des Weiteren kann ein Angreifer auf Bitübertragungsschicht mit Hilfe eines Störsenders die gesamte Kommunikation des Sensornetzes zum Erliegen bringen. Ein gängiges Verfahren um diese Art von DoS[4]-Angriffen zu erschweren, besteht in der Verwendung von Spreizband- oder Frequenzsprungverfahren.

Auch auf der Sicherungsschicht können DoS-Angriffe durchgeführt werden: Selbst ein einzelner bösartiger Sensorknoten kann in seiner direkten Umgebung die Kommunikation anderer Knoten effektiv stören, indem er im richtigen Moment durch das Senden eines kurzen Pakets eine Kollision herbeiführt. Die andauernden Sendewiederholungen des angegriffenen Knoten führen zusätzlich zu einer raschen Erschöpfung der Energieressourcen.

Falls Datenpakete unverschlüsselt übertragen werden, können Angreifer durch Abhören der Luftschnittstelle unbefugt Zugriff auf die übertragenen Daten erhalten sowie manipulierte Datenpakete einschleusen. Selbst wenn die Datenübertragung verschlüsselt erfolgt, besteht die Möglichkeit empfangene Datenpakete zu einem späteren Zeitpunkt unverändert erneut auszusenden. Üblicherweise versucht man diese so genannten *Replay-Angriffe* mit Hilfe von Zeitstempeln zu erkennen.

2.4.1.2 Angriffe auf der Vermittlungsschicht

Ein naheliegender Angriff auf Routingprotokolle besteht in der Verbreitung falscher Routinginformationen. Durch die dadurch entstehenden Umwege, Schleifen oder Partitionierungen wird der Transport von Datennachrichten stark eingeschränkt oder gänzlich verhindert.

Selective-Forwarding-Attacks stellen ebenfalls einen naheliegende Form eines DoS-Angriffs dar. Bösartige Knoten verwerfen dort zur Weiterleitung bestimmte Datenpakete zufällig oder nach einem bestimmten Schema. Dazu muss sich der bösartige Knoten auf dem Pfad zwischen den beiden Knoten, deren Kommunikation er stören möchte, befinden. Die folgenden beiden Angriffe ermöglichen dem Knoten sich in den gewünschten Pfad zu integrieren.

Als *Sinkhole-Attacks* werden Angriffe bezeichnet, bei denen der bösartige Knoten versucht einen Großteil des ihn umgebenden Datenverkehrs zu sich selbst umzuleiten. Üblicherweise erreicht ein Knoten dies indem er behauptet optimale Routen zu verschiedenen Zielen zu besitzen. Insbesondere datenorientierte Sensornetze sind für diese Angriffsform anfällig: Aufgrund deren dortigen Kommunikationsstruktur reicht es aus, einen optimalen Weg zur Basisstation zu propagieren, um möglichst viele der umliegenden Knoten zu beeinträchtigen.

[4]Denial of Service

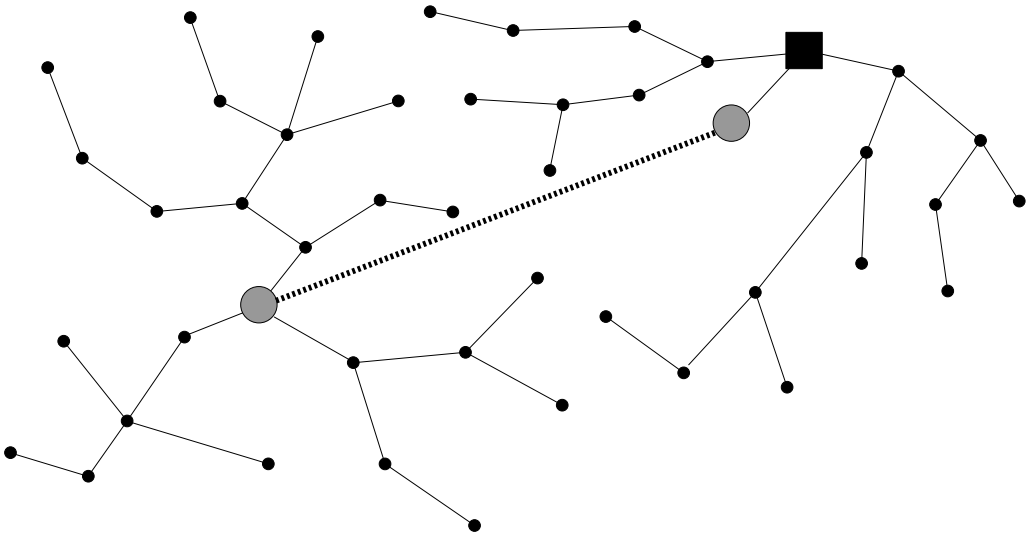

Abbildung 2.7: Die beiden grauen Angreifer bilden mit Hilfe eines Tunnels ein so genanntes Wurmloch

Eine effektive und schwer zu erkennenden Methode zur Erzeugung von Sinkholes ist die *Wormhole-Attack*. Ein solcher Angriff ist in Abbildung 2.7 dargestellt. Der Angreifer verwendet dazu eine direkte Verbindung mit geringer Latenz zu einem weiteren bösartigen Knoten um Nachrichten von einem Teil des Netzes in eine weit entfernte Gegend zu tunneln. Dadurch erreicht dieser, dass ein Großteil des Datenverkehrs einer Region durch den Tunnel geleitet und dort beispielsweise verworfen werden kann.

Sofern ein Knoten mit hoher Sendeleistung zur Verfügung steht, kann auch eine Variante dieses Verfahrens eingesetzt werden, um die Sicht der Knoten auf die Netztopologie zu verschleiern: Der Knoten wiederholt dazu mitgehörte Broadcast-Signalisierungsnachrichten mit ausreichender Sendeleistung, so dass die Nachricht von allen Knoten im gesamten Netzwerk empfangen wird.

Das Auftreten eines einzelnen Knotens unter mehreren Identitäten wird als *Sybil-Attack* [Douc02] bezeichnet. Durch diesen Angriff kann die Effektivität von Schutzmechanismen wie die Verwendung von Replikaten oder *Multipath-Routing* reduziert werden. Diese Mechanismen basieren auf dem Prinzip Nachrichten redundant über verschiedene Pfade zum Ziel zu schicken und dadurch Angreifer auf einzelnen Pfaden tolerieren zu können. In Abschnitt 4.1 werden diese Verfahren näher beschrieben. Durch eine Sybil-Attack kann ein Angreifer nun vortäuschen, dass die vermeintlich verschiedenen Pfade von disjunkten Knoten gebildet werden, obwohl diese in der Realität alle über den selben einen Angreifer führen. Auf diese Weise werden die genannten Schutzmechanismen wirkungslos gemacht.

Viele der hier vorgestellten Angriffe können gegenüber Outsidern durch einfach Verschlüsselung und Authentisierung auf der Sicherungsschicht mittels eines gemeinsamen Schlüssels vereitelt werden. Zur Verhinderung von Wormhole-Angriffen ist dieser Mechanismus jedoch nicht geeignet, da eine Teilnahme des angreifenden Knotens am Signalisierungsverkehr bei Sybil-Angriffen nicht nötig ist - der Angreifer wiederholt nur Signalisierungspakete legitimierter Knoten.

Gegenüber Insidern bzw. kompromittierten Knoten ist diese Art der Verschlüsselung jedoch nutzlos. Sofern ein Knoten für jeden seiner Nachbarn einen eigenen Schlüssel verwendet, kann der Schaden zumindest lokal begrenzt werden. In diesem Fall stellt sich jedoch wieder die Frage nach einem sicheren und ressourcensparenden Schlüsselverteilungsverfahren.

2.4.1.3 Angriffe auf DHTs

Aufgrund der Tatsache, dass DHTs als Overlaynetze realisiert werden, hängt deren Verfügbarkeit direkt von der Verfügbarkeit des darunterliegenden Netzes ab. Falls das Routing im Underlay infolge eines Angriffs gestört ist, kann auch zwischen Overlaynachbarn keine Kommunikation mehr stattfinden. DoS-Angriffe auf DHTs können somit also nur verhindert werden, wenn die Protokolle der darunterliegenden Schichten ebenfalls robust gegenüber Angriffen auf die Verfügbarkeit sind.

Eine kurze Zusammenstellung möglicher Angriffe und Abwehrmaßnahmen auf DHTs in P2P-Systemen gibt [SiMo02]. Die dort vorgestellten Angriffe können großteils in zwei Gruppen aufgeteilt werden: Angriffe auf das Routing sowie Angriffe auf das Speichern und Abfragen von Daten.

Viele der im vorherigen Abschnitt vorgestellten Angriffe auf der Vermittlungsschicht können in ähnlicher Weise auch auf die Wegewahl in DHTs angewandt werden. Bösartige Knoten können Nachrichten, für dessen Weiterleitung sie zuständig sind, verwerfen oder an einen falschen Knoten weiterleiten. Da der bösartige Knoten gegenüber seinen Nachbarn als funktionsfähig und vertrauenswürdig auftritt, werden auch die Sendewiederholungen der verworfenen Nachrichten wieder über diesen Knoten verschickt und somit die Kommunikation zwischen den Endgeräten wirksam unterbunden. Zusätzlich besteht für den vermittelnden Knoten die Möglichkeit, den Inhalt einer weiterzuleitenden Nachricht zu modifizieren.

Da die Routingtabellen der Knoten aufgrund der von Nachbarn verbreiteten Informationen aufgestellt werden, kann ein bösartiger Knoten durch die Verbreitung falscher Routinginformationen die Routingtabellen seiner Nachbarn unbrauchbar machen. Des Weiteren kann mit Hilfe mehrerer Angreifer ein zweites, parallel existierendes Netz aufgebaut werden. Ein neu hinzukommender Knoten könnte sich nun versehentlich an diesem Netz anmelden ohne zu bemerken, dass es sich um das Netz der Angreifer handelt. Dieser Angriff kann zur Überwachung des Verhalten des hinzukommenden Knotens oder zur Beeinträchtigung der Verfügbarkeit des Netzes verwendet werden.

In [CDGR+02] wird detailliert auf sicheres Routing in P2P-Overlaynetzen eingegangen. Die Autoren stellen dort drei Voraussetzungen auf, die für sicheres Routing in P2P-Overlays benötigt werden:

- *Sicheres Zuweisen der Knoten-ID*: Falls in einem Netz die Teilnehmer ihre Knoten-ID frei wählen können, kann dies leicht von einem Angreifer ausgenutzt werden um gezielt einzelne Knoten im Netz anzugreifen. Dazu muss der Angreifer nur die Identität aller Nachbarknoten des betroffenen Knoten auswählen und erlangt dadurch die vollständige Kontrolle über dessen Routingtabelle. Alternativ kann der Angreifer durch geschickte Wahl seiner ID auch die Zuständigkeit für einen bestimmten Datensatz erlangen und dadurch

anfragenden Knoten den Zugriff auf diese Daten verweigern. Auch in Netzen ohne freie ID-Wahl besteht diese Gefahr durch den zuvor bereits erwähnten Sybil-Angriff, falls es einem Knoten gelingt mehrere Identitäten anzunehmen.

- *Sichere Wartung der Routingtabelle*: Overlaynetze, die ihre Topologie an die darunterliegende Netzwerktopologie anpassen, um das Routing effizienter zu gestalten, sind anfällig für den in Abschnitt 2.4.1.2 vorgestellten Wurmlochangriff. Durch eine (scheinbare) Änderung in der Underlaytopologie, kann dadurch Einfluss auf die Routingtabelle des Overlayknotens genommen werden. Durch die Verbreitung falscher Routinginformationen kann alternativ auch versucht werden die Routingtabelle direkt zu manipulieren. In DHTs, in denen die Wahl der Nachbarknoten jedoch stark von deren Knoten-ID abhängt (wie z.B. in CAN), sind solche Angriffe jedoch schwer durchzuführen.

- *Sicheres Weiterleiten von Nachrichten*: Falls eine Nachricht auf ihrem Weg zum Ziel einen bösartigen Knoten passiert, kann dieser die Nachricht modifizieren oder verwerfen. Daher kann eine Nachricht nur sicher zum Ziel geleitet werden, wenn alle h Knoten auf dem Weg zum Ziel gutartig sind. Im Fall von f bösartigen Knoten beträgt die Wahrscheinlichkeit hierfür nur $(1-f)^{h-1}$. Die Erfolgswahrscheinlichkeit hängt also direkt von der durchschnittlichen Weglänge h ab. Sofern durch das Overlay mehrere unabhängige Wege zum Ziel zur Verfügung gestellt werden, kann durch das redundante Verschicken der Nachricht über einige dieser Wege die Wahrscheinlichkeit für eine erfolgreiche Zustellung der Nachricht erhöht werden.

Die zweite Klasse von Angriffen richtet sich gegen die Abfrage und Speicherung der eigentlichen Daten. Ein Knoten kann die Daten, die in seinem ID-Bereich liegen, beliebig modifizieren oder einem anfragenden Knoten die Daten vorenthalten. Diese Angriffe können durch die Verwendung von Redundanz abgewehrt werden. Dazu werden mehrere Replikate der Daten bei verschiedenen Knoten im Netz gespeichert. Ein anfragender Knoten kann dann durch Vergleich der Replikate bösartige Knoten erkennen und Daten von diesen Knoten verwerfen.

Schließlich kann ein Angreifer noch versuchen die Verfügbarkeit des Overlaynetzes durch die Überlastung einzelner Knoten einzuschränken, indem er beispielsweise das Netz mit sinnlosen Anfragen fluten oder einzelne Knoten veranlasst rechenintensive Operationen durchzuführen.

2.4.2 Kryptographie

Dieser Abschnitt gibt einen kurzen Überblick über kryptographisch Verfahren, die zur Erfüllung der genannten Schutzziele eingesetzt werden können. Einen umfassende Überblick der Kryptographie und kryptographischer Protokolle gibt [MeOV01]. Das Schutzziel Vertraulichkeit wird in der Regel durch Einsatz von *Verschlüsselungsverfahren* erreicht, welches einen Klartext unter Verwendung ein oder mehrerer *Schlüssel* in ein Chiffrat überführt. Bei den Verschlüsselungsverfahren kann zwischen symmetrischen und asymmetrischen Verfahren unterschieden werden.

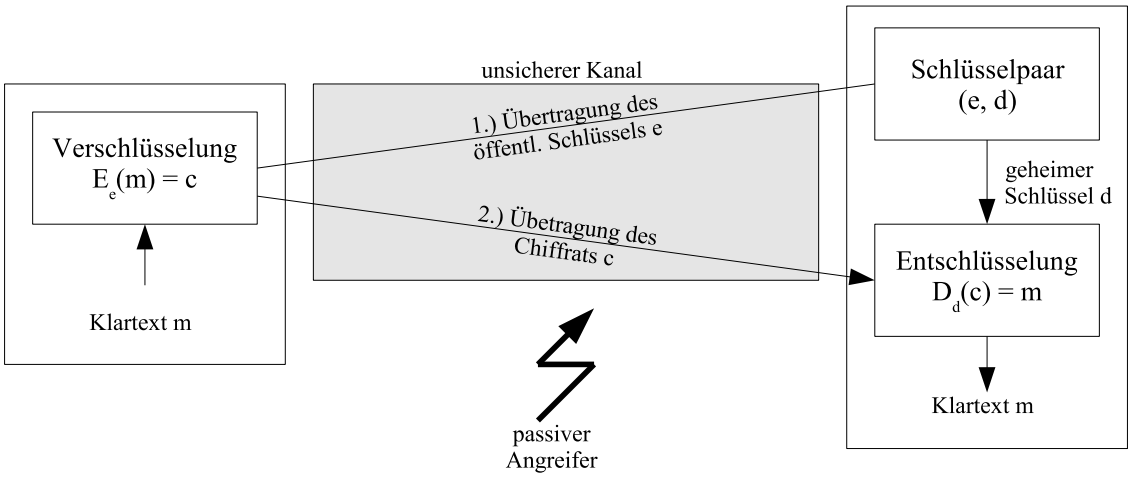

Abbildung 2.8: Verschlüsselte Übertragung eines Klartexts m über einen unsicheren Kanal mit Hilfe eines asymmetrischen Verschlüsselungsverfahrens

2.4.2.1 Symmetrische Verfahren

Symmetrische Verschlüsselungsverfahren verwenden einen *gemeinsamen Schlüssel*, der sowohl zur Ver- als auch zur Entschlüsselung verwendet wird. Dieser gemeinsame Schlüssel muss im Voraus über einen sicheren Kanal zwischen den Kommunikationspartnern ausgetauscht werden. Typische Vertreter dieser Klasse sind die Algorithmen DES, IDEA und AES. Ein symmetrischer Verschlüsselungsalgorithmus gilt üblicherweise als sicher, sofern keine schnellere Methode als ein *brute-force Angriff*, das heißt das vollständige Durchprobieren des gesamten Schlüsselsraums, bekannt ist, um den Klartext aus einem Chiffrat ohne Kenntnis des gemeinsamen Schlüssels zu erhalten. Sofern dies gegeben ist, werden heutzutage Schlüssellängen von 128-Bit als ausreichend angesehen.

2.4.2.2 Asymmetrische Verfahren

Im Unterschied zu den symmetrischen Verfahren kommen bei asymmetrischen Verfahren wie RSA [RiSA83] und ElGamal [Gama85] getrennte Schlüssel für die Ver- und Entschlüsselung zum Einsatz. Der Schlüssel, der zur Verschlüsselung verwendet wird, heißt *öffentlicher Schlüssel*[5] und kann allen Kommunikationspartnern öffentlich mitgeteilt werden. Der zugehörige *geheime Schlüssel* dient zum Entschlüsseln der Nachrichten. Die Übertragung des öffentlichen Schlüssels kann, im Gegensatz zu den symmetrischen Verfahren, über einen unsicheren Kanal erfolgen und ermöglicht somit die sichere Kommunikation zwischen zwei Kommunkationspartnern ohne vorherigen Austausch von geheimem Schlüsselmaterial.

Der Ablauf einer solchen verschlüsselten Übertragung ohne vorherigen Austausch von Schlüsselmaterial ist in Abbildung 2.8 dargestellt. In einem ersten Schritt wird der öffentliche Schlüssel e über den unsicheren Kanal übertragen. Der Schlüssel kann dann zum Verschlüsseln des Klartexts m verwendet werden. Im zweiten Schritt wird dann das so erhaltene Chiffrat c übermittelt. Der Empfänger daraufhin mit Hilfe seines geheimen Schlüssels e das Chiffrat dekodieren.

[5] asymmetrische Kryptographie wird daher auch *Public-Key-Kryptographie* genannt

Asymmetrische Verfahren basieren in der Regel auf mathematischen Problemen wie dem Faktorisieren großer Zahlen oder dem Berechnen des diskreten Logarithmus, von denen angenommen wird, dass diese aufwändig zu lösen sind. Da Algorithmen zur Lösung dieser „schweren“ Probleme dennoch effektiver sind als ein direkter brute-force Angriff auf den Schlüssel, müssen bei Public-Key-Verfahren längere Schlüssel eingesetzt als dies bei symmetrischen Verfahren notwendig ist. Momentan werden Schlüssellängen in der Größenordnung von 2048 Bit als sicher angesehen [MeOV01].

Ein großer Nachteil von public-key Verfahren stellt deren hoher Ressourcenverbrauch dar: Beispielsweise ist RSA im Vergleich zu DES bei einer Softwareimplementierung um ungefähr den Faktor 100, bei einer Hardwareimplementierung um den Faktor 1000 – 10000 langsamer. Nach Ansicht vieler Autoren scheidet daher die Verwendung von public-key Kryptographie auf leistungsschwachen Sensorknoten momentan aus [KaSW04, PSWC+01]. Durch den Einsatz effizienterer Verschlüsselungsverfahren, die auf elliptischen Kurven basieren, und der Entwicklung leistungsfähigerer Sensorknoten könnte sich dieser Umstand jedoch in naher Zukunft ändern [MaWS04].

2.4.2.3 Authentizität und Integrität

Das direkte Verschlüsseln von Nachrichten bietet nur Schutz gegen das unerwünschtes Mitlesen der Nachrichten durch Dritte. Mit einfachen Maßnahmen lassen sich durch Verwendung eines symmetrischen Kryptoverfahrens jedoch auch die Schutzziele Authentizität und Integrität erreichen:

Sofern zwischen allen Kommunikationspartnern separate symmetrische Schlüssel ausgetauscht wurden, kann mit Hilfe eines *Challenge-Response-Verfahrens* die Identität des Kommunikationspartners überprüft werden. Dazu wird als „Herausforderung“ eine Zufallszahl im Klartext an diesen gesendet. Der Partner verschlüsselt die erhaltene Zufallszahl mit dem gemeinsamen symmetrischen Schlüssel und antwortet mit diesem Chiffrat. Da der Schlüssel nur diesen beiden Knoten bekannt ist, hat sich der Knoten durch seine Antwort identifiziert. Um Replay-Attacken zu erschweren, wird für weitere Identitätsüberprüfung jeweils eine neue Zufallszahl gewählt.

Zur Integritätsprüfung einer Nachricht kommen in der Regel *Hashfunktionen* zu Einsatz. Diese bilden eine Nachricht beliebiger Länge auf einen kurzen Hashwert mit fester Länge ab. Damit eine Hashfunktion zu kryptographischen Zwecken geeignet ist, muss diese verschiedene Eigenschaften erfüllen. Zu einem gegebenen Hashwert darf es mit vertretbaren Rechenaufwand praktisch nicht möglich sein eine Nachricht zu finden, die zu diesem Hashwert passt. Eine solche Hashfunktion stellt also eine Einwegfunktion dar. Des Weiteren darf zu einer gegebenen Nachricht keine zweite Nachricht gefunden werden, die den gleichen Hashwert wie die vorgegebene Nachricht besitzt. Schließlich muss das Prinzip der Kollisionsfreiheit gelten, das heißt es muss praktisch unmöglich sein zwei beliebige Nachrichten zu finden, die den gleichen Hashwert ergeben. Der Hashwert stellt dabei einen digitalen Fingerabdruck der ursprünglichen Nachricht dar und wird auch als *modification detection code* (MDC) bezeichnet.

Um eine Integritätsprüfung der Nachrichten zu ermöglichen, wird vom Sender der MDC über die Nachricht berechnet und zusammen mit der Nachricht verschlüsselt und übermittelt. Der Empfänger überprüft nach dem Entschlüsseln der Nachricht, ob der MDC mit dem Hashwert der erhaltenen Nachricht übereinstimmt. Falls Teile der Nachricht von einem Angreifer verändert wurden, schlägt diese Überprüfung

fehl, da der Angreifer ohne Kenntnis des geheimen Schlüssels nicht in der Lage ist, einen neuen MDC zu berechnen. Sofern auf dem Kommunikationssystem keine effiziente Hashfunktion zur Verfügung steht, kann ein vorhandene Blockchiffre verwendet werden, um eine Hashfunktionen zu realisieren.

3. Analyse von SCANv2

In diesem Kapitel wird zunächst das zu untersuchende Dienstverzeichnis „Secure Content Addressable Network Version 2“ (SCANv2)[HoBZ04] vorgestellt. Daraufhin folgt eine Sicherheitsanalyse, die zeigt, ob das Protokoll gegenüber den in Abschnitt 2.4.1 aufgeführten Angriffe robust ist und ob die von den Autoren geforderten Schutzziele erreicht werden.

3.1 Das Dienstverzeichnis SCANv2

SCANv2 ist ein Dienstverzeichnis, das für die Verwendung in dienstorientierten Sensornetzen entworfen wurde (siehe Abschnitt 2.1) und auf dem in Abschnitt 2.3 beschriebenen CAN-Overlay basiert. Da die Wahl der Kommunikationspartner in dienstorientierten Sensornetzen dynamisch aufgrund der von diesen angebotenen Diensten erfolgt, muss ein effizientes Verfahren zur Suche von Diensten (*Lookup-Operation*) und für das Hinterlegen von Diensten (*Insert-Operation*) zur Verfügung stehen. Um der dezentralen Struktur von Sensornetzen gerecht zu werden, verzichtet SCANv2 weitgehend auf die Verwendung zentraler Komponenten. Das vorgeschlagene Protokoll verwendet das CAN-Overlay um Dienstbeschreibungen dezentral zu speichern. Auf diese Weise wird die Verteilung der Last auf mehrere Knoten ermöglicht. Außerdem ist das Dienstverzeichnis robuster gegenüber Knotenausfällen.

Die Sicherheitsarchitektur von SCANv2 ist wesentlicher Bestandteil des Protokolls. Diese soll das sichere Einfügen von Diensten in das Verzeichnis ermöglichen und anfragenden Knoten korrekte Dienstbeschreibungen auf gestellte Suchanfragen liefern. Da das CAN-Overlay anfällig für die in Abschnitt 2.4.1.3 aufgeführten Angriffe ist, wurde das Protokoll in SCANv2 um Sicherheitsmechanismen erweitert. Dadurch soll das Verzeichnis vor Angriffen durch Außenstehende geschützt und der durch Insiderangriffe verursachte Schaden lokal begrenzt werden.

In Hinblick auf die beschränkten Ressourcen der Sensorknoten wird versucht, diese Ziele mit möglichst geringem Rechen- und Kommunikationsaufwand zu erreichen. Insbesondere wird auf den Sensorknoten auf die Verwendung von Public-Key-Kryptographie verzichtet.

3.1.1 Sicherheitsarchitektur von SCANv2

Damit das Overlay vor Angriffen durch Außenstehende geschützt werden kann, erfolgt die Kommunikation zwischen Overlay-Nachbarn nur nach einer erfolgreichen Authentifizierung. Diese Authentifizierung wird wie im Grundlagenkapitel in Abschnitt 2.4.2.3 erläutert durch die Verwendung symmetrischer Schlüssel erreicht. Dazu besitzt jeder Overlayknoten gemeinsame Schlüssel mit allen im Overlay benachbarten Knoten. Der Zugang zum Overlay wird dabei nur denjenigen Sensorknoten gewährt, die vom Administrator des Sensornetzes dazu explizit autorisiert wurden. Die Sicherheitsarchitektur sieht hierzu die Verwendung eines so genannten *Master-Device* (MD) vor, das als Vertrauensanker fungiert und selbst nicht Bestandteil des Sensornetzes ist.

Bei dem Master-Device handelt es sich um ein tragbares Gerät, das leistungsfähiger ist als ein typischer Sensorknoten und insbesondere den Einsatz asymmetrischer Verschlüsselungsverfahren ermöglicht. Durch den erstmaligen physischen Kontakt mit dem Master-Device wird der Sensorknoten personalisiert und ein durch das Master-Device generierter symmetrischer Schlüssel auf den Knoten übertragen. Diese Vorgehensweise entspricht weitgehend dem in [StAn00] vorgestellten Sicherheitsmodell. Durch diesen gemeinsamen Schlüssel kann sich das Master-Device zu einem späteren Zeitpunkt wieder gegenüber dem Sensorknoten authentifizieren ohne erneut den physischen Kanal zu verwenden. Das Master-Device ist das einzige Gerät im Netz, dem die Knoten absolutes Vertrauen entgegenbringen und das berechtigt ist, eine Änderungen der Overlaytopologie zu erwirken. Auf diese Weise werden die in Abschnitt 2.4.1.3 genannten Angriffe auf die Overlaytopologie verhindert. Wird ein neuer Knoten zum Netz hinzugefügt, entscheidet das MD aus seiner Sichtweise welche neuen Nachbarschaftsbeziehungen nach dem Beitritt des Knotens benötigt werden und legitimiert diese durch Etablierung eines symmetrischen Schlüssels auf den beteiligten Nachbarknoten.

Damit das Master-Device nicht permanent Informationen über die aktuelle Topologie des Overlays speichern muss, verwendet es *Zertifikate* um diese Daten auf den Sensorknoten zwischenzuspeichern. Jeder neue Knoten erhält vom Master-Device nach einem erfolgreichen Netzbeitritt ein solches Zertifikat, das die Zonenkoordinaten des Knoten enthält und vom Master-Device digital signiert wurde. Tritt der Knoten zu einem späteren Zeitpunkt erneut in Kontakt mit dem Master-Device, kann er die Zuständigkeit für seine Zone durch Vorzeigen des Zertifikats beweisen. Da somit auf dem Master-Device bis auf einen Schlüssel keine weiteren Zustände gespeichert werden, kann es im Fall einer Beschädigung oder des Verlusts problemlos ersetzt werden.

3.1.2 Protokoll zum sicheren Hinzufügen neuer Knoten

Das Hinzufügen eines neuen Knotens in das SCANv2-Overlay verläuft in 10 Schritten. Der folgende Protokollablauf entspricht der Darstellung in [HoBZ04]. Zur Darstellung der versendeten Nachrichten wird die folgende Notation verwendet:

$$Schicht|A \rightarrow B : Nachricht$$

Schicht steht für die verwendete Netzwerkschicht auf der die Nachricht übertragen wird. Mögliche Netzwerkschichten sind in diesem Fall der physische Kontakt

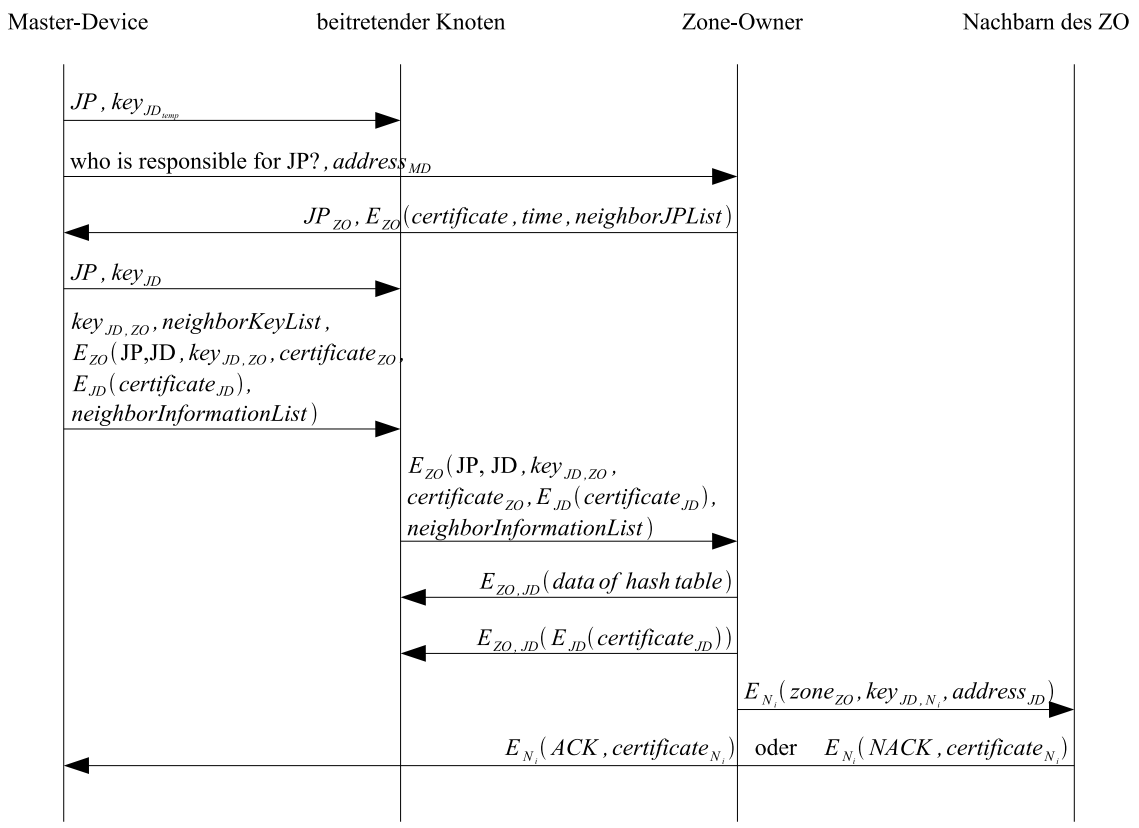

Abbildung 3.1: Protokollablauf für das Hinzufügen neuer Knoten

(*PHY*), die Datenübertragung auf Underlayebene (*MC*) sowie die Verwendung des CAN-Overlays (*CAN*). Der Sender A sowie der Empfänger B einer Nachricht können das Master-Device (*MD*), der beitretende Knoten (*JD*), der Zone-Owner (*ZO*) oder die Nachbarn des Zone-Owners (N_i) sein. In Abbildung 3.1 ist der Ablauf des Knotenbeitritts graphisch dargestellt.

- 1.) PHY|MD→JD : JP, $\text{key}_{\text{JD}_{\text{temp}}}$

  Das Master-Device schickt einen zufällig gewählten Join-Point JP über den sicheren, physischen Kanal an den beitretenden Knoten. Die zufällige Wahl ist wichtig, um eine statistisch gleichmäßige Verteilung der Zonengrößen zu erhalten. Durch Verschlüsselung des Join-Points mit dem geheimen Schlüssel des Master-Device wird der vorläufige symmetrische Schlüssel $\text{key}_{\text{JD}_{\text{temp}}} = \text{E}_{\text{SK}}(\text{JP})$ erzeugt und an den beitretenden Knoten übermittelt. Jede weitere Kommunikation zwischen dem Master-Device und dem beitretenden Knoten wird mit diesem Schlüssel verschlüsselt. Ab diesem Zeitpunkt ist die Benutzung des sicheren physischen Kanals nicht mehr notwendig und kann durch eine drahtlose Verbindung ersetzt werden. Da sich der Join-Point und mit diesem auch der Schlüssel key_{JD} zu einem späteren Zeitpunkt noch ändern[1] können, wird der Schlüssel als temporär markiert und ist nur bis zur Zuweisung des endgültigen Schlüssels im vierten Schritt gültig.

[1]Der Join-Point eines Knotens muss immer innerhalb der eigenen Zone liegen. Falls dies nach der Teilung einer Zone nicht mehr gegeben ist, muss dem beitretenden Knoten ein neuer Join-Point zugewiesen werden.

- 2.) CAN|MD→ZO : "*who is responsible for JP*?", $address_{MD}$

  Im nächsten Schritt versendet das Master-Device eine Nachricht mit der Zieladresse JP über das CAN-Overlay. Der für den Punkt JP zuständige Zone-Owner wird so über den Beitrittswunsch des neuen Knotens informiert. Da die Nachricht die Netzwerkadresse des Master-Device enthält, kann die weitere Kommunikation zwischen dem Master-Device und dem Zone-Owner im Underlay stattfinden und damit der benötigte Kommunikationsaufwand reduziert werden. Falls die Hardware des Master-Device keine direkte Kommunikation mit dem Sensornetz ermöglicht, kann das Master-Device die Kommunikationseinrichtungen des beitretenden Knotens verwenden, um mit den Netz zu kommunizieren.

- 3.) MC|ZO→MD : $\mathrm{JP_{ZO}}, \mathrm{E_{ZO}}(\mathit{certificate}, \mathit{time}, \mathit{neighborJPList})$

  Der Zone-Owner antwortet dem Master-Device mit seinem Join-Point $\mathrm{JP_{ZO}}$ und dem Zertifikat seiner Zone. Die Nachricht ist mit dem gemeinsamen Schlüssel $\mathrm{key_{ZO}}$ von Master-Device und Zone-Owner verschlüsselt. Wie im ersten Schritt kann das Master-Device diesen Schlüssel durch Verschlüsseln von $\mathrm{JP_{ZO}}$ mit seinem geheimen Schlüssel herleiten. Um Replay-Attacken zu verhindern enthält die Nachricht den Zeitstempel *time*. Mit Hilfe des Zertifikats *certificate*, das die Zonenkoordinaten des Zone-Owners enthält und vom Master-Device signiert ist, weist sich der Zone-Owner als rechtmäßiger Verwalter der Zone aus. Die *neighborJPList* besteht aus einer Liste aller Netzwerkadressen und Join-Points der Nachbarn des Zone-Owners. Dadurch wird dem Master-Device ermöglicht, die für die Kommunikation mit den Nachbarn des Zone-Owners benötigten Schlüssel herzuleiten. Diese werden im 9. Schritt verwendet um Angriffe während des Knotenbeitritts erkennen zu können.

- 4.) PHY|MD→JD : $\mathrm{JP}, \mathrm{key_{JD}}$

  Das Master-Device teilt die Zone des Zone-Owners und wählt soweit erforderlich einen neuen Join-Point für den beitretenden Knoten. Aus diesem wird erneut ein symmetrischer Schlüssel key_{JD} hergeleitet und dem beitretenden Knoten mitgeteilt. Dieser Schlüssel ersetzt den bisherigen temporären Schlüssel $key_{JD_{temp}}$ und wird ab sofort für die weitere Kommunikation zwischen Master-Device und dem beitretenden Knoten verwendet.

- 5.) PHY|MD→JD : $\mathrm{key_{JD,ZO}}$, *neighborKeyList*, $\mathrm{E_{ZO}}$("JP, JD", $\mathrm{key_{JD,ZO}}$, $certificate_{ZO}, \mathrm{E_{JD}}(certificate_{JD})$, *neighborInformationList*)

  Mit dieser Nachricht erhält der beitretende Knoten den symmetrischen Schlüssel $key_{JD,ZO}$ zur sicheren Kommunikation mit dem Zone-Owner. Die *neighborKeyList* ist eine Liste der neuen Nachbarn des beitretenden Knotens. Für jeden Nachbarn enthält sie neben dessen Netzwerkadresse und Join-Point einen symmetrischen Schlüssel. Somit können alle Nachrichten, die zwischen zwei benachbarten Overlayknoten ausgetauscht werden, verschlüsselt übertragen werden.

  Der zweite Teil der Nachricht ermächtigt den beitretenden Knoten gegenüber dem Zone-Owner die Teilung der Zone anzustoßen. Der Zone-Owner kann aufgrund der Verschlüsselung mit key_{ZO} sicherstellen, dass die Zonenteilung vom Master-Device legitimiert wurde. Die *neighborInformationList* wird verwendet, um die Nachbarn des Zone-Owners über die Teilung seiner Zone zu informieren.

- 6.) MC|JD→ZO : $\mathrm{E_{ZO}}(\text{“JP, JP“}, \mathrm{key_{JD,ZO}}, \mathit{certificate_{ZO}}, \mathrm{E_{JD}}(\mathit{certificate_{JD}}), \mathit{neighborInformationList})$

  Mit dem Erhalt dieser Nachricht beginnt der Zone-Owner mit der Teilung seiner Zone. Die *neighborInformationList* enthält für jeden Nachbar des Zone-Owners ein Tupel der Form (*Zonenkoordinaten des ZO, symmetrische Schlüssel mit JD, symmetrischer Gruppenschlüssel mit ZO, Join-Point des JD*). Die Tupel sind jeweils mit dem gemeinsamen Schlüssel von Master-Device und Nachbar verschlüsselt. Da das bisherige Zonenzertifikat des Zone-Owners aufgrund der geänderten Zonenkoordinaten ungültig wird, erhält dieser mit $\mathit{certificate_{ZO}}$ ein aktualisiertes Zertifikat.

- 7.) MC|ZO→JD : $\mathrm{E_{ZO,JD}}(\mathit{data\ of\ hash\ table})$

  Der Zone-Owner überträgt alle gespeicherten Dienstbeschreibungen, die in die Zone des beitretenden Knotens fallen. Die Manipulation der übertragenen Daten soll durch Verschlüsselung mit dem gemeinsamen Schlüssel $\mathrm{key_{ZO,JD}}$ verhindert werden. Der Empfang der Daten wird daraufhin vom beitretenden Knoten bestätigt.

- 8.) MC|ZO→JD : $\mathrm{E_{ZO,JD}}(\mathrm{E_{JD}}(\mathit{certificate_{JD}}))$

  Sobald der Datentransfer abgeschlossen ist, händigt der Zone-Owner dem beitretenden Knoten dessen Zonenzertifikat aus. Durch die Übertragung des Zertifikats am Ende des Transfers wird dem Zone-Owner die Möglichkeit gegeben die Glaubwürdigkeit des beitretenden Knotens zu prüfen, indem der Zone-Owner einige der transferierten Dienstbeschreibungen anfordert und mit lokal gespeicherten Beschreibungen vergleicht. Obwohl der beitretende Knoten nicht daran gehindert wird die Daten zu einem späteren Zeitpunkt zu verwerfen, soll durch diese Maßnahme das Betrügen erschwert werden. Nach Abschluss des Transfers vernichtet der Zone-Owner sein altes Zonenzertifikat und ersetzt es durch das im 6. Schritt erhaltene neue Zertifikat. Ab diesem Zeitpunkt ist der Zone-Owner für die an den beitretenden Knoten abgetretene Zonenhälfte nicht mehr verantwortlich.

- 9.) MC|ZO→$\mathrm{N_i}$: $\mathrm{E_{N_i}}(\mathit{zone_{ZO}}, \mathrm{key_{JD,N_i}}, \mathit{address_{JD}})$

  Der Zone-Owner informiert alle betroffenen Nachbarn über die erfolgte Zonenteilung. Dazu versendet er die vom Master-Device vorbereiteten (und von diesem verschlüsselten) Nachrichten, die er im 6. Schritt erhalten hat. Diese enthalten die Zonenkoordinaten des Zone-Owners, einen symmetrischen Schlüssel für die Kommunikation zwischen dem beitretenden Knoten und dem Zone-Owner sowie die Netzwerkadresse des beitretenden Knotens. Die Nachbarn überprüfen daraufhin, ob die angegebenen Zonenkoordinaten des Zone-Owners aus ihrer lokalen Sicht korrekt sind.

- 10.) MC|$\mathrm{N_i}$→MD : $\mathrm{JP}, \mathrm{E_{N_i}}(\text{ACK or NACK}, \mathit{certificate_{N_i}})$

  Erscheinen die angegebenen Zonenkoordinaten des Zone-Owners aus Sicht eines Nachbarn korrekt, bestätigt er diese mit einem *ACK*. Andernfalls antwortet er mit einer *NACK*-Nachricht. In beiden Fällen sendet der Nachbar sein Zonenzertifikat mit, welches vom Master-Device auf Gültigkeit überprüft wird. Sollte einer der Nachbarn mit einem *NACK* geantwortet haben, hat entweder

der Zone-Owner oder einer der Nachbarn ein altes Zonenzertifikat verwendet. Das Master-Device kann nun mit Hilfe eines Mehrheitsentscheids feststellen, welcher der Knoten betrogen hat. Um das Problem zu beheben, bestimmt das Master-Device zufällig einen neuen Zone-Owner für die betroffene Zone und teilt die Entscheidung allen betroffenen Nachbarn mit.

3.1.3 Behandlung von Knotenausfällen

Ausfälle von Nachbarknoten sollen in SCANv2 durch das Ausbleiben periodisch gesendeter Update-Nachrichten erkannt werden. Sobald ein Nachbar den Ausfall eines Knotens erkennt, informiert er die anderen Nachbarn des ausgefallenen Knotens unter Verwendung des symmetrischen Gruppenschlüssels, den alle Nachbarn der ausgefallenen Zone gemeinsam haben. Diese wählen daraufhin einen neuen Zone-Owner für die ausgefallene Zone. Der neu gewählte Zone-Owner erhält von jedem Nachbarknoten eine Bescheinigung über dessen Wahl. Die Wahlbescheinigungen enthalten die Zonenzertifikate der Knoten und werden mit dem symmetrischen Schlüssel, den die Knoten mit dem Master-Device gemeinsam haben, verschlüsselt.

Ist bei einem der nächsten Knotenbeitritte die Zone des neuen Zone-Owners involviert, werden dessen Wahlbescheinigungen vom Master-Device geprüft. Sobald die Nachbarn des Zone-Owners die angegebenen Zonenkoordinaten bestätigt haben, stellt das Master-Device für die betroffene Zone ein neues Zonenzertifikat aus.

3.2 Sicherheitsanalyse von SCANv2

Die folgende Analyse des Dienstverzeichnisses SCANv2 soll zeigen, ob das Protokoll die von den Autoren geforderten Ziele erreicht: Das Dienstverzeichnis soll das sichere Einfügen von Diensten ermöglichen und anfragenden Knoten jederzeit korrekte Dienstbeschreibungen auf gestellte Suchanfragen liefern. Sofern Angriffe auf das Verzeichnis nur durch Außenstehende durchgeführt werden, sollen die genannten Ziele jederzeit erreicht werden. Im Fall von Insiderangriffen soll der verursachte Schaden lokal begrenzt werden.

Für die Bewertung des Dienstverzeichnisses wurden die folgenden, von den Autoren in [HoBZ04] getroffenen Prämissen berücksichtigt. Aus den bereits in Abschnitt 2.4 genannten Gründen wird davon ausgegangen, dass Sensorknoten nicht manipulationssicher sind und von einem Angreifer übernommen werden können. In diesem Fall hat der Angreifer auch die Möglichkeit den übernommenen Knoten zu klonen und durch eine Reihe eigener Knoten zu ersetzen. Um den größtmöglichen Schaden zu erzielen, können alle bösartigen Knoten im Netz miteinander kommunizieren und kooperieren. Des Weiteren wird davon ausgegangen, dass alle neuen Sensorknoten zum Zeitpunkt ihres Netzbeitritts gutartig sind und der Angreifer keine Möglichkeit hatte diese zu modifizieren[2]. Der Angreifer hat zudem kein Zugriff auf das Master-Device, da dieses nur beim Hinzufügen neuer Knoten zum Einsatz kommt und die restliche Zeit an einem sicheren Ort aufbewahrt wird. Das Protokoll stellt an die vom ihm verwendeten Kommunikationsprotokolle auf den darunterliegenden Schichten keine Sicherheitsanforderungen. Damit kann ein Angreifer die in Abschnitt 2.4.1.1 und 2.4.1.2 dargestellten Angriffe einsetzen, um beispielsweise seine Netzwerkadresse zu fälschen (so genanntes *Spoofing*).

[2]Beispielsweise könnten die Knoten von einem vertrauenswürdigen Hersteller produziert und versiegelt worden sein

3.2.1 Angriffe durch Outsider

In diesem Abschnitt werden Angriffe betrachtet, die von einem außenstehenden Angreifer durchgeführt werden können. Da die Kommunikation zwischen Overlay-Nachbarn untereinander sowie zwischen Overlayknoten und Master-Device verschlüsselt wird und ein Outsider per Definition kein Vorwissen über verwendete Schlüssel besitzt, wird das Schutzziel *Vertraulichkeit* erreicht.

Integrität übertragener Nachrichten

Um die *Integrität* der erhaltenen Nachrichten überprüfen zu können, ist die alleinige Verwendung eines symmetrischen Verschlüsselungsverfahrens nicht ausreichend. Wie bereits Abschnitt 2.4.2.3 erläutert ist zusätzlich die Berechnung und Übermittlung eines Hashwerts über die Nachrichten notwendig, um die Integrität empfangener Nachrichten prüfen zu können. Auf die Verwendung einer solchen Prüfsumme könnte nur in Fällen verzichtet werden, in denen die Nachrichten einen hohen Grad an Redundanz aufweisen. Da der Kommunikationsaufwand in Sensornetzen jedoch möglichst auf ein Minimum reduziert werden muss, enthalten die dort übertragenen Nachrichten kaum Redundanz. Eine Integritätsprüfung ohne Verwendung einer Prüfsumme ist somit also nicht möglich. In der Protokollbeschreibung von SCANv2 in [HoBZ04] findet sich kein Hinweis auf die Verwendung einer Prüfsumme. Ohne die Prüfsumme könnte ein Angreifer daher unbemerkt Protokollpakete modifizieren.

Angriffe auf die Verfügbarkeit

Da das Dienstverzeichnis SCANv2 als Overlaynetz realisiert wurde, hängt dessen *Verfügbarkeit* direkt von der Verfügbarkeit der Dienste unterer Netzschichten ab. Sofern diese keine Schutzmaßnahmen gegen DoS-Angriffe bieten, kann ein Angreifer mit einfachsten Maßnahmen eine Benutzung des Dienstverzeichnisses unterbinden. Sofern SCANv2 in Netzen mit sicherheitskritischen Anwendungen eingesetzt werden soll, kann jedoch davon ausgegangen werden, dass die dort eingesetzten Telekommunikationsprotokolle mit den entsprechenden Schutzmechanismen ausgestattet sind und damit diese Form des Angriffs erschwert wird.

Authentifizierung und Replay-Angriffe

Durch *Authentifizierung* werden Outsider von der Kommunikation mit dem Overlay ausgeschlossen. Die Authentifizierung kann anhand der symmetrischen Schlüssel erfolgen, die die legitimen Kommunikationspartner gemeinsam haben. Ein Angreifer hat jedoch die Möglichkeit die Kommunikation zweier Overlay-Nachbarn mitzuhören und deren Pakete zu einem späteren Zeitpunkt erneut in das Netz einzuspielen (siehe Abschnitt 2.4.1.1). Eine Methode diese Angriffe zu erschweren, besteht in der Verwendung von Zeitstempeln. SCANv2 macht aus diesem Grund im dritten Protokollschritt von Zeitstempeln Gebrauch (zur Problematik von Zeitstempeln siehe auch Abschnitt 3.3). Allerdings fehlt diese Schutzmaßnahme an einigen Stellen und ermöglicht einem Angreifer beispielsweise durch Wiederholung des 6. Schritts dem Zone-Owner ein altes Zonenzertifikat zuzuspielen.

3.2.2 Angriffe durch Insider

Gelingt es einem Angreifer an die symmetrischen Nachbarschaftsschlüssel eines Overlayknotens zu gelangen, kann er dessen Identität annehmen und sich gegenüber anderen Overlayknoten authentifizieren. Im Folgenden wird untersucht, ob dem Angreifer einer der in Abschnitt 2.4.1.3 vorgestellten Angriffe auf die CAN-Struktur ermöglicht wird.

Angriff durch freie Wahl der Overlay-Knoten-ID

Durch geschickte Wahl seines Join-Point kann ein Angreifer in CAN gezielt die Zuständigkeit für einen bestimmten Datensatz erlangen. Falls es dem Angreifer ermöglicht wird unter verschiedenen Identitäten dem Netz mehrfach beizutreten, so kann dieser sukzessive seinen Zuständigkeitsbereich erweitern und dadurch beliebige Datensätzen kontrollieren. In SCANv2 wird dieser Angriff verhindert, da dort der Join-Point durch das Master-Device zugewiesen wird und der Knoten, der seine Zone teilt, diese Zuweisung überprüfen kann. Ein mehrfaches Beitreten unter verschiedenen Identitäten wird ebenfalls unterbunden, da hierfür ein erneuter Kontakt mit dem Master-Device notwendig wäre.

Angriff durch Manipulation der Overlay-Routingtabelle

Ein weiterer Angriff auf DHTs besteht in der Verbreitung falscher Routinginformationen, um dadurch Datenverkehr umzuleiten oder die Routingtabelle anderer Knoten unbrauchbar zu machen. Da in CAN die Wahl der Nachbarn direkt von deren Zonenkoordinaten abhängt, ist eine Änderung der Routingtabelle hier nur durch die Wahl anderer Zonenkoordinaten möglich. Für das Dienstverzeichnis SCANv2 stellt dieser Angriff somit keine Gefahr dar, da eine Änderung der Routingtabelle sowie die Zuweisung der Zonenkoordinaten nur durch das Master-Device erfolgen kann.

Angriff auf die Weiterleitung von CAN-Nachrichten

Da eine Nachricht im CAN-Overlay über mehrere Zwischenknoten geleitet wird bis diese am Zielknoten ankommt, besteht die Möglichkeit, dass diese auf ihrem Weg zum Ziel von einem bösartigen Knoten modifiziert oder verworfen wird. Die Wahrscheinlichkeit auf einen bösartigen Knoten zu treffen, steigt dabei mit zunehmender Weglänge. Die Sicherheitsarchitektur von SCANv2 bietet gegenüber diesem Angriff keinen besonderen Schutz, so dass auch hier die Modifikation von Overlaynachrichten durch Insider-Angreifer möglich ist. Im Gegensatz zu der zuvor betrachteten Modifikation von Nachrichten durch Outsider-Angreifer, bieten in diesem Fall die verwendeten Verfahren zur Sicherstellung von Integrität und Authentizität keinen Schutz, da der Angreifer im Besitz gültiger Kommunikationsschlüssel ist.

Angriff auf die CAN-Datenspeicherung

Sobald es einem Angreifer gelungen ist die Zuständigkeit für eine CAN-Zone zu erlangen, kann dieser die dort gespeicherten Daten beliebig modifizieren oder verwerfen. Dies trifft ebenso auf die Speicherung von Dienstbeschreibungen im Dienstverzeichnis SCANv2 zu, sofern die diese auf einem bösartigen Knoten gespeichert werden. Da durch SCANv2 jedoch eine freie Wahl der Zonenkoordinaten verhindert wird, bleibt der Schaden durch einen Angreifer in diesem Fall lokal begrenzt.

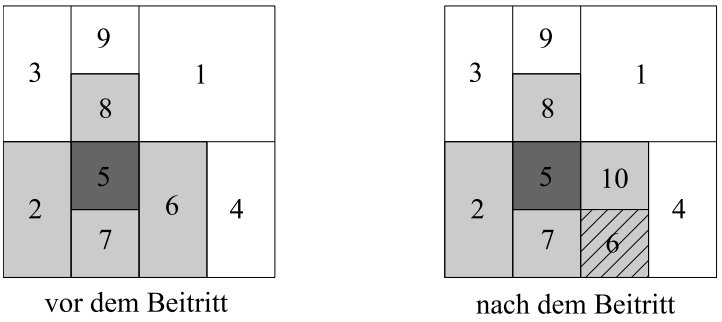

Abbildung 3.2: Unsichere Verwendung von Gruppenschlüsseln nach einer Zonenteilung

Unsichere Verwendung von Gruppenschlüsseln

Gruppenschlüssel werden in SCANv2 verwendet um im Fall eines Knotenausfalls eine sichere Kommunikation unter den Nachbarn des ausgefallenen Knotens zu ermöglichen. Eine Schwachstelle in SCANv2 führt dazu, dass nach dem Hinzufügen eines neuen Knotens in das Overlay sich einer der verwendeten Gruppenschlüssel im Besitz eines Knotens befindet, der nicht mehr Mitglied dieser Gruppe ist. Dieser Knoten kann somit das Wahlverfahren zur Behandlung eines Knotenausfalls (siehe Abschnitt 3.1.3) beeinflussen und möglicherweise die Zuständigkeit für die ausgefallene Zone erlangen.

Abbildung 3.2 zeigt ein Beispiel der Sicherheitslücke. Alle Knoten, die den gemeinsamen Gruppenschlüssel von Knoten 5 besitzen, sind in der Abbildung hellgrau hinterlegt. Durch den Beitritt von Knoten 10 wird die Zone von Knoten 6 geteilt. Nach der Teilung ist die Zone von Knoten 6 nicht mehr mit der Zone von Knoten 5 benachbart. Somit ist Knoten 6 im Fall eines Knotenausfalls von Knoten 5 nicht mehr berechtigt am Knotenausfallprotokoll teilzunehmen, besitzt jedoch noch den dazu benötigten Gruppenschlüssel und kann mit diesem Einfluss auf das Wahlverfahren nehmen.

Verfahren zum sicheren Einfügen und Auffinden von Diensten

Die bisherige Beschreibung von SCANv2 spezifiziert keinen Mechanismus, der das sichere Einfügen und Auffinden von Diensten ermöglicht. Da das Overlay keine Authentifizierung sowie Ende-zu-Ende Verschlüsselung über mehrere Hops ermöglicht, können bösartige Overlayknoten unbemerkt Dienstanfragen beantwortet, die nicht in Ihren Zuständigkeitsbereich fallen. Somit kann bereits ein einzelner Insider-Angriff die Funktionalität des Dienstverzeichnisses massiv beeinträchtigen. In dieser Form wird SCANv2 dem Ziel im Fall eines Insider-Angriff den Schaden lokal zu begrenzen daher nicht gerecht.

3.3 Eignung für Sensornetze

Die schlanke Sicherheitsarchitektur, die auf die Verwendung von Public-Key-Kryptographie auf den Sensorknoten verzichtet, sowie die dezentrale Kommunikationsstruktur des Overlays sind speziell auf die Anforderungen drahtloser Sensornetze

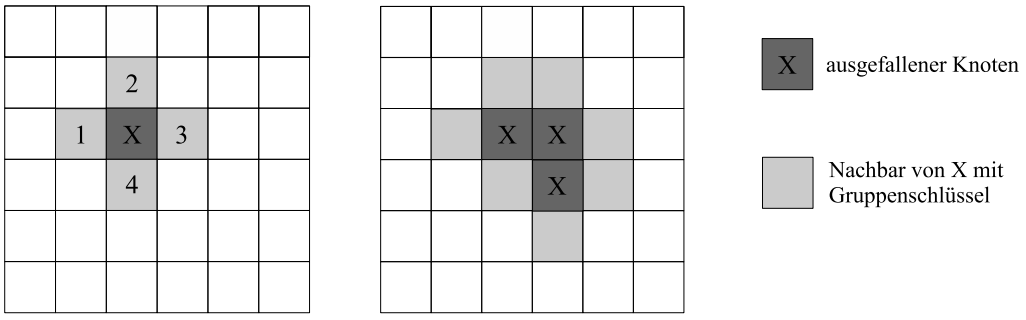

Abbildung 3.3: Verfahren zur Zonenübernahme nach einem Knotenausfall

zugeschnitten. Inwieweit sich der Einsatz einer kommunikationsaufwändigen Overlaystruktur mit den beschränkten Ressourcen eines Sensorknotens vereinbaren lässt, muss mittels entsprechender Simulationen untersucht werden.

Ein Kritikpunkt an dem vorgestellten Dienstverzeichnis ist die Unabhängigkeit der Sicherheitsarchitektur von den Sicherheitsmerkmalen darunterliegender Schichten, die zu einem unnötigen Verbrauch von knappen Ressourcen führen kann. Für sicherheitskritische Anwendungen, die das Anwendungsfeld dieses Dienstverzeichnisses sind, werden in der Regel bereits auf diesen Schichten Maßnahmen ergriffen, um das Netz vor Angriffen zu schützen. Durch eine schichtenübergreifende Sicherheitsarchitektur könnte die Anzahl der benötigen Schlüssel und notwendigen kryptographischen Operationen deutlich reduziert werden.

Problematisch scheint außerdem die Verwendung von Zeitstempeln, die in SCANv2 eingesetzt werden um Replay-Angriffe zu verhindern. Diese erfordern den aufwändigen Einsatz synchroner Uhren im Sensornetz.

Unzureichende Behandlung von Knotenausfällen

Das in Abschnitt 3.1.3 vorgeschlagene Verfahren zur Behandlung von Knotenausfällen weist mehrere gravierende Schwachstellen auf. Diese resultieren aus dem Umstand, dass das Master-Device als einziges vertrauenswürdiges Gerät zum Zeitpunkt des Knotenausfalls nicht zur Verfügung steht. Im übrigen Protokollablauf werden alle sicherheitskritische Änderungen der Overlaytopologie allein durch das Master-Device angefordert und legitimiert. Ohne Verwendung des Master-Device müssen die an der Knotenausfallbehandlung beteiligten Knoten selbständig Angriffe bösartiger Knoten abwehren und untereinander ein Vertrauensverhältnis aufbauen können.

Das vorgeschlagene Protokoll sieht vor, dass die Overlay-Nachbarn eines ausgefallenen Knotens unter Verwendung des gemeinsamen Gruppenschlüssels in Kontakt treten. In Abbildung 3.3 ist dieses Szenario dargestellt: Die vier Nachbarknoten des ausgefallenen Knotens X sind im Besitz des Gruppenschlüssels und müssen wählen, welcher der Knoten die ausgefallene Zonen übernehmen soll. Damit die Knoten miteinander kommunizieren können, benötigen diese eine Liste der Netzwerkadressen aller Nachbarknoten von X. Das in Abschnitt 3.1.2 dargestellte Protokoll sieht den Austausch dieser Adressen jedoch nicht vor. Zudem ist die Übertragung und das

Speichern der Liste relativ aufwändig, da für n Nachbarknoten insgesamt $n(n-1)$ Adressen gespeichert werden müssen[3].

Des Weiteren sind Gruppenschlüssel nicht dazu geeignet ein sicheres Wahlverfahren durchzuführen. Die Schlüssel ermöglichen zwar, dass die am Wahlverfahren beteiligten Knoten sich vor nicht am Verfahren teilnehmenden Knoten schützen können. Die Teilnehmer haben jedoch keine Möglichkeit sich untereinander als individuelle Knoten zu authentifizieren, da hierzu für allen Teilnehmerpaaren ein eigener Schlüssel erforderlich wäre. Die vorherige Verteilung solcher Schlüssel durch das Master-Device würde jedoch genau wie bei der bereits angesprochenen Netzwerkadressliste einen großen Speicherbedarf erfordern.

Problematisch ist auch der in Abbildung 3.3 im rechten Beispiel dargestellte gleichzeitige Ausfall mehrerer benachbarter Knoten. In diesem Fall muss das Wahlverfahren mit Hilfe einer reduzierten Anzahl von Nachbarn durchgeführt werden. Sofern viele Knoten von dem Ausfall betroffen sind, kann dies zum vollständigen Verlust eines Gruppenschlüssels führen. Schließlich stellt auch die eigentliche Erkennung eines Knotenausfalls ein Problem dar. Wird ein Ausfall allein durch das Ausbleiben periodischer Update-Nachrichten erkannt, kann durch gezielte DoS-Angriffe mit einfachen Mitteln der Knotenausfallmechanismus initiiert werden und damit ein gutartiger Knoten vom Overlay ausgeschlossen werden. Aufgrund der Tatsache, dass in Sensornetzen häufig mit dem Ausfall von Sensorknoten gerechnet werden muss, ist der bisherige Ansatz zur Behandlung von Knotenausfällen nicht ausreichend.

Insgesamt stellt das Dienstverzeichnis SCANv2 einen vielversprechenden Ansatz zur Dienstauffindung in Sensornetzen dar. Für den praktischen Einsatz müssen die genannten Schwachstellen jedoch noch verbessert werden.

[3] In einem 7-dimensionalen CAN hat ein Knoten im Schnitt 14 Nachbarn. Somit würde die Liste $14 \cdot 13 = 182$ Netzwerkadressen enthalten.

4. Optimierung von SCANv2

Die im Rahmen dieser Arbeit durchgeführten Optimierungen betreffen größtenteils den Entwurf eines Verfahrens um das sichere Einfügen und Auffinden von Diensten in SCANv2 zu ermöglichen. Des Weiteren werden einige der in Kapitel 3 erkannten Schwachstellen behoben. Abschließend wird diese verbesserte Version (SCANv3) des Dienstverzeichnisprotokolls beschrieben.

4.1 Sicheres Auffinden und Einfügen

Wie in Abschnitt 3.2.2 dargestellt, bietet CAN keinen Mechanismus um die Authentizität und Integrität von Overlay-Nachrichten zu verifizieren. Durch die Sicherheitsarchitektur von SCANv2 werden diese Schutzziele zumindest für die Kommunikation zwischen benachbarten Overlay-Knoten erreicht. SCANv2 bietet jedoch keine Möglichkeit den Urheber einer Overlay-Nachricht zu authentifizieren, wenn diese über mehrere Overlay-Hops zum Empfänger gelangt ist. Ein Insider-Angreifer kann somit zum Beispiel unbemerkt Suchanfragen für Dienste beantworten, die nicht in seinen Zuständigkeitsbereich fallen.

Die Grundidee des hier vorgestellten Verfahrens, das diesen Angriff erschweren soll, beruht auf der Etablierung eines gemeinsamen symmetrischen Sitzungsschlüssels zwischen dem anfragendem Knoten S und dem für den jeweiligen Dienst zuständigen Zielknoten D. Der eigentliche Austausch der angeforderten Daten kann somit über eine direkte, mit dem Sitzungsschlüssel geschützte Verbindungen zwischen den beiden Knoten außerhalb des Overlays erfolgen. Die Overlay-Kommunikation dient nur noch dem Austausch dieses Sitzungsschlüssels. Dadurch wird die kommunikationsaufwändige Benutzung des Overlays auf ein Minimum reduziert und die knappen Ressourcen der Sensorknoten werden geschont. Durch die Verwendung von Ende-zu-Ende Verschlüsselung, die durch den ausgetauschten Sitzungsschlüssel ermöglicht wird, werden die übertragenen Daten gegenüber unbeteiligten Knoten geheimgehalten und vor unbemerkter Manipulation durch bösartige Knoten geschützt.

In [EsGl02] wird ein Verfahren beschrieben wie unter Zuhilfenahme weiterer Knoten, die bereits im Besitz gemeinsamer Schlüssel sind, zwischen zwei Sensorknoten ein neuer symmetrischer Schlüssel etabliert werden kann. Diese Idee wird in [ChPS03]

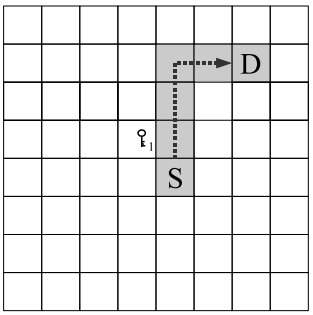

Abbildung 4.1: Schlüsselaustausch mit SPX

aufgegriffen und erweitert. Die Autoren schlagen vor, durch die Verwendung mehrerer unabhängiger Pfade zwischen den Knoten die Sicherheit dieses Schlüsselaustauschverfahrens noch zu verbessern. Die in den folgenden Abschnitten vorgeschlagenen Schlüsselaustauschverfahren basieren auf diesen Ideen. Die Tatsache, dass sich in einem CAN-Overlay die Routingtabelle (und damit auch der Weg, den ein Paket durch das Overlay nimmt) direkt aus den Zonenkoordinaten der Overlay-Teilnehmer ergibt und zwischen zwei Overlay-Knoten in der Regel mehrere disjunkte Pfade existieren, ermöglicht den Entwurf eines Schlüsselaustauschverfahrens, das zusätzlich eine Authentifizierung der Knoten anhand ihrer Zonenkoordinaten erlaubt. Im Folgenden werden zwei solche Verfahren vorgeschlagen, die das sichere Eintragen und Auffinden von Diensten in SCANv2 ermöglichen sollen.

4.1.1 Schlüsselaustauschverfahren SPX

Das Schlüsselaustauschverfahren SPX (*Single Path key eXchange*) ähnelt dem *Path-Key-Establishment* wie es in [EsGl02] vorgeschlagen wird. Der anfragende Knoten S beginnt den Schlüsselaustausch mit der Erzeugung eines zufälligen Schlüssels k. Dieser wird zusammen mit der Netzwerkadresse von S über das Overlay in Richtung D geroutet. In Abbildung 4.1 ist dieser Weg des Schlüssels durch das Overlay graphisch dargestellt. Da die Kommunikation zwischen zwei CAN-Nachbarn in SCANv2 verschlüsselt erfolgt, kann der Schlüssel k nur von den h CAN-Knoten, die auf dem Weg zu D liegen, mitgelesen werden. Die durchschnittliche Pfadlänge h hängt dabei von der Gesamtknotenanzahl N sowie von der Dimensionalität d des CAN ab und kann mit Gleichung (2.1) aus Abschnitt 2.3 abgeschätzt werden (siehe dort auch Abbildung 2.6).

Sofern alle $h-1$ Knoten auf dem Weg zwischen S und D gutartig sind und den Schlüssel k nach dessen Übertragung ordnungsgemäß vernichten, kann dieser von S und D als symmetrischer Sitzungschlüssel verwendet werden. Ein einzelner bösartiger Knoten auf dem Pfad ist jedoch ausreichend, um den Schlüssel nutzlos werden zu lassen. Bei einem Anteil von m bösartige Knoten im Netz beträgt die Wahrscheinlichkeit, dass alle Knoten auf dem Pfad sowie der Zielknoten D gutartig sind $(1-m)^h$. Die Gesamtwahrscheinlichkeit p mit diesem Verfahren zu einem sicheren Sitzungsschlüssel zu gelangen, beträgt somit:

$$p = (1-m)^{\frac{d}{4}N^{\frac{1}{d}}} \tag{4.1}$$

In SCANv2 ermöglicht der so erzeugte Sitzungsschlüssel k einem rechtmäßigen Zone-Owner sich gegenüber dem anfragenden Knoten zu authentifizieren. Ein Insider-

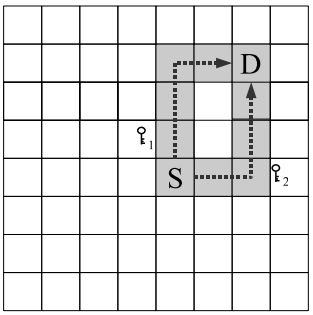

Abbildung 4.2: Schlüsselaustausch mit MPX

Angreifer kann somit nur noch dann unbemerkt Dienstanfragen beantworten, die nicht in seinen Zuständigkeitsbereich fallen, falls er als einer der Zwischenknoten während des Schlüsselaustauschverfahrens in den Besitz des Schlüssels k gelangt ist. Die Wahrscheinlichkeit p aus Gleichung 4.1 entspricht daher auch der Wahrscheinlichkeit, dass (bei Verwendung von SPX) das Einfügen oder Auffinden eines Dienstes bei einem Anteil von m Insider-Angreifern korrekt durchgeführt werden kann. Falls zu einem späteren Zeitpunkt erneut eine Kommunikation zwischen den Knoten S und D erforderlich wird um beispielsweise einen Diensteintrag zu aktualisieren, kann der Sitzungsschlüssel zwischengespeichert und wiederverwendet werden. Somit kann ein erneuter kommunikationsaufwändiger Schlüsselaustausch entfallen.

4.1.2 Schlüsselaustauschverfahren MPX

Ist das durch SPX erzielte Sicherheitsniveau für ein bestimmtes Anwendungszenario nicht ausreichend, kann durch das Schlüsselaustauschverfahren MPX (*Multi Path key eXchange*) die Wahrscheinlichkeit für eine korrekte Lookup- oder Insert-Operation noch erhöht werden. Der Nachteil dieses Verfahrens ist ein gesteigerter Kommunikationsaufwand im Vergleich zu SPX.

Das Verfahren MPX verwendet ebenfalls das CAN-Overlay um einen zufälligen Schlüssel k zwischen S und D auszutauschen. Um die Sicherheit gegenüber SPX zu erhöhen wird der Schlüssel auf r verteilte Geheimnisse k_i aufgeteilt und über *disjunkte Pfade* zum Zielknoten D geschickt (siehe Abbildung 4.2) und dort wieder zusammengesetzt. Der Ansatz, mehrere Pfade zwischen zwei Sensorknoten zu verwenden, um einen gemeinsamen Schlüssel zu etablieren, wurde erstmals in [ChPS03] unter dem Namen *Multipath-Key-Reinforcement* veröffentlicht.

Für die weiteren Betrachtungen wird eine Menge von Pfaden als *disjunkt* bezeichnet, falls die Pfade untereinander bis auf den Start- und Zielknoten keine weiteren Knoten gemeinsam haben. Die Overlay-Topologie von SCANv2 eignet sich besonders für den Einsatz eines Mehrwegeausbreitungsverfahrens, da in einem d-dimensionalen CAN-Overlay zu einem Ziel im Schnitt d disjunkte Pfade mit annähernd optimaler Länge zur Verfügung stehen. Dies ergibt sich aus der Tatsache, dass in der Regel anfangs in jeder der d Dimensionen eine Nachbarzone existiert, die näher am Zielknoten liegt als die eigene Zone. Lediglich in Netzen mit geringer Knotenzahl muss von einer deutlichen kleineren Anzahl von Pfaden ausgegangen werden, da die Gesamtzahl der Knoten nicht ausreichend ist, um jedem Knoten $2d$ disjunkte Nachbarknoten zuzuteilen. Dieser Effekt kann auch an den in Abschnitt 6.3.1 dargestellten Simulationsergebnissen nachvollzogen werden.

Wird der Schlüssel mit $k = k_1 \oplus k_2 \oplus \ldots \oplus k_r$ aufgeteilt und über r disjunkte Pfade verschickt, können bis zu $r-1$ Pfade mit bösartigen Knoten besetzt sein, ohne dass der Angreifer Informationen über den übertragenen Schlüssel erhält. Da der Zielknoten in diesem Fall den Schlüssel jedoch nur rekonstruieren kann, falls er alle Schlüsselteile k_i unverändert erhalten hat, kann ein Angreifer bereits durch Modifikation eines einzelnen Schlüsselteils die Rekonstruktion des Schlüssels verhindern. Damit dieser Angriff auf die Verfügbarkeit erschwert und bösartige Knoten auf einzelnen Pfaden toleriert werden können, kann das Verfahren so ausgelegt werden, dass nur r der s möglichen Pfade gutartig sein müssen, wobei $r > \frac{s}{2}$. Dazu verschickt der anfragende Knoten S wie oben erläutert den Schlüssel verteilt über r zufällig ausgewählte Pfade an den Zielknoten S. Daraufhin überprüfen S und D mit Hilfe eines Challenge-Response-Verfahrens, ob der Schlüssel korrekt übertragen wurde (siehe Abschnitt 2.4.2.3). Falls der Schlüssel durch einen Angreifer unbrauchbar gemacht wurde, wird der Schlüsselaustausch so lange mit einer neuen, zufälligen Auswahl an Pfaden wiederholt, bis das Challenge-Response-Verfahren erfolgreich verläuft. Die Wahrscheinlichkeit, dass mindestens r der s möglichen Pfade nur aus gutartigen Knoten bestehen beträgt

$$\sum_{i=r}^{s} \binom{s}{i} ((1-m)^h)^i (1-(1-m)^h)^{s-i}.$$

Die Gesamtwahrscheinlichkeit p, dass mit MPX erfolgreich ein sicherer Sitzungsschlüssel etabliert wird, beträgt in einem d-dimensionalen CAN mit der Pfadlänge h aus Gleichung (2.1) somit:

$$p = \sum_{i=r}^{s} \binom{s}{i} ((1-m)^{\frac{d}{4}N^{\frac{1}{d}}})^i (1-(1-m)^{\frac{d}{4}N^{\frac{1}{d}}})^{s-i} \tag{4.2}$$

Ein Nachteil dieses Verfahrens ist der hohe Kommunikationsaufwand, der durch einen Angreifer verursacht wird, falls dieser Schlüsselteile manipuliert und das Verfahren deshalb mehrfach wiederholt werden muss. Die Wiederholungen können vermieden werden, falls der Zielknoten manipulierte Schlüsselteile erkennen kann. Die so genannte *Schwellwert-Kryptographie* ermöglicht ein Geheimnis so auf s Teilnehmer aufzuteilen, dass mindestens r Teilnehmer (mit $r < s$) kooperieren müssen, um das Geheimnis zu rekonstruieren. Das Verteilen der Geheimnisse auf verschiededene Teilnehmer entspricht im Fall von SCANv2 dem Versenden über disjunkte Pfade. Viele der Schwellwert-Verfahren basieren auf Shamirs *Secret-Sharing* [Sham79]. Diese Verfahren bietet jedoch ebenfalls keine Möglichkeit die von einem Angreifer manipulierten Geheimnisse zu erkennen. Mit *Verifiable-Secret-Sharing* (VSS)[GeRR98] gelingt die Rekonstruktion eines Geheimnisses auch, falls einige Teilnehmer ihre Geheimnisteile modifiziert haben. Um eine erfolgreiche Rekonstruktion des Schlüssels zu ermöglichen, muss der Anteil der bösartigen Teilnehmer $< \frac{1}{2}$ sein. In [GeRR98] schlagen die Autoren ein einfaches VSS-Verfahren vor, das durch seinen geringen Rechen- und Kommunikationsaufwand für den Einsatz in Sensornetzen geeignet erscheint und eine effizientere Verwendung von MPX ermöglichen könnte. Im Rahmen dieser Arbeit wird MPX aus Komplexitätsgründen durch Verwendung des obigen Challenge-Response-Verfahrens evaluiert.

4.1.3 Verwendung von Replikaten

Der Einsatz von Replikaten in SCANv2 stellt ein weiteres Verfahren dar mit dem die Wahrscheinlichkeit für das korrekte Auffinden eines Dienstes erhöht werden kann.

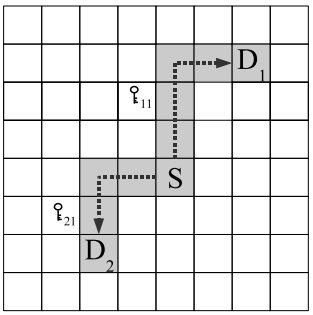

Abbildung 4.3: Schlüsselaustausch mit SPX und Replikaten

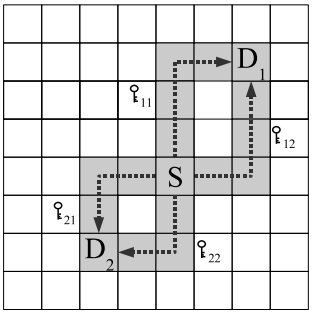

Abbildung 4.4: Schlüsselaustausch mit MPX und Replikaten

Falls der Hashwert eines Dienstnamens in den Zuständigkeitsbereich eines bösartigen Knoten fällt und keine Replikate verwendet werden, sind alle unter diesem Namen gespeicherten Dienstbeschreibungen verloren. Das gleiche gilt für den Fall, dass der ausgetauschte Sitzungsschlüssel, der für das Auffinden eines Dienstes verwendet wird, in den Besitz eines Angreifers gelangt ist.

Bei der Verwendung von Replikaten wird der gleiche Datensatz auf verschiedenen Knoten im Netz gespeichert. Durch die Benutzung mehrerer verschiedener Hashfunktionen können die Replikate gleichmäßig auf die Knoten des Sensornetzes verteilt werden. Ein ähnlicher Effekt kann durch einfaches Anhängen einer fortlaufenden Nummer an den zu suchenden Dienstnamen erreicht werden, bevor dieser durch die Hashfunktionen auf einen Punkt im CAN-Overlay abgebildet wird.

Die Verwendung von Replikaten kann sowohl mit SPX als auch mit MPX kombiniert werden und ist in den Abbildungen 4.3 und 4.4 dargestellt. Durch die gleichmäßige Streuung der Hashfunktion ist auch in diesem Fall die Wahrscheinlichkeit hoch, dass die benötigten Sitzungsschlüssel über disjunkte Pfade ausgetauscht werden können und somit eine kleine Anzahl von Angreifern effektiv daran gehindert wird, die Kontrolle über Mehrzahl der Replikate eines Dienstes zu erlangen. Damit eine ausreichende Anzahl an Pfaden zur Verfügung steht, sollte jedoch analog zum Schlüsselaustauschverfahren MPX die Dimensionalität des CAN-Overlay entsprechend hoch ($\approx$ Anzahl der Replikate) gewählt werden. Durch gleichzeitiges Anfordern mehrerer Replikate kann ein anfragender Knoten die Wahrscheinlichkeit für das korrekte Auffinden eines Dienstes erhöhen. Die einzelnen Antworten können dann miteinander verglichen werden und ungültige Datensätze mit Hilfe eines Mehrheitsentscheids erkannt und ignoriert werden. Sind also weniger als die Hälfte der verwendeten Repli-

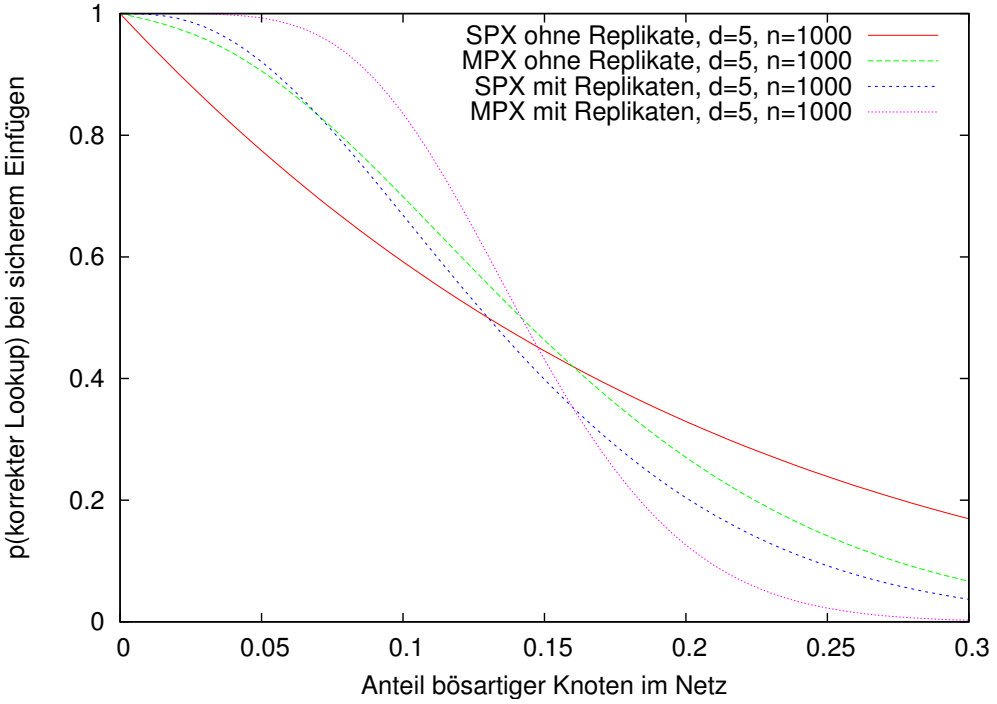

Abbildung 4.5: Wahrscheinlichkeit, dass in einem 5-dimensionalen CAN mit $n = 1000$ Knoten eine korrekte Lookup-Operation durchgeführt werden kann, ohne die Berücksichtigung von Fehlern beim Einfügen der Daten.

kate bösartig bzw. die verwendeten Sitzungsschlüssel im Besitz des Angreifer, können alle Knoten mit diesem Dienst korrekt aufgefunden werden. Der Kommunikationsaufwand dieses Verfahrens kann reduziert werden, indem der Sensorknoten zuerst nur einen Teil der Replikate anfordert und überprüft. Sind die Dienstbeschreibungen identisch, kann die Abfrage der übrigen Replikate entfallen. Anderenfalls werden mit einer zweiten Anfrage die übrigen Replikate angefordert und wie bisher mittels Mehrheitsentscheid manipulierte Dienstbeschreibungen identifiziert.

4.1.4 Analyse der Verfahren

Um die Effektivität der vorgeschlagenen Methoden zum sicheren Einfügen und Auffinden von Diensten vergleichen zu können, wurde deren theoretische Erfolgswahrscheinlichkeit anhand der Gleichungen (4.1) und (4.2) für unterschiedliche Gesamtknotenanzahl und CAN-Dimensionalität berechnet. Die Anzahl der verwendeten Pfade für MPX sowie die Anzahl der Replikate wurde entsprechend gleich der Dimensionalität d des CAN-Overlays gewählt. Auf diese Weise erfolgt eine optimale Ausnutzung aller d zur Verfügung stehenden Pfade.

Abbildung 4.5 zeigt für die verschiedenen Schlüsselaustauschverfahren eine Abschätzung, mit welcher Wahrscheinlichkeit in einem 5-dimensionalen CAN-Overlay mit $n = 1000$ Knoten eine Lookup-Operation das korrekte Ergebnis liefert, falls ein bestimmter Anteil der Overlay-Knoten bösartig ist. Die Berechnungen zeigen, dass die aufwändigeren Verfahren MPX sowie die Verwendung von Replikaten bei einem geringen Anteil bösartiger Knoten zu wesentlich höheren Erfolgswahrscheinlichkeiten führen, als die einfache Verwendung von SPX ohne Replikate. Durch die Kombination von MPX mit der Verwendung von Replikaten lässt sich dieses Ergebnis nochmals

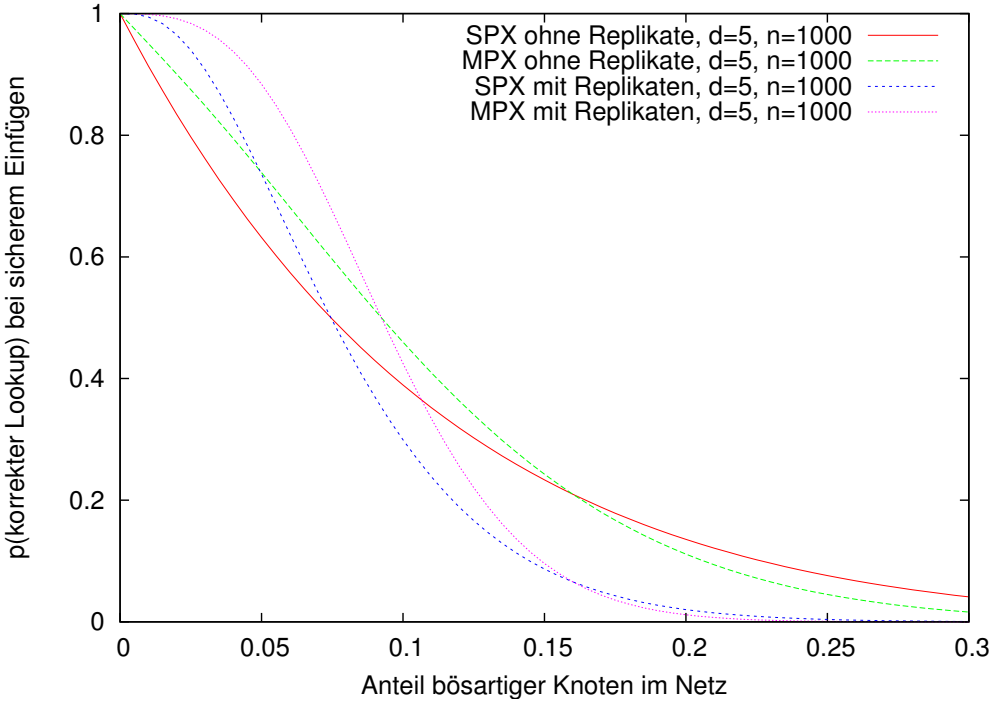

Abbildung 4.6: Wahrscheinlichkeit, dass in einem 5-dimensionalen CAN mit $n = 1000$ Knoten eine korrekter Lookup-Operation durchgeführt werden kann, falls die Daten unter Verwendung von SPX eingefügt wurden.

deutlich verbessern. Auffallend ist das schlechte Abschneiden der aufwändigeren Verfahren, falls der Anteil bösartiger Knoten 15% übersteigt. Durch den hohen Anteil bösartiger Knoten wird in diesem Fall die Mehrzahl der möglichen Pfade von einem Angreifer kontrolliert. Der in den aufwändigeren Verfahren eingesetzte Mehrheitsentscheid schlägt hier somit ins Gegenteil um. Da 15% in einem Netz mit $n = 1000$ Knoten jedoch immerhin 150 Knoten sind, die ein Angreifer einzeln manipulieren müsste, dürfte das schlechtere Abschneiden der Verfahren bei einem so hohen Anteil bösartiger Knoten in der Praxis keine bedeutende Rolle spielen.

Für die Darstellung der Erfolgswahrscheinlichkeit in Abbildung 4.5 wurde nicht berücksichtigt, dass ein Angreifer bereits das korrekte Einfügen eines Dienstes verhindern und somit indirekt Einfluss auf das spätere Auffinden des Dienstes nehmen kann. In den Abbildung 4.6 und 4.7 wird daher dargestellt, wie sich die Erfolgswahrscheinlichkeit für das korrekte Auffinden eines Dienstes verändert, wenn Angriffe beim Einfügen des Dienstes mitberücksichtigt werden. Wird, wie in Abbildung 4.6 gezeigt, zum Einfügen der Dienste das einfache SPX-Verfahren verwendet, führt dies im Vergleich mit Abbildung 4.5 zu einer deutlich niedrigeren Erfolgswahrscheinlichkeit für eine Lookup-Operation. Durch die Verwendung des sichereren Schlüsselaustauschverfahrens MPX zum Einfügen neuer Dienste, kann die Erfolgswahrscheinlichkeit, wie in Abbildung 4.7 dargestellt, jedoch verbessert werden.

Einen guten Kompromiss zwischen erforderlichem Kommunikationsaufwand und gewonnener Sicherheit scheint somit die Verwendung von *MPX* zum Einfügen sowie *SPX mit Replikaten* zum Auffinden von Diensten darzustellen. Die folgenden Abbildungen 4.9 bis 4.10 sollen zeigen, welchen Einfluss die Dimensionalität des CAN-

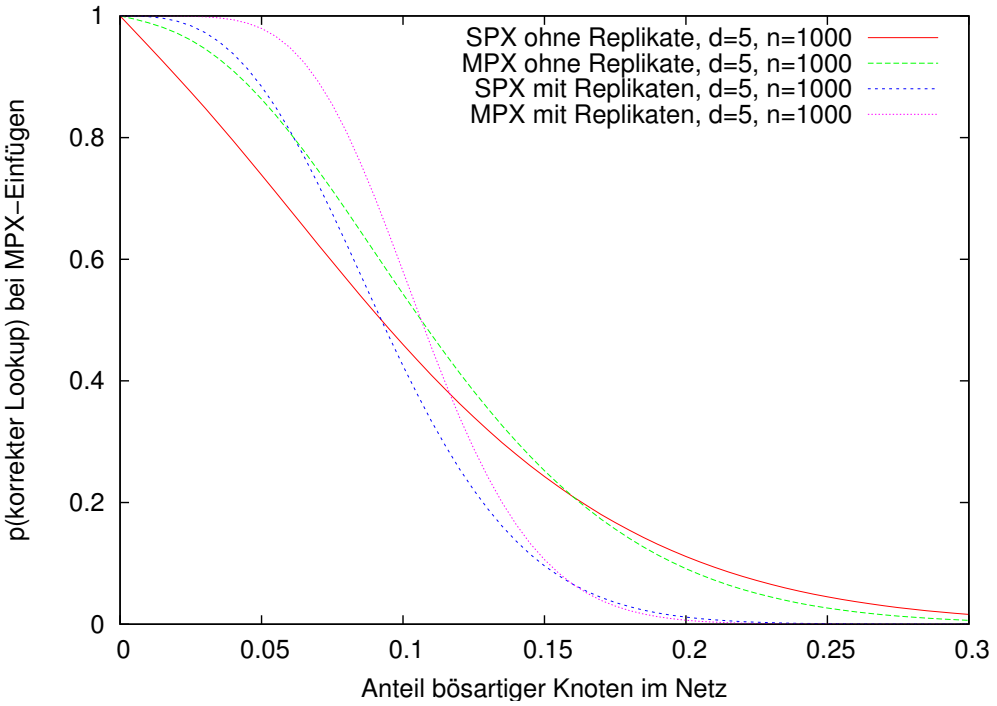

Abbildung 4.7: Wahrscheinlichkeit, dass in einem 5-dimensionalen CAN mit $n = 1000$ Knoten eine korrekte Lookup-Operation durchgeführt werden kann, falls die Daten unter Verwendung von MPX eingefügt wurden.

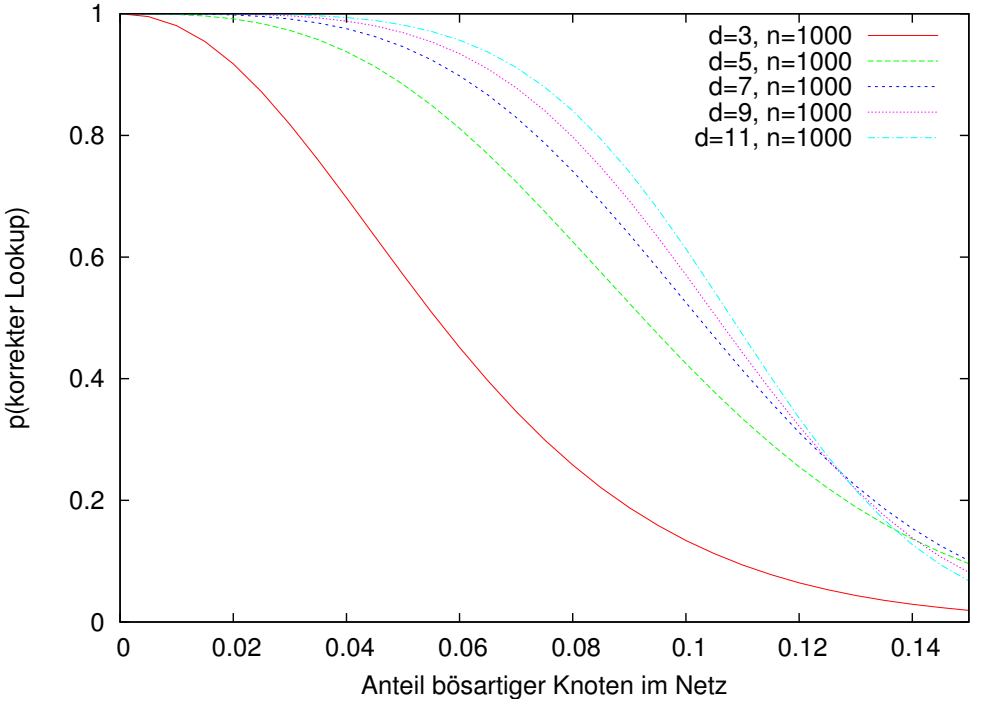

Abbildung 4.8: Wahrscheinlichkeit, dass in einem d-dimensionalen CAN mit $n = 1000$ Knoten unter Verwendung des vorgeschlagenen Verfahrens eine korrekte Lookup-Operation durchgeführt werden kann.

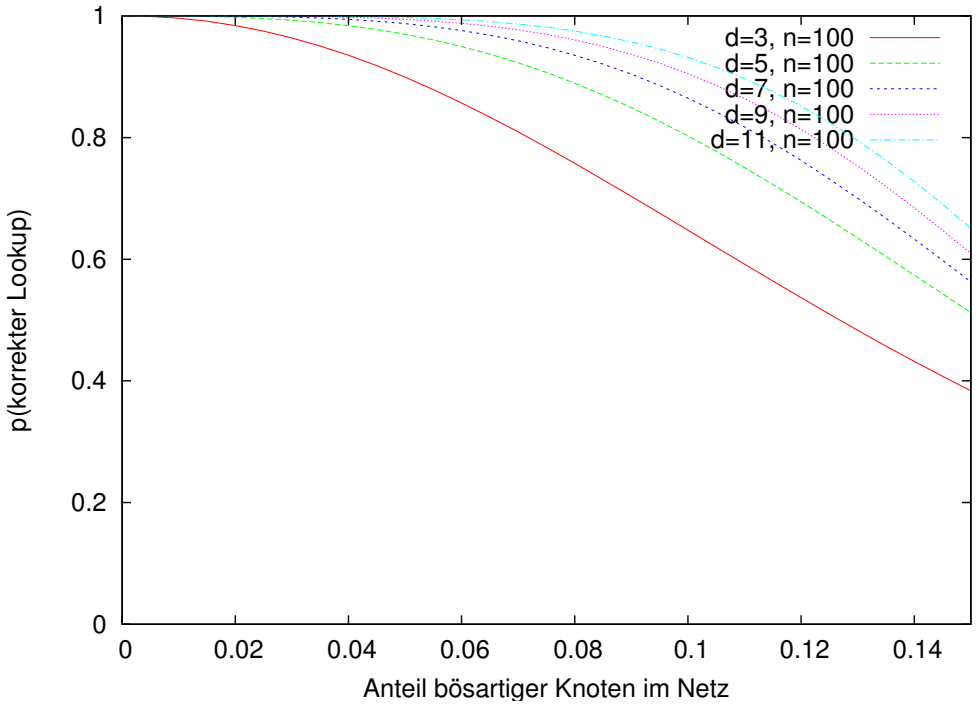

Abbildung 4.9: Wahrscheinlichkeit, dass in einem d-dimensionalen CAN mit $n = 100$ Knoten unter Verwendung des vorgeschlagenen Verfahrens eine korrekte Lookup-Operation durchgeführt werden kann.

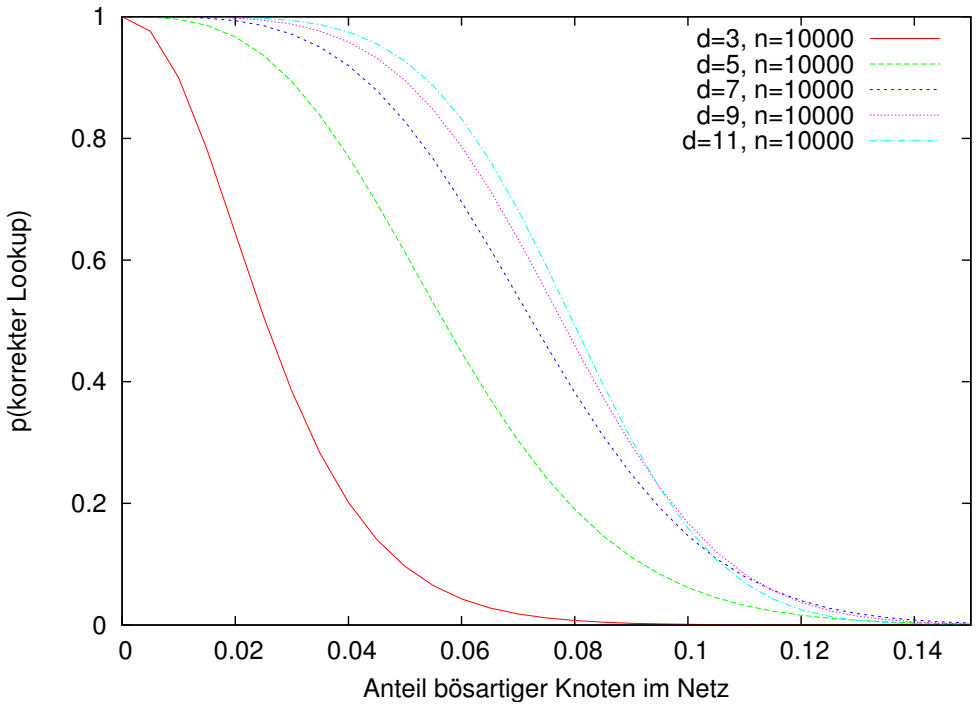

Abbildung 4.10: Wahrscheinlichkeit, dass in einem d-dimensionalen CAN mit $n = 10000$ Knoten unter Verwendung des vorgeschlagenen Verfahrens eine korrekte Lookup-Operation durchgeführt werden kann.

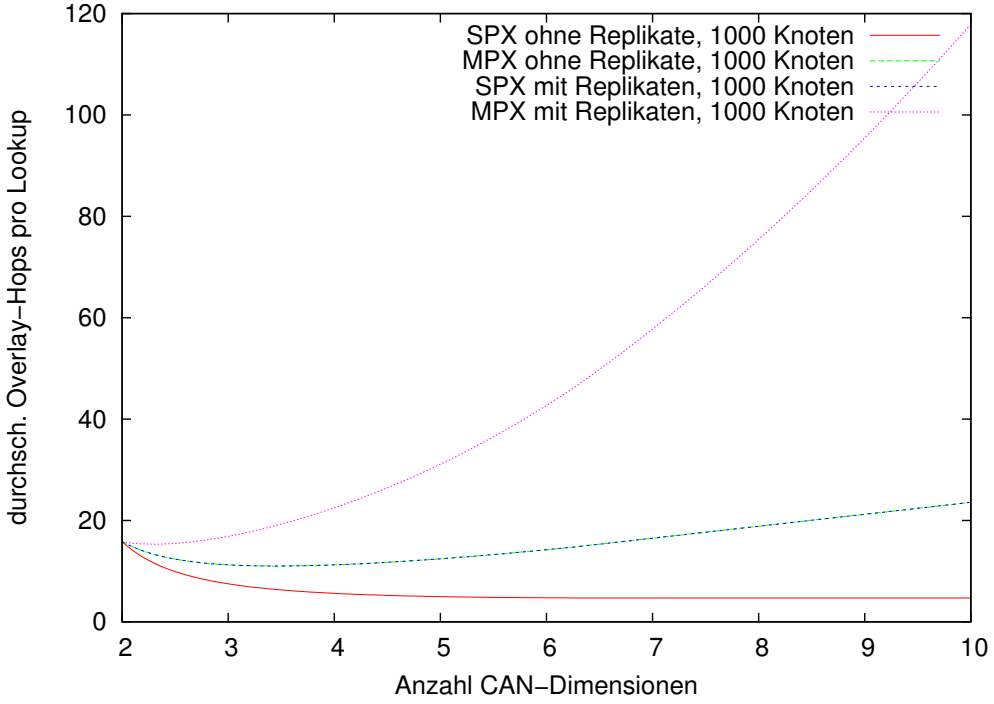

Abbildung 4.11: Anzahl der durchschnittlich benötigten Overlay-Hops in einem d-dimensionalen CAN um einen Sitzungsschlüssel auszutauschen.

Overlays und die damit verknüpfte Anzahl der Pfade und Replikate auf die Erfolgswahrscheinlichkeit haben, wenn das vorgeschlagene Verfahren verwendet wird. In Abbildung 4.8 sind die Wahrscheinlichkeiten für das erfolgreiche Auffinden eines Dienstes in einem Sensornetz mit $n = 1000$ Knoten dargestellt. Die Berechnungen zeigen, dass mit Hilfe einer größeren Anzahl CAN-Dimensionen die Erfolgswahrscheinlichkeit deutlich erhöht werden kann. Dies resultiert zum einen aus der kürzeren durchschnittlichen Pfadlänge. Durch diese steigt die Wahrscheinlichkeit, dass ein Pfad nur von gutartigen Knoten besetzt ist und ein erfolgreicher Schlüsselaustausch stattfinden kann. Zum anderen kann durch die Verwendung einer größeren Anzahl disjunkter Pfade die Effektivität von MPX und Replikaten verbessert werden. In einem hochdimensionales CAN fällt der Gewinn für eine Erhöhung der Dimensionalität jedoch zunehmend geringer aus, so dass der dadurch verursachte Kommunikationsaufwand nicht mehr gerechtfertigt scheint.

In Abbildung 4.9 ist zum Vergleich die Erfolgswahrscheinlichkeit für ein Netz mit $n = 100$ Knoten dargestellt. Die wesentlich höheren Erfolgswahrscheinlichkeiten im Vergleich zu einem Netz mit $n = 1000$ Knoten resultieren aus der kürzeren durchschnittlichen Pfadlänge im Overlay. Daher fällt auch der durch eine höhere CAN-Dimensionalität erzielte Gewinn im Verhältnis zu einem größeren Netz wesentlich geringer aus. In einem großen Netz mit $n = 10000$ Knoten wird analog zu den vorherigen Überlegungen bei einem gleichen Anteil bösartiger Knoten ein scheinbar deutlich schlechteres Ergebnis erzielt. Bei diesen Überlegung sollte jedoch beachtet werden, dass die absolute Anzahl der bösartigen Knoten, die auf den Schaubildern dargestellt sind, linear mit der Gesamtzahl der dargestellten Knoten wächst. Die 15% bösartigen Knoten in Abbildung 4.9 entsprechen 15 Knoten, während es in Abbildung 4.10 bereits 1500 Knoten sind. Da davon ausgegangen werden kann, dass die

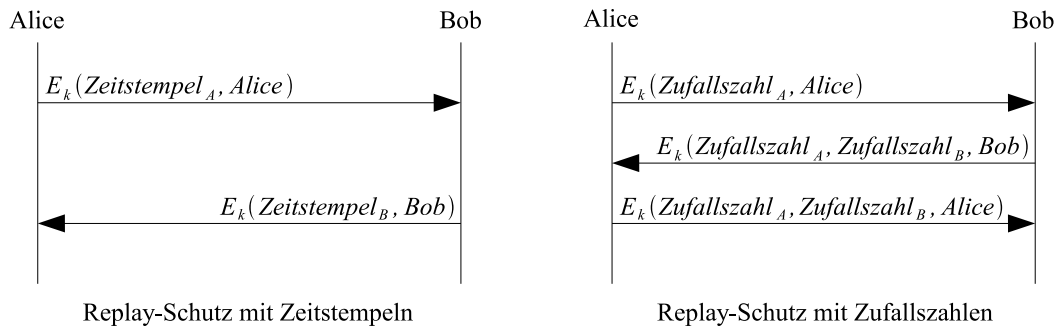

Abbildung 4.12: Schutz vor Replay-Angriffen durch Zeitstempel und Zufallszahlen

physische Manipulation eines Knotens durch einen Angreifer pro Knoten eine konstante Zeitspanne in Anspruch nimmt, scheint eine Manipulation hunderter Knoten nicht praktikabel.

Anhand der durchgeführten theoretischen Überlegungen bieten die vorgeschlagenen Verfahren daher auch für größere Netzen ein ausreichendes Maß an Sicherheit. Dieses wird jedoch durch einen deutlich gesteigerten Kommunikationsaufwand erkauft. Abbildung 4.11 zeigt die Gesamtzahl der für einen Schlüsselaustausch benötigten Overlay-Hops. Insbesondere der durch Kombination von *MPX mit Replikaten* benötigte Kommunikationsaufwand scheint in großen Netzen mit tausenden Knoten nicht praktikabel. Eine weitergehende Bewertung der Praktikabilität der Verfahren erfolgt in Kapitel 6 anhand der Ergebnisse der durchgeführten Simulationen.

4.2 Weitere Optimierungen

In diesem Abschnitt wird beschrieben, wie einige der in Kapitel 3 dargestellten Schwachstellen des Protokolls behoben werden können.

Replay-Angriffe und Zeitstempel

Das bisherige Protokoll zum Hinzufügen neuer Knoten in SCANv2 bietet nur in Ansätzen Schutz vor Replay-Angriffen. Bisher wird dieser Schutz durch die Verwendung von Zeitstempeln realisiert. Dies erfordert jedoch den Einsatz synchroner Uhren auf den Sensorknoten. Diese Problematik kann umgangen werden, indem der Zeitstempel durch eine jeweils nur einmal verwendete Zahl (so genannter *Nonce*) ersetzt wird. Ein Nonce kann beispielsweise zufällig oder durch einen fortlaufenden Zähler generiert werden. Es muss lediglich sichergestellt werden, dass eine Wiederholung der Zahl unwahrscheinlich ist.

Die beiden Kommunikationspartner Alice und Bob können ihre Kommunikation mit den folgenden Schritten vor Replay-Angriffen schützen: Alice verschlüsselt den Nonce zusammen mit den Nutzdaten und schickt diese an Bob. Dieser antwortet mit einer verschlüsselten Nachricht, die den gleichen Nonce enthält, den er zuvor von Alice empfangen hat. Damit sichergestellt werden kann, dass Bob im Besitz des gemeinsamen Schlüssels ist und er deshalb die Nachricht entschlüsseln konnte, darf die von Bob gesendete Nachricht nicht identisch mit der Nachricht von Alice sein. Falls die Nachricht keine zusätzlichen Nutzdaten enthält, die eine Unterscheidung ermöglichen

würde, kann der Nachrichteninhalt zum Beispiel um die jeweilige Absenderadresse ergänzt werden. Gegenüber dem Zeitstempel-Verfahren muss bei Verwendung eines Nonce in der Regel eine zusätzliche Nachricht gesendet werden. In Abbildung 4.12 ist der Ablauf der beiden Verfahren nochmals dargestellt.

In SCANv3 werden die Zeitstempel durch Nonces ersetzt. Damit möglichst wenig zusätzliche Daten übertragen werden müssen und insbesondere das Verschicken zusätzlicher Nachrichten vermieden werden kann, werden die Nonces in SCANv3 durch den Hashwert bereits übermittelter Daten gebildet. Das Master-Device authentisiert sich gegenüber seinen Kommunikationspartnern durch Übertragung des Hashwertes über den Gruppenschlüssel uid_{grpZO}. Für die Gegenrichtung wird der Hashwert über den Join-Point des hinzukommenden Knotens uid_{JP} verwendet. Durch die Anwendung der Hashfunktion werden kurze Nonces mit konstanter Länge erzeugt. Der Join-Point eines hinzukommenden Knotens sowie der Gruppenschlüssel eines Zone-Owners werden zufällig gewählt und bei jedem Beitritt eines neuen Knotens geändert. Da deren Hashwerte mit hoher Wahrscheinlichkeit bei jedem Join-Vorgang einen neuen Wert annehmen, können diese als Nonce verwendet werden. In SCANv3 werden in den Protokollschritten, die vor Replay-Angriffen geschützt werden sollen, die Nachrichten um die Nonces uid_{JP} und uid_{grpZO} erweitert. Der Einsatz synchroner Uhren ist in der verbesserten Protokollversion nicht mehr erforderlich, da auf die Verwendung von Zeitstempeln verzichtet wird.

Erzeugung neuer Gruppenschlüssel

In Abschnitt 3.2.2 wurde gezeigt, dass nach einer Zonenteilung einer der Nachbarknoten der geteilten Zone im Besitz eines Gruppenschlüssels einer Gruppe ist, in der er aufgrund der Zonenteilung nicht mehr Mitglied ist (siehe auch Abbildung 3.2). Dadurch kann der Knoten unberechtigt Einfluss auf das Verfahren zur Behandlung von Knotenausfällen nehmen. In SCANv3 erzeugt der die betroffene Zone verwaltende Knoten nach der Zonenteilung einen neuen Gruppenschlüssel und verteilt diesen durch eine *UPDATE_GRPKEY* Nachricht an seine Nachbarn. In dem in Abbildung 3.2 dargestellten Beispiel verteilt Knoten 5 den neuen Gruppenschlüssel an seine Nachbarn 2, 7, 8 und 10. Der alte Gruppenschlüssel von Knoten 6 wird damit ungültig. Durch diese Maßnahme wird die Sicherheitslücke geschlossen.

Verzicht auf Zertifikate

Das Master-Device verwendet in SCANv2 Zonenzertifikate um die Zonenkoordinaten eines Knotens überprüfen zu können. Nach jeder Zonenteilung stellt das Master-Device durch digitales Signieren der Zonenkoordinaten ein neues Zertifikat aus und übergibt es an den Besitzer der Zone. Zu einem späteren Zeitpunkt kann der Knoten dann durch Vorweisen des Zertifikats den rechtmäßigen Besitz der Zone gegenüber dem Master-Device beweisen. Die Erzeugung des gemeinsamen Schlüssels von Master-Device und Sensorknoten erfolgt in SCANv2 durch Verschlüsselung des Join-Points mit dem Superschlüssel key_{SK} des Master-Device (siehe Abschnitt 3.1.2).

In SCANv3 wird auf die Verwendung von Zertifikaten verzichtet. Damit das Master-Device weiterhin die Zonenkoordinaten eines Overlay-Knotens verifizieren kann, wird der gemeinsame Schlüssel von Master-Device und Sensorknoten durch Verschlüsselung der Zonenkoordinaten und der Netzwerkadresse des Sensorknotens generiert.

Teilt ein Knoten seine Zone, wird ihm vom Master-Device ein neuer Schlüssel $key_{ZO} = E_{SK}(hash(zone_{ZO}|addr_{ZO}))$ zugewiesen. Bei der Verwendung dieses Schlüssel erfolgt somit implizit eine Überprüfung der Zonenkoordinaten durch das Master-Device. Da die Erzeugung des Schlüssels auf dem Master-Device auch durch symmetrische Verschlüsselung mit dem Superschlüssel key_{SK} erfolgen kann, werden auf dem Master-Device keine asymmetrischen Kryptoverfahren mehr verwendet. Des Weiteren kann auf den Sensorknoten die Speicherung ihres Join-Points unterbleiben, da dieser für die Schlüsselerzeugung nicht mehr benötigt wird.

Einfacher Protokollablauf

Das Protokoll zum Hinzufügen neuer Knoten in SCANv2 sieht vor, dass nach einer Zonenteilung die Liste mit Nachbarschaftsinformationen, die unter anderem die neuen Nachbarschaftsschlüssel enthält, zuerst an den Zone-Owner übermittelt wird. Dieser verwendet dann die Informationen der Liste um die von der Zonenteilung betroffenen Nachbarn über die Teilung zu informieren. Durch diesen Ablauf müssen die kompletten Nachbarschaftsinformationen zweimal übertragen werden (siehe 6. Schritt und 9. Schritt in Abschnitt 3.1.2).

In SCANv3 werden diese Informationen direkt vom Master-Device an die betroffenen Nachbarn übertragen. Dadurch wird der Kommunikationsaufwand reduziert. Analog dazu erhält der Zone-Owner seinen neuen Schlüssel direkt vom Master-Device ohne wie in der alten Version den Umweg über den beitretenden Knoten zu nehmen. Diese Maßnahmen ermöglichen dem Master-Device eine bessere Kontrolle über den Join-Vorgang und erleichtern den Umgang mit Fehlern, die durch Paketverluste oder bösartige Knoten verursacht werden. Da die ursprüngliche Protokollversion die Präsenz des Master-Device bis zum Abschluss des Beitrittsvorgangs erfordert, entsteht dem Anwender durch die stärkere Integration des Master-Device in den Join-Vorgang kein Nachteil. Die einfachere Struktur des Protokollablaufs und die Verwendung prägnanter Bezeichner für die einzelnen Protokollnachrichten erleichtert zudem das Verständnis.

4.3 Verbessertes Protokoll

In diesem Abschnitt wird die um die im vorherigen Abschnitt genannten Optimierungen erweiterte Version des Protokolls zum Hinzufügen neuer Knoten vorgestellt. Des Weiteren erfolgt die Spezifikation des Protokolls zum sicheren Auffinden und Einfügen von Diensten. Die bisherige Notation aus Abschnitt 3.1.2 wurde weitgehend beibehalten und lediglich um Bezeichner für die einzelnen Nachrichtentypen ergänzt. Die Bezeichnung $E_K(data)$ ist eine Kurzschreibweise für die Verschlüsselung von $(data|mdc_{data})$ mit dem symmetrischen Schlüssel key_k, wobei $mdc_{data} = hash(data)$. Durch die Konkatenation mit einem MDC wird die Integritätsprüfung der Nachricht ermöglicht.

4.3.1 Hinzufügen neuer Knoten

In Abbildung 4.13 ist die verbesserte Version des Protokollablaufs zum Hinzufügen eines Knoten graphisch dargestellt.

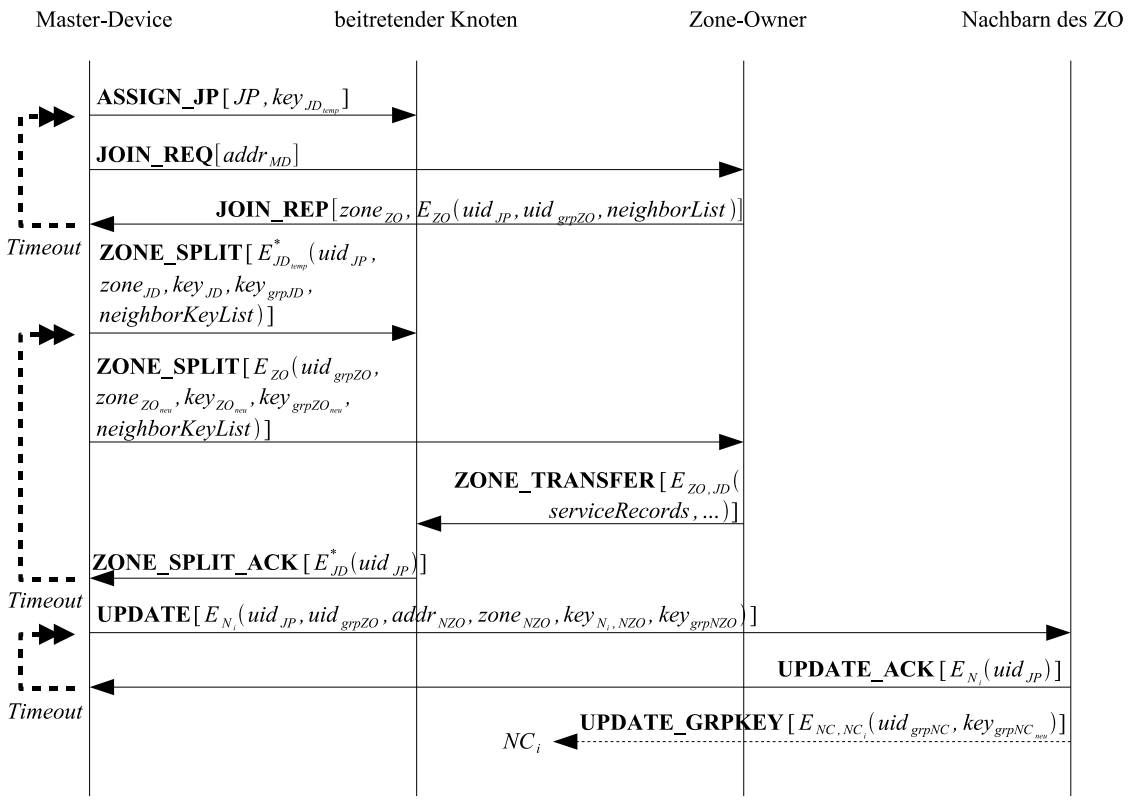

Abbildung 4.13: Hinzufügen eines Knoten in SCANv3

- 1.) PHY|MD→JD : **ASSIGN_JP**[JP, $\text{key}_{\text{JD}_{\text{temp}}}$]

  Das Master-Device sendet über den sicheren physischen Kanal einen zufällig gewählten Join-Point *JP* an den beitretenden Knoten. Durch Verschlüsselung des JP und der Knotenadresse des beitretenden Knotens mit dem symmetrischen Superschlüssel key_{SK} des Master-Device wird ein vorläufiger Schlüssel $\text{key}_{\text{JD}_{\text{temp}}} = \text{E}_{\text{SK}}(hash(\text{JP}|addr_{JD}))$ zur weiteren Kommunikation zwischen Master-Device und dem beitretenden Knoten erzeugt. Das Master-Device ist für die weitere Kommunikation mit dem beitretenden Knoten somit nicht mehr auf die Verwendung des sicheren Kanals angewiesen. Der erforderliche physische Kontakt zwischen Master-Device und beitretendem Knoten kann damit auf ein Minimum reduziert werden.

- 2.) CAN|MD→*JP* : **JOIN_REQ**[addr_{MD}]

  Das Master-Device sendet eine JOIN_REQ Nachricht über das CAN-Overlay in Richtung Join-Point JP. Diese erreicht den für diesen Hashwert zuständigen Zone-Owner *ZO*. Damit der Zone-Owner dem Master-Device ohne Verwendung des Overlays antworten kann, enthält die Nachricht die Netzwerkadresse des Master-Device.

- 3.) MC|ZO→MD : **JOIN_REP**[zone_{ZO}, $\text{E}_{\text{ZO}}(\text{uid}_{\text{JP}}, \text{uid}_{\text{grpZO}}, neighborList)$]

  Der Zone-Owner antwortet mit einer JOIN_REP Nachricht, die er direkt (ohne Verwendung des Overlays) an das Master-Device schickt. Dieses berechnet aus der Netzwerkadresse und den Zonenkoordinaten des Zone-Owners den symmetrischen Schlüssel $\text{key}_{\text{ZO}} = E_{SK}(hash(zone_{ZO}|addr_{ZO}))$ und entschlüsselt

damit den zweiten Nachrichtenteil. Durch die Kenntnis von key_{ZO} hat sich der Zone-Owner authentifiziert und seine angegebenen Zonenkoordinaten bestätigt.

Der Hashwert uid_{JP} des Join-Points JP wird als Nonce zur Verhinderung von Replay-Angriffen verwendet. Um auch für Nachrichten, die vom Master-Device an den Zone-Owner und dessen Nachbarn N_i gesendet werden, einen solchen Schutz zu ermöglichen, ist für diese Kommunikationsrichtung der Hashwert des Gruppenschlüssels des Zone-Owners uid_{grpZO} in der Nachricht enthalten. Die *neighborList* enthält eine Aufstellung aller Overlay-Nachbarn des Zone-Owners in der Form $(\text{addr}_{\text{N}_\text{i}}, \text{zone}_{\text{N}_\text{i}}, \text{key}_{\text{grpN}_\text{i}})$.

Sofern die JOIN_REP Nachricht ausbleibt, fährt das Master-Device nach Ablauf eines Timeouts mit dem ersten Schritt fort.

- 4.) PHY|MD→JD : **ZONE_SPLIT**[$\text{E}^{*}_{\text{JD}_\text{temp}}(\text{uid}_{\text{JP}}, \text{zone}_{\text{JD}}, \text{key}_{\text{JD}}, \text{key}_{\text{grpJD}},$ *neighborKeyList*)]

  Das Master-Device teilt die alte Zone des Zone-Owners und weist dem beitretenden Knoten mit einer ZONE_SPLIT Nachricht seine neue Zone sowie den zugehörigen Schlüssel key_{JD} zu.

  Die *neighborKeyList* enthält für jeden Nachbarn des beitretenden Knotens einen Eintrag und besteht aus Tupeln der Form $(\text{addr}_{\text{N}_\text{i}}, \text{zone}_{\text{N}_\text{i}}, \text{key}_{\text{ZO,N}_\text{i}}, \text{key}_{\text{grpN}_\text{i}})$. Der erste Eintrag enthält die Daten über den Zone-Owner.

  Die Verschlüsselung der Nachricht (*) mit $\text{key}_{\text{JD}_\text{temp}}$ ist optional und kann bei Verwendung eines sicheren Kommunikationskanals entfallen. In SCANv2 erfolgt die Zuweisung von Zone und Schlüssel verteilt im 4., 5., und 8. Schritt. Durch die Zusammenfassung der Nachrichten in SCANv3 wird die Zahl der gesendeten Pakete reduziert und der Protokollablauf übersichtlicher. Zudem kann die Zuweisung einer neuen Zone an den beitretenden Knoten und den Zone-Owner einheitlich mit einer Nachricht gleichen Formats erfolgen.

- 5.) MC|MD→ZO : **ZONE_SPLIT**[$\text{E}_{\text{ZO}}(\text{uid}_{\text{grpZO}}, \text{zone}_{\text{ZO}_\text{neu}}, \text{key}_{\text{ZO}_\text{neu}},$ $\text{key}_{\text{grpZO}_\text{neu}}$, *neighborKeyList*)]

  Analog zum 4. Schritt wird dem Zone-Owner seine neue Zone zugeteilt. Dieser überprüft, ob die enthaltene uid_{grpZO} mit dem Hashwert seines Gruppenschlüssels key_{grpZO} übereinstimmt. Falls dies zutrifft und es sich somit um eine aktuelle Nachricht handelt, aktualisiert er seine Zonenkoordinaten, Schlüssel und Nachbarschaftsinformationen. Der erste Eintrag der *neighborKeyList* enthält die Daten des beitretenden Knotens. Durch die dort gespeicherte Netzwerkadresse des beitretenden Knotens und den gemeinsamen Schlüssel $key_{ZO,JD}$ kann der Zone-Owner sicher Kontakt mit dem neu hinzukommenden Knoten aufnehmen.

- 6.) MC|ZO→JD : **ZONE_TRANSFER**[$\text{E}_{\text{ZO,JD}}(\text{serviceRecords}, ...)$])

  Optional kann der Zone-Owner Einträge aus dem Dienstverzeichnis, die jetzt in der Zone des beitretenden Knoten liegen, an diesen übertragen. Alternativ können die Einträge auch durch eine periodischen Aktualisierung durch den Dienstanbieter erneut eingetragen werden.

- 7.) PHY|JD→MD : **ZONE_SPLIT_ACK**[$E^{*}_{JD}(uid_{JP})$]

  Der beitretende Knoten bestätigt die erfolgreiche Teilung der Zone mit einer ZONE_SPLIT_ACK Nachricht beim Master-Device. Sofern die Bestätigung ausbleibt, fährt das Master-Device nach Ablauf eines Timeouts mit dem 4. Schritt fort.

  Falls ein sicherer Kommunikationskanal verwendet wird, kann die Verschlüsselung (*) in diesem Schritt ebenfalls entfallen.

- 8.) MC|MD→N_i : **UPDATE**[$E_{N_i}(uid_{JP}, uid_{grpZO}, addr_{NZO}, zone_{NZO}, key_{N_i,NZO}, key_{grpNZO})$]

  Das Master-Device informiert alle Nachbarn N_i über den neuen Zone-Owner NZO (= JD oder ZO) und verteilt die symmetrischen Schlüssel zur Kommunikation mit diesem. Zur Verhinderung von Replay-Angriffen überprüfen die Nachbarn N_i, ob der enthaltene uid_{grpZO} mit dem Hashwert des lokal gespeicherten Gruppenschlüssels key_{grpZO} übereinstimmt.

- 9.) MC|N_i→MD : **UPDATE_ACK**[$E_{N_i}(uid_{JP})$]

  Jeder Nachbar bestätigt den korrekten Empfang der Update-Nachricht mit einer UPDATE_ACK Nachricht an das Master-Device. Falls die Bestätigung eines Nachbarn ausbleibt, wiederholt das Master-Device nach Ablauf eines Timeouts die Update-Nachricht aus dem 8. Schritt.

- 10.) MC|NC→NC_i : **UPDATE_GRPKEY**[$E_{NC,NC_i}(uid_{grpNC}, key_{grpNC_{neu}})$]

  Einer der alten Nachbarn des Zone-Owners hat diesen jetzt nicht mehr als Nachbarn. Dieser Knoten NC muss seinen Gruppenschlüssel ändern, da der alte Zone-Owner noch im Besitz dieses Schlüssel ist, aber der Gruppe nicht mehr angehört. In Abschnitt 3.2.2 wird dies anhand eines Beispiels verdeutlicht.

  Der NC wählt daher zufällig einen neuen Gruppenschlüssel $key_{grpNC_{neu}}$. Diesen Gruppenschlüssel teilt er allen aktuellen Nachbarn NC_i durch Versand einer UPDATE_GRPKEY Nachricht mit.

4.3.2 Austausch eines Sitzungsschlüssels

In Abschnitt 4.1 wurden die Verfahren SPX und MPX zum Austausch eines Sitzungsschlüssels vorgestellt. Das folgende Protokoll wird in SCANv3 verwendet um mit SPX oder MPX einen Sitzungsschlüssel zwischen dem anfragenden Knoten S und dem Zielknoten ZO, der den angefragten Hashwert *serviceHash* verwaltet, zu etablieren.

- 1.) CAN|S→*serviceHash* : **KEY_XCHG_REQ**[uid, $addr_S$, r, $secret_i$]

  Der anfragende Knoten wählt eine zufällige Kennung *uid*, die eine Zuordnung weiterer Nachrichten dieses Schlüsselaustauschs erleichtert, sowie r verschiedene Geheimnisse $secret_i$. Diese schickt er mit Hilfe von r KEY_XCHG_REQ Nachrichten über disjunkte Pfade des CAN-Overlays in Richtung des Zielpunkts *serviceHash*, welcher in der Zone des Zielknotens liegt. Da SPX nur ein Spezialfall (Zahl der verwendeten Pfade $r = 1$) des Verfahrens MPX darstellt, können für beide Verfahren die gleichen Protokollnachrichten verwendet werden.

- 2.) MC|ZO→S : **KEY_XCHG_REP**[serviceHash, $E_{key_{session}}$(uid)]

  Der Zielknoten speichert alle eingehenden KEY_XCHG_REQ Nachrichten und berechnet, sobald alle r KEY_XCHG_REQ Nachrichten (mit gleicher *uid*) eingetroffen sind, den Sitzungsschlüssel $key_{session} = secret_1 \oplus secret_2 \oplus \ldots \oplus secret_r$.

  Um die Gültigkeit des Schlüssels mittels eines Challenge-Response-Verfahrens zu prüfen, schickt der Zielknoten eine KEY_XCHG_REP Nachricht an den anfragenden Knoten, die mit dem neu erzeugten Sitzungsschlüssel $key_{session}$ verschlüsselt wird. Falls der anfragende Knoten keine gültige KEY_XCHG_REP Nachricht erhält, wurden Teile des Schlüssels durch einen Angreifer manipuliert. In diesem Fall wiederholt der anfragende Knoten den ersten Schritt unter Verwendung alternativer Pfade zum Zielknoten.

4.3.3 Auffinden von Diensten

In SCANv3 wird folgendes Protokoll verwendet um einen gesuchten Dienst mit dem Hashwert *serviceHash* sicher aufzufinden.

- 1.) MC|S→ZO : **LOOKUP_REQ**[$E_{key_{session}}$(uid, serviceHash, parms)]

  Um einen Dienst aufzufinden, muss zuerst ein gemeinsamer Sitzungsschlüssel zwischen dem suchenden Knoten S und dem Knoten ZO, der das Verzeichnis für diesen Dienst verwaltet, ausgetauscht werden (siehe Abschnitt 4.3.2). Sobald der Sitzungsschlüssel etabliert ist, startet der suchende Knoten die Anfrage mit Hilfe einer LOOKUP_REQ Nachricht, die er direkt an Knoten ZO schickt. Die Nachricht enthält den Hashwert des angefragten Dienstnamens, eine *uid* zur Verhinderung von Replay-Angriffen, sowie weitere Parameter mit denen der Dienst genauer spezifiziert werden kann.

- 2.) MC|ZO→S : **LOOKUP_REP**[serviceHash, $E_{key_{session}}$(uid, serviceRecords]

  Der Knoten ZO beantwortet die Anfrage mit einer LOOKUP_REP Nachricht, die alle Dienstbeschreibungen enthält, die den geforderten Parametern entsprechen.

  Um die Wahrscheinlichkeit für einen erfolgreichen Lookup zu erhöhen, kann der suchende Knoten S den LOOKUP_REQ an mehrere Replikate schicken. In diesem Fall muss mit jedem beteiligten Zielknoten ein eigener Sitzungsschlüssel ausgetauscht werden. Falls einzelne Zielknoten bösartig sind und manipulierte Datensätze verbreiten, kann dies durch Vergleich der erhaltenen Datensätze erkannt und korrigiert werden.

4.3.4 Einfügen von Diensten

Das folgende Protokoll ermöglicht einem Knoten in SCANv3 das sichere Einfügen eines von ihm angebotenen Dienstes.

- 1.) MC|S→ZO : **INSERT_REQ**[serviceHash, $E_{key_{session}}$(uid, serviceRecord)]

  Analog zur Auffindung von Diensten muss zuerst der gemeinsame Sitzungsschlüssel zwischen dem eintragenden Knoten S und dem Zielknoten ZO ausgetauscht werden. Danach erfolgt das eigentliche Eintragen des Dienstes durch

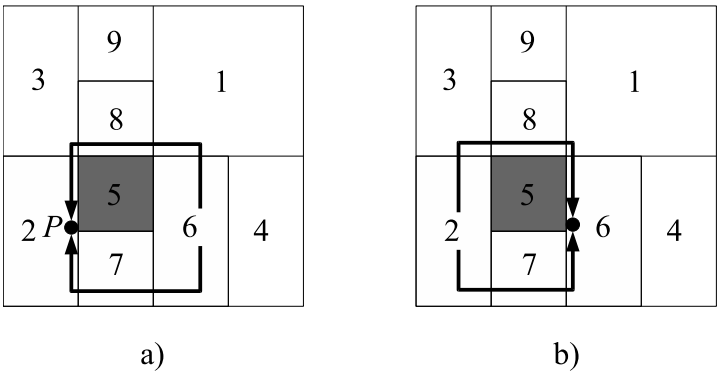

Abbildung 4.14: Schlüsselaustausch nach Ausfall von Knoten 5

direkten Versand einer INSERT_REQ Nachricht an den Zielknoten. Dieser Vorgang wird für alle Replikate wiederholt.

Der Diensteintrag besitzt eine begrenzte Lebenszeit und wird nach deren Ablauf automatisch aus dem Dienstverzeichnis gelöscht. Bietet der Knoten einen Dienst über eine längere Zeitspanne an, muss der Eintrag somit periodisch erneuert werden. Um den Kommunikationsaufwand zu reduzieren, kann in diesem Fall der Sitzungsschlüssel zwischengespeichert werden. Ein erneuter Schlüsselaustausch kann somit entfallen.

- 2.) MC|ZO→S : **INSERT_REP**[serviceHash, $\mathrm{E}_{\mathrm{key}_{\mathrm{session}}}$(uid)]

  Der Zielknoten quittiert den korrekten Empfang mit einer INSERT_REP Nachricht, welche die Kennung *uid* aus der INSERT_REQ Nachricht enthält.

  Für den Fall, dass der Zielknoten aufgrund einer Zonenteilung für den angegebenen *serviceHash* nicht mehr zuständig ist, antwortet er dem eintragenden Knoten mit NOT_RESPONSIBLE, worauf dieser einen neuen Schlüsselaustausch initiiert.

4.4 Behandlung von Knotenausfällen

In Abschnitt 3.3 wurde dargestellt, warum das in SCANv2 vorgeschlagene Verfahren nicht zur sicheren Behandlung von Knotenausfällen geeignet scheint. Im Rahmen der Diplomarbeit konnte keine abschließende Lösung für dieses Problem gefunden werden. Ein möglicher Lösungsansatz soll aber im Folgenden skizziert werden.

Ein wesentlicher Kritikpunkt an dem bisherigen Verfahren ist der Einsatz eines Wahlverfahrens um den Nachfolgerknoten für die ausgefallene Zone zu ermitteln. Dieses Verfahren erfordert das Versenden vieler Nachrichten und in der Regel einen Mechanismus um die Teilnehmer untereinander zu authentifizieren. Auf ein Wahlverfahren kann verzichtet werden, wenn im Voraus eindeutig festgelegt ist, von welchem Knoten eine bestimmte Zone bei einem Ausfall übernommen wird. Eine mögliche Regel könnte beispielsweise lauten:

Gegeben sie eine Zone Z mit $Z = [u_1; v_1] \times [u_2; v_2] \times \ldots \times [u_n; v_n]$ sowie eine Menge von Punkten P_i am Rand dieser Zone mit $P_i = (u_1, u_2, \ldots, u_{i-1}, u_i - 1, u_{i+1}, \ldots, u_n)$. Weiter sei $P \in \{P_i | \forall i, j \in N, 1 \leq j < i : P_i \in Z, P_j \notin Z\}$ der Punkt dieser Menge,

der als erster nicht in der Zone Z selbst liegt. Bei einem Ausfall des Knotens mit der Zone Z wird diese immer von dem Knoten übernommen, in dessen Zuständigkeitsbereich der Punkt P fällt.

Der die ausgefallene Zone übernehmende Knoten besitzt mit seinen neuen Nachbarn bisher keine gemeinsamen symmetrischen Schlüssel mit dem er Overlay-Nachrichten verschlüsseln und sich gegenüber seinen Nachbarn authentisieren könnte. Die Schlüssel müssen dynamisch im Bedarfsfall generiert werden, da ein vorheriges Verteilen solcher Schlüssel durch das Master-Device aus den in Abschnitt 3.3 genannten Gründen nicht möglich ist. Mit Hilfe des Schlüsselaustauschverfahrens MPX kann durch die Verwendung disjunkter Pfade relativ sicher ein gemeinsamer Schlüssel zwischen den Knoten etabliert werden.

Sobald ein Knoten den Ausfall einer Nachbarzone bemerkt, kann er nach obiger Vorschrift den Punkt P berechnen, welcher in der Zone des Nachfolgerknotens liegt. An diesen Punkt schickt er verteilt über das Overlay seine Schlüsselteile. Um unberechtigterweise die ausgefallene Zone in seinen Besitz bringen zu können, muss ein Angreifer somit alle Schlüsselteile abfangen. Die Sicherheit des Verfahrens hängt also maßgeblich von der Anzahl der noch zur Verfügung stehenden disjunkten Pfade ab. Durch den Knotenausfall kann mindestens einer der Pfade nicht mehr für den Schlüsselaustausch verwendet werden. Falls nicht genügend disjunkte Pfade zur Verfügung stehen, können gegebenenfalls zusätzliche Pfade durch Weitergabe der Overlay-Nachrichten an Nachbarn entgegen der direkten Routingrichtung benutzt werden.

In Abbildung 4.14 ist ein Beispiel für dieses Verfahren zu sehen. Knoten 6 stellt durch das Ausbleiben periodischer Update-Nachrichten den Ausfall von Knoten 5 fest. Daraufhin berechnet Knoten 6 anhand des vorher festgelegten Verfahrens den Punkt P, welcher den Knoten bestimmt, der die ausgefallene Zone übernimmt. An diesen sendet er über zwei disjunkte Pfade verteilt einen neuen symmetrischen Schlüssel. Mit diesem kann sich der die ausgefallene Zone verwaltende Knoten 2 gegenüber Knoten 6 authentifizieren. Knoten 2 sendet nun ebenfalls mit Hilfe von MPX einen symmetrischen Schlüssel an einen Punkt, der in der Zone von Knoten 6 und an die ausgefallene Zone von Knoten 5 angrenzt. Damit kann er überprüfen, ob es sich bei Knoten 6 um einen ehemaligen Nachbarn von Knoten 5 handelt und er damit als neuer Overlay-Nachbar von Knoten 2 akzeptiert werden soll.

Das vorgestellte Verfahren stellt einen viel versprechenden Ansatz für die sichere Behandlung von Knotenausfällen dar. Allerdings besteht weiterhin das Problem, dass durch einen DoS-Angriff die periodischen Update-Nachrichten unterdrückt werden können und damit die Knotenausfallbehandlung angestoßen wird. Des Weiteren muss das Verfahren zur Weitergabe von Nachrichten entgegen der direkten Routingrichtung zur Gewinnung zusätzlicher Pfade noch genauer spezifiziert werden. Schließlich muss durch weitere Arbeiten geklärt werden, wie groß die Anzahl der verwendbaren Pfade ist und ob ein mit diesem Verfahren gewonnener Schlüssel zur sicheren Kommunikation benutzt werden kann.

5. Implementierung im Simulator

In diesem Kapitel werden der Aufbau und die Besonderheiten der im Rahmen dieser Arbeit angefertigten Implementierung von SCANv3 in einer Simulationsumgebung beschrieben.

5.1 Der Simulator GloMoSim

Zur Simulation des Dienstverzeichnisses SCANv3 wird der Netzwerksimulator GloMoSim [ZeBG98] als Grundlage verwendet und entsprechend erweitert. GloMoSim ist ein diskreter, ereignisgesteuerter Simulator für drahtlose Netze mit mobilen Knoten. Beim ereignisgesteuerten Simulationsmodell erfolgen Zustandsänderungen nur wenn Ereignisse eintreten, wie beispielsweise die Bewegung eines Knotens oder der Empfang einer Nachricht. Der Simulator verwendet die Simulationssprache Parsec [BMTC+98], die an die Sprache C angelehnt ist und sowohl sequentielle als auch parallele Simulationen unterstützt.

Aufgrund seiner modularen Struktur, die sich am ISO/OSI-Schichtenmodell orientiert, kann GloMoSim auf einfache Weise um zusätzliche Funktionalität erweitert werden. In Tabelle 5.1 sind die in GloMoSim zur Verfügung stehenden Modelle und Protokolle der einzelnen Schichten aufgeführt. Bisher unterstützt GloMoSim keine Protokolle speziell für Sensornetze sondern nur Protokolle aus dem Bereich der drahtlosen Ad-hoc-Netze. Sobald geeignete Protokolle für Sensornetze zur Verfügung stehen, könne diese jedoch aufgrund der modularen Struktur einfach ausgetauscht werden.

Für jeden simulierten Knoten existiert in GloMoSim eine eigene Datenstruktur `GlomoNode`. Diese enthält unter anderem die Adresse des Knotens sowie Zeiger auf protokolleigene Datenstrukturen. In diesen Datenstrukturen wird für jedes Protokoll der Zustand des Knotens gespeichert. Die komplette Kommunikation zwischen den einzelnen Schichten erfolgt durch das Versenden von Nachrichten mit der Datenstruktur `Message`. Auf die gleiche Weise werden einer Protokollimplementierung Ereignisse mitgeteilt wie zum Beispiel der Ablauf eines Timers.

Schicht	Modelle/Protokolle
Knotenmobilität	Random-Waypoint, Random-Drunken
Funkausbreitung	Two-Ray, Free-Space
Funkmodell	Noise-Accumulating
Paketempfangsmodell	SNR-begrenzt, BER-basiert mit BPSK/QPSK-Modulation
Sicherungsschicht	CSMA, IEEE 802.11, MACA
Vermittlungsschicht	IP mit AODV, Bellman-Ford, DSR, Fisheye, LAR1, ODMRP, WRP
Transportschicht	TCP, UDP
Anwendungsschicht	CBR, FTP, HTTP, Telnet

Tabelle 5.1: Vorhandene Protokolle in GloMoSim

5.2 Integration in GloMoSim

Um die Implementierung von SCANv3 in den bestehenden Simulator zu integrieren, mussten einige Dateien von GloMoSim modifiziert werden. In Tabelle 5.2 sind die modifizierten Dateien aufgelistet. Eine kurze Einführung zur Erweiterung von GloMoSim um zusätzliche Protokolle findet sich in [Nuev03].

Da das Dienstverzeichnis SCANv3 ein Overlay-Netz darstellt, wurde es als zusätzliches Protokoll auf Anwendungsschicht in GloMoSim integriert. Das Dienstverzeichnis kann somit auf die bereits vorhandenen Transportprotokolle von GloMoSim aufsetzen. Durch die modulare Architektur kann die Implementierung somit auf einfache Weise im Zusammenspiel mit verschiedenen Routingprotokollen evaluiert werden. Die Implementierung des Dienstverzeichnisses befindet sich in den Dateien `application/scan.pc` und `application/scan.h`. Die Schnittstelle zwischen Simulatorkern und dem einzelnen Anwendungsprotokoll wird durch drei Funktionen realisiert, über die jedes Anwendungsprotokoll mindestens verfügen muss:

1. **Initialisierungsfunktion.** Diese muss Speicher für protokolleigene Datenstrukturen reservieren und initialisieren.
2. **Finalisierungsfunktion.** Diese gibt am Ende der Simulation die gesammelten Statistikdaten des Protokolls aus.
3. **Event-Handler-Funktion.** Diese führt Protokollaktionen aus, sobald ein Ereignis eingetreten ist.

In der Implementierung von SCANv3 sind dies die Funktionen `AppScanInit()`, `AppScanFinalize()` sowie `AppLayerScan()`, die zur Registrierung des neuen Protokolls in `application/application.pc` hinzugefügt wurden. Zusätzlich zum eigentlichen Dienstverzeichnis musste noch ein Verfahren zur statischen Berechnung von Routen im Underlay implementiert werden (siehe Abschnitt 5.3.4). Die Einbindung dieses „Routingprotokolls“ erfolgte in der Datei `network/network.pc` analog zu der Einbindung des Anwendungsprotokolls.

5.3 Besonderheiten der Implementierung

Im Folgenden werden kurz einige Besonderheiten der Implementierung beschrieben.

Datei	Grund der Modifikation
`application/scan.pc`	*neu:* Anwendungsprotokoll SCAN
`application/application.pc`	Schnittstelle zu SCAN hinzugefügt
`include/application.h`	SCAN zur Liste der Anwendungsgprotokolle hinzugefügt
`network/precalc.pc`	*neu:* Routingprotokoll PRECALC
`network/network.pc`	Schnittstelle zu PRECALC hinzugefügt
`include/network.h`	PRECALC zur Liste der Routingprotokolle hinzugefügt
`radio/radio_accnoise.pc`	Simulation von Knotenausfällen hinzugefügt
`radio/radio_nonoise.pc`	Simulation von Knotenausfällen hinzugefügt
`include/api.h`	Datenstruktur `GlomoNode` für Simulation von Knotenausfällen erweitert

Tabelle 5.2: Modifizierte Dateien

5.3.1 Annahmen und Vereinfachungen

Anhand der Implementierung im Simulator sollen verschiedene Aspekte von SCANv3 untersucht werden. Zunächst soll geklärt werden, ob das in Abschnitt 4.3 spezifizierte Protokoll dazu geeignet ist, eine korrekte Overlay-Struktur aufzubauen. Des Weiteren soll gezeigt werden, dass die vorgeschlagenen Verfahren zum sicheren Einfügen und Auffinden von Diensten geeignet sind in Netzen mit bösartigen Knoten mit hoher Wahrscheinlichkeit korrekte Lookup-Ergebnisse zurückzuliefern. Schließlich soll der Kommunikationsaufwand für verschiedene Operationen abhängig vom gewählten Schlüsselaustauschverfahren gemessen werden.

Damit die Komplexität der Implementierung möglichst gering gehalten werden kann, wird die Implementierung durch einige Annahmen vereinfacht. Um Rechenaufwand zu sparen wird auf die Verwendung von kryptographischen Verfahren und Hashfunktionen vollständig verzichtet. Um eine exakte Messung des Kommunikationsaufwands zu ermöglichen, enthalten die übertragenen Nachrichten dennoch alle Schlüssel und MDCs. Lediglich die eigentliche Verschlüsselung und Integritätsprüfung der Nachrichten entfällt. Die Hashwerte möglicher Dienstnamen werden zu Beginn der Simulation zufällig erzeugt. Im weiteren Simulationsverlauf wird dann auf diese „vorausberechneten“ Werte zurückgegriffen. Um die Sicherheit des Schlüsselaustauschverfahrens bewerten zu können, wird ein einfacher Insider-Angreifer simuliert (siehe Abschnitt 5.3.3).

5.3.2 Simulation des Master-Device

In der Implementierung wurde die Funktionalität des Master-Device in die Sensorknoten integriert. Somit wird die aufwändige Simulation eines heterogenen Netzes sowie die Simulation des physischen Kanals zwischen Master-Device und Sensorknoten vermieden. Damit im Quellcode ersichtlich wird, ob eine Funktion durch das Master-Device oder den Sensorknoten erbracht wird, enthalten alle Funktionen, die das Master-Device simulieren, das Schlüsselwort `Master` im Funktionsnamen. Der Zustand des Master-Device wird während eines Knotenbeitritts in der globalen Struktur `ScanMaster` gespeichert. Um sicherzustellen, dass ein Sensorknoten

die Fähigkeiten des Master-Device nur einmalig beim Beitritt zum Overlay nutzen kann und eine gleichzeitige Mehrfachbenutzung des Master-Device verhindern wird, enthält diese Struktur den Zeiger `joiningScanPtr`, die auf den Sensorknoten verweist, der momentan dem Netz hinzugefügt und physischen Kontakt mit dem Master-Device hat. Sobald der Knoten erfolgreich zum Overlay hinzugefügt wurde, werden durch die Funktion `MasterJoinFinished()` die in der Struktur `ScanMaster` gespeicherten Zustände des Master-Device gelöscht. Die Zeitspanne zwischen zwei Knotenbeitritten ist das Master-Device somit zustandslos.

5.3.3 Simulation bösartiger Knoten

Ein Maß für die Praxistauglichkeit eines Dienstverzeichnisses ist die Gesamtwahrscheinlichkeit für das erfolgreiche Auffinden eines Dienstes. Diese hängt von den Anforderungen ab, die eine Anwendung an den Dienstauffindungsmechanismus stellt: Für einige Anwendungen ist es ausreichend, wenn mindestens ein Knoten gefunden wird, der den gesuchten Dienst anbietet, während für andere Anwendungen ein Lookup nur erfolgreich ist, wenn alle Knoten, die diesen Dienst momentan anbieten, auch gefunden werden. Allgemein fordert eine Anwendung also, dass mindestens eine feste Anzahl x bzw. mindestens y Prozent der angeforderten Datensätze für einen bestimmten Dienst zurückgeliefert werden.

Um die Eignung von SCANv3 für Netze mit einem bestimmten Anteil von bösartigen Knoten beurteilen zu können, enthält die Implementierung die Simulation eines Angreifers, der Knoten manipulieren kann und dadurch zum Insider-Angreifer wird. Um die Komplexität der Implementierung zu reduzieren, wird nur ein „worst-case“ Angreiferverhalten simuliert, welches maximalen Schaden hervorruft. Der Angreifer zeigt daher gegenüber seinen Overlaynachbarn zunächst kein auffälliges Verhalten und hält sich bis auf die im Folgenden angegebenen Ausnahmen an das Protokoll. Somit wird verhindert, dass der Angreifer vorzeitig von der Kommunikation im Overlay ausgeschlossen wird. Ziel des simulierten Angriffs ist es, das Auffinden eines Dienstes zu verhindern und, falls genug kooperierende Angreifer zur Verfügung stehen, manipulierte Datensätze zurückzuliefern.

Ein einzelner Angreifer kann in der Schlüsselaustauschphase das verteilte Geheimnis einer `KEY_XCHG_REQ` Nachricht (siehe Abschnitt 4.3.2) verändern, sofern er sich auf einem der Overlay-Pfade zum Zielknoten befindet, und dadurch den Schlüssel unbrauchbar machen. Dies verzögert den erfolgreichen Schlüsselaufbau und führt zu einem erhöhten Kommunikationsaufwand, da erneut `KEY_XCHG_REQ` Nachrichten (über alternative Pfade) versendet werden müssen. Die Manipulation des Schlüsselmaterials wird in der Implementierung durch Setzen eines Flags in der weitergeleiteten Overlaynachricht simuliert. Da in der Realität diese Manipulation erst durch ein Scheitern des Challenge-Response-Verfahrens[1] bei demjenigen Knoten erkannt werden kann, der den Schlüsselaustausch initiiert hat, bestätigt der Zielknoten in der Simulation auch den Empfang einer als manipuliert markierten Nachricht mit einer `KEY_XCHG_REP` Nachricht. Der Empfänger dieser Nachricht erkennt an der Markierung, dass der Schlüsselaustausch gescheitert ist und startet einen neuen Versuch.

Sofern ein Angreifer die Mehrzahl der Pfade zum Zielknoten kontrolliert oder falls der Zielknoten selbst bösartig ist, kann der Angreifer in den Besitz eines gültigen Sit-

[1] Rücksendung der mit dem Sitzungsschlüssel verschlüsselten *uid* durch eine `KEY_XCHG_REP` Nachricht

zungsschlüssels gelangen. Mit diesem kann er jede Anfrage, die mit diesem Schlüssel verschlüsselt wird, mit beliebig generierten Datensätzen beantworten. Ein erfolgreiches Auffinden von Diensten, ist mit diesem Schlüssel also nicht mehr möglich. Dieser Fall wird ebenfalls durch eine entsprechende Markierung des `KEY_XCHG_REP` Pakets simuliert.

5.3.4 Statisches Routingprotokoll

Da die vorhandenen Routingprotokolle von GloMoSim nicht zur Simulation großer Sensornetze geeignet sind (siehe Abschnitt 6.1), wurde zusätzlich ein statisches „Routingprotokoll“ unter dem Namen PRECALC implementiert. Die Implementierung befindet sich in den Dateien `network/precalc.pc` und `network/precalc.h` Die Verwendung von PRECALC ermöglicht mit geringem Rechenaufwand die Simulation eines Multi-Hop-Underlay mit Tausenden von Knoten. Solange keine geeignete Implementierung eines Routingprotokolle für Sensornetze dieser Größenordnung zur Verfügung steht, stellt die Verwendung von PRECALC eine einfache Möglichkeit dar um das Dienstverzeichnis SCANv3 unabhängig von Unzulänglichkeiten vorhandener Routingprotokolls evaluiert zu können. Da die Wahl des Routingprotokolls keinen direkten Einfluss auf den Kommunikationsaufwand und die Sicherheit von SCANv3 hat, werden durch diese Vereinfachung die Simulationsergebnisse nicht verfälscht.

Durch die Funktion `RoutingPrecalcCreateNextHopTable()` wird einmalig zu Beginn der Simulation der Topologiegraph des Underlays anhand der aktuellen Position und der maximalen Sendereichweite der Sensorknoten aufgestellt. Anhand des Topologiegraphen werden mit Hilfe des Floyd-Algorithmus die kürzesten Pfade zwischen allen Knotenpaaren ermittelt und in die Routingtabelle `nextHop` eingetragen. Für jedes Paket, das die Vermittlungsschicht an die Sicherungsschicht weiterreicht, wird durch Aufruf der Funktion `RoutingPrecalcRouterFunction()` anhand der statischen Routingtabelle der nächste Knoten auf dem Pfad ermittelt und zurückgeliefert.

Da die Erstellung des Topologiegraphen und die Berechnung der Routingtabelle aus Effizienzgründen nur einmalig zu Beginn der Simulation erfolgt, muss bei der Verwendung dieses Verfahrens auf Mobilität der Knoten verzichtet werden. Des Weiteren muss darauf geachtet werden, dass bei einer Änderung von Parametern des Funkmodells die Konstante `PRECALC_RADIO_RANGE` an die resultierende Sendereichweite der Knoten angepasst wird. Die Berechnung der Sendereichweite aus den Parametern des Funkmodells kann durch das separate Programm `./bin/radio_range` erfolgen.

5.4 Zeitlicher Ablauf der Simulation

Jeder Knoten der dem SCANv3-Overlay hinzugefügt wird, führt die gleiche Folge von Operationen durch. In Abbildung 5.1 ist der zeitliche Ablauf der simulierten Dienstverzeichnis-Operationen dargestellt. Der Ablauf beginnt mit dem Hinzufügen des Knotens zum Overlay zu dem im `APP-CONFIG-FILE` (siehe Abschnitt 5.5) festgelegten Startzeitpunkt. Der Ablauf der Join-Operation entspricht dem in Abschnitt 4.3.1 spezifizierten Protokoll. Sobald die Join-Operation erfolgreich abgeschlossen ist, führt der Knoten eine (in der Konfigurationsdatei) festgelegte Anzahl von Lookup-Operationen für verschiedene Dienste durch. Hierzu etabliert der Knoten einen Sitzungsschlüssel durch das Versenden einer `KEY_XCHG_REQ` Nachricht

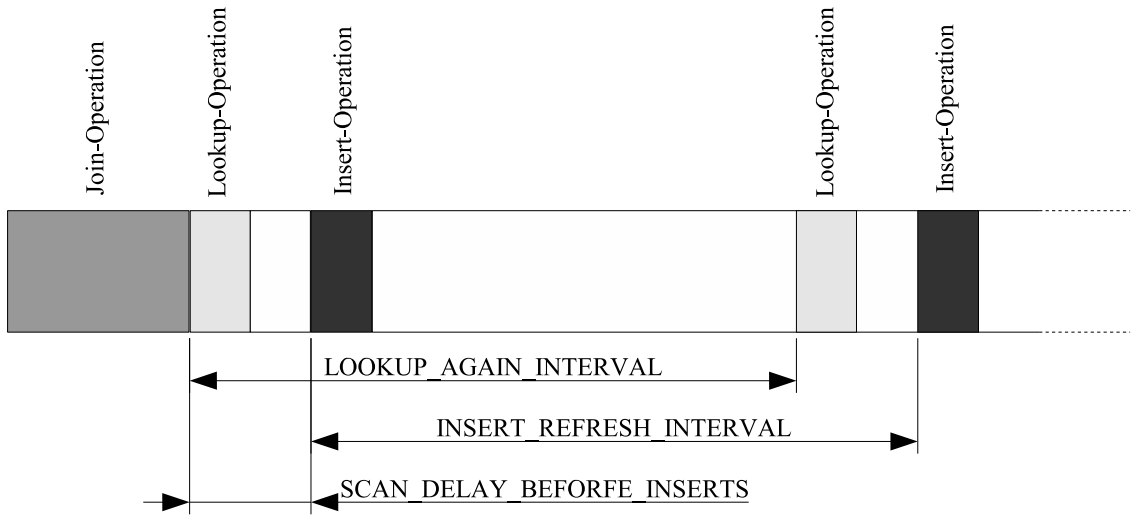

Abbildung 5.1: Zeitlicher Ablauf der Simulation eines Overlay-Knotens

(siehe Abschnitt 4.3.2). Sobald der Schlüsselaustausch erfolgreich abgeschlossen ist, fragt der Knoten mit Hilfe einer `LOOKUP_REQ` den eigentlichen Eintrag im Dienstverzeichnis ab. Der Knoten beginnt nach Ablauf der in `application/scan.h` definierten Zeitdauer `SCAN_DELAY_BEFORE_INSERTS` mit dem Einfügen seiner Dienste. Dazu wird ebenfalls zuerst ein Schlüsselaustausch durchgeführt und dann wie in Abschnitt 4.3.4 beschrieben durch eine `INSERT_REQ` Nachricht die eigentliche Insert-Operation durchgeführt. Um den Diensteintrag periodisch aufzufrischen, wird die Insert-Operation jeweils nach Ablauf des `INSERT_REFRESH_INTERVAL` wiederholt. Sofern möglich (siehe Abschnitt 4.3.4) wird dazu der alte Sitzungsschlüssel wiederverwendet, sodass in vielen Fällen ein erneuter Schlüsselaustausch entfällt. Auf die gleiche Weise wird nach jedem Ablauf des `LOOKUP_AGAIN_INTERVAL` die festgelegte Anzahl von Diensten erneut aufgesucht, um auch Dienste von in der Zwischenzeit neu hinzugekommenen Knoten berücksichtigen zu können.

5.5 Ein- und Ausgabedateien des Simulators

GloMoSim verwendet mehrere Eingabedateien zur Konfiguration des Simulationsszenarios. Der Name der Hauptkonfigurationsdatei kann frei gewählt werden und muss beim Start von GloMoSim auf der Kommandozeile angegeben werden. Eine Beispielkonfiguration befindet sich in `bin/glomo.config`. Die Namen der weiteren Konfigurationsdateien werden in dieser Hauptkonfigurationsdatei festgelegt. Für diese Arbeit ist nur die durch die Option `APP-CONFIG-FILE` festgelegte Datei von weiterem Interesse. Diese wird zur Parametrisierung der Anwendungsprotokolle verwendet. Der Syntax für die Konfiguration eines Knotens für SCANv3 lautet:

```
SCAN <nodeaddr> <bootstrap> <dimensions> <start time> <lookups>
     <inserts> <security mask> <malicious time> <failure time>
```

Die Bedeutung der einzelnen Parameter wird in Tabelle 5.3 erklärt. Ein solcher Konfigurationseintrag wird für jeden Knoten benötigt, der dem Overlay beitreten soll.

Parameter	Bedeutung
`nodeaddr`	Adresse des Knoten, der dem Overlay beitreten soll
`bootstrap`	Adresse eines beliebigen Knotens, der bereits Teil des Overlays ist
`dimensions`	Dimensionalität des CAN-Overlays
`start time`	Zeitpunkt zu dem Knoten dem Overlay beitritt
`lookups`	Anzahl der Dienste, die der Knoten nach dem Beitritt aufsucht
`inserts`	Anzahl der Dienste, die der Knoten nach dem Beitritt einfügt
`security mask`	Bitmaske, die das Sicherheitsniveau festlegt (siehe `application/scan.h`)
`malicious time`	Zeitpunkt ab dem der Knoten bösartig wird
`failure time`	Zeitpunkt zu dem der Knoten ausfällt

Tabelle 5.3: Parameter für den SCAN-Konfigurationseintrag in `APP-CONFIG-FILE`

Das Programm `./bin/create-scan-traffic` ermöglicht die einfache Erzeugung einer solchen Konfigurationsdatei für eine große Anzahl von Knoten.

Die Ausgabe der Statistikdaten erfolgt in die Datei `EXPERIMENT-NAME.stat`, wobei `EXPERIMENT-NAME` für einen Parameter aus der Hauptkonfigurationsdatei steht. Die Datei enthält für jeden simulierten Knoten verschiedene Statistikdaten der einzelnen Protokollschichten. Für SCANv3 werden unter anderem der Anteil erfolgreicher Lookup-Operationen, der benötigte Kommunikationsaufwand im Overlay und Underlay sowie die durchschnittliche Overlay-Pfadlänge ausgegeben. Um eine genaue Analyse der erzielten Lookup-Ergebnisse zu ermöglichen, wird zusätzlich eine Datei `EXPERIMENT-NAME.scan` mit detaillierten Ergebnissen für jede durchgeführte Lookup-Operation erstellt.

6. Evaluierung der Implementierung

In diesem Kapitel werden zunächst die zur Evaluierung von SCANv3 durchgeführten Simulationen beschrieben. Im zweiten Teil werden die erhaltenen Simulationsergebnisse analysiert und bewertet.

6.1 Simulation des Sensornetz-Underlays

Zum Zeitpunkt der Implementierung stand kein ausgereifter Simulator zur Verfügung, der für die Simulation von drahtlosen Sensornetzen geeignet war. Für die Evaluierung des Dienstverzeichnisses SCANv3 im Simulator GloMoSim (siehe Abschnitt 5.1) werden daher zur Modellierung des Sensornetz-Underlays Protokolle aus dem verwandten Bereich der drahtlosen Ad-hoc-Netze verwendet.

Die für die Simulation des Underlays gewählten Parameter der Bitübertragungsschicht ergeben eine Datenübertragungsrate von 250kbit/s bei einer maximalen Sendereichweite von 158m. Dies entspricht den Fähigkeiten eines relativ leistungsfähigen Sensorknotens wie beispielsweise einem aktuellen Mica-Mote [HiCu02]. Auf der Sicherungsschicht kommt das MAC-Protokoll des IEEE 802.11-Standard zum Einsatz. Für den Transport und die Vermittlung von Nachrichten im Underlay werden die Internet-Protokolle IP und UDP verwendet. Um drahtlose Sensornetze zu simulieren, wird in viele Veröffentlichungen (beispielsweise [HSLA03] und [BNWA+03]) GloMoSim mit ähnlichen Parametern verwendet.

Ursprünglich sollten die in GloMoSim vorhandenen Routingprotokolle für Ad-Hoc-Netze (AODV [PeRo99], DSR [JoMa96] und OLSR [CJLM+01]) zur Wegewahl im Underlay eingesetzt werden. Anhand mehrerer Simulationen wurde jedoch festgestellt, dass diese Routingprotokolle nicht für Sensornetze mit 1000 Knoten und mehr geeignet sind. Das proaktive[1] Routingprotokoll OLSR benötigt allein für die Über-

[1] *Proaktive Routingprotokolle* verwenden periodische Signalisierungsnachrichten um kontinuierlich die Routingtabelle zu aktualisieren. Im Gegensatz dazu wird die Suche nach einer Route durch *reaktive Routingprotokolle* erst dann initiiert, wenn die Route zur Weiterleitung eines Datenpakets benötigt wird

Parameter	Wert
NODE-PLACEMENT	RANDOM
MOBILITY	NONE
RADIO-BANDWIDTH	250 kbits/s
MAC-PROTOCOL	802.11
PROPAGATION-PATHLOSS	TWO-RAY
RADIO-TYPE	RADIO-NONOISE
RADIO-FREQUENCY	0.433e9
RADIO-TX-POWER	0.0 dBm
RADIO-RANGE	158m
ROUTING-PROTOCOL	PRECALC
NETWORK-PROTOCOL	IP
TERRAIN-DIMENSIONS	500x500m
NUMBER-OF-NODES	100, 250, 1000
SIMULATION-TIME	NUMBER-OF-NODES * 120s

Tabelle 6.1: Verwendete GloMoSim-Simulationsparameter

tragung der periodischen Hello-Nachrichten in Netzen mit hoher Knotendichte (siehe Abschnitt 6.2) ein Vielfaches der zur Verfügung stehenden Bandbreite. Durch die große Anzahl erzeugter Simulatorereignisse steigt zudem der für die Simulation benötigte Rechenaufwand drastisch an, so dass eine sinnvolle Simulation langer Zeitspannen nicht mehr möglich ist.

Die reaktiven Routingprotokolle AODV und DSR zeigen in Hinblick auf den benötigten Kommunikationsaufwand und die Simulationsgeschwindigkeit ein deutlich besseres Verhalten. Dennoch benötigt der Signalisierungsverkehr dieser Protokolle einen Großteil der vorhandenen Übertragungskapazität und verhindert somit den sinnvollen Einsatz von kommunikationsaufwändigeren Anwendungen. Schließlich treten ab einer hohen Knotenanzahl gelegentlich Fehler wie das Duplizieren von Paketen oder der vollständige Verlust der Routingtabelle auf.

Da für die Simulationen kein Sensornetz-Routingprotokoll zur Verfügung stand und die vorhandenen Protokolle für große Netze nicht geeignet sind, wurden die Routingtabellen (wie in Abschnitt 5.3.4 beschrieben) statisch berechnet und auf Knotenmobilität verzichtet. Auf die Simulationsergebnisse hat diese Vereinfachung keinen direkten Einfluss, da das Dienstverzeichnis SCANv3 als Overlaynetz unabhängig von der Realisierung des Underlays ist solange das Netz nicht partitioniert. Aufgrund der hohen Knotendichte eines Sensornetzes ist dieses Ereignis jedoch unwahrscheinlich. Durch die höhere Paketverlustrate sowie die größere Ende-zu-Ende-Verzögerung, die sich durch Verwendung eines dynamischen Sensornetz-Routingprotokolls ergeben könnten, würde lediglich das Aufsuchen eines Dienstes mehr Zeit benötigen und der Kommunikationsaufwand durch notwendige Sendewiederholungen steigen. Sobald entsprechende Implementierungen der Sensornetzprotokolle zur Verfügung stehen, können diese aufgrund der modularen Schichtenarchitektur von GloMoSim einfach integriert und die Simulationen wiederholt werden.

In Tabelle 6.1 sind die GloMoSim-Simulationsparameter, die für diese Arbeit verwendet wurden, nochmals aufgelistet.

6.2 Simulationsszenarien

Die Simulation von SCANv3 soll zeigen, ob das Dienstverzeichnis in typischen Anwendungsszenarien für dienstorientierte Sensornetze das sichere Auffinden von Diensten mit akzeptablem Kommunikationsaufwand ermöglicht. Die Anzahl der verwendeten Sensorknoten in einem Netz hängt dabei stark vom jeweiligen Anwendungsgebiet ab. Für den anfangs genannten Einsatz in der häuslichen Pflege werden üblicherweise wesentlich weniger Sensoren benötigt als zur Überwachung einer Großveranstaltung. Um die Auswirkungen der Sensornetzgröße auf das Dienstverzeichnisses untersuchen zu können, werden die Simulationen jeweils mit 100, 250 und 1000 Sensorknoten, die zufällig auf einem Gebiet von 500x500m platziert sind, durchgeführt. Aufgrund der maximalen Sendereichweite von 158m erfordert die Kommunikation zwischen zwei Knoten im Underlay somit im Schnitt 2-3 Hops. Um die Heterogenität der Sensoren und Aktoren eines dienstorientierten Netzes zu simulieren, variiert die Anzahl und Art der Dienste, die ein Sensorknoten während der Simulation einträgt bzw. aufsucht.

Die Simulationen bestehen aus dem vollständigen Aufbau eines neuen Sensornetzes. Dazu werden die folgenden Parameter so gewählt, dass sie einem typischen Anwendungsszenario in der Praxis entsprechen. Beim Aufbau des Netzes wird alle 120s ein neuer Knoten zum Overlay hinzugefügt. Nach dem erfolgreichen Beitritt sucht der Knoten maximal 10 von insgesamt 100 verschiedenen Diensten auf. Danach fügt er die Dienste, die er selbst anbietet (maximal 7), in das Dienstverzeichnis ein. Alle 30 Minuten werden die Dienste durch den Knoten erneut eingetragen und aufgesucht. Die Simulation wird beendet, sobald alle Knoten dem Netz beigetreten sind.

Um die Effektivität der in Abschnitt 4.1 vorgeschlagen Mechanismen überprüfen zu können, werden die Dienste in verschiedenen Simulationsläufen mit den 3 Verfahren SPX, SPX mit Replikaten und MPX mit Replikaten eingetragen. Um Ressourcen zu sparen, wird zum Auffinden der Dienste immer SPX verwendet. Falls ein Dienst mit Replikaten eingefügt wird, werden auch zum Auffinden des Dienstes alle Replikate verwendet. Wie bereits bei den theoretischen Überlegungen in Abschnitt 4.1.4 entspricht die Anzahl der verteilten Geheimnisse bei MPX sowie die Zahl der verwendeten Replikate der Dimensionalität des SCAN-Overlays. Damit der Anteil der erfolgreich durchgeführten Lookup-Operationen in einem Netz mit bösartigen Knoten ermittelt werden kann, wird ein Angreifer auf die in Abschnitt 5.3.3 dargelegte Weise simuliert.

6.3 Simulationsergebnisse

6.3.1 Topologie des Overlays

Zunächst soll ein Blick auf die Topologie des Overlays geworfen werden, sobald alle Knoten dem Netz beigetreten sind. In Abbildung 6.1 ist die durchschnittliche Anzahl der Overlay-Nachbarn eines Knotens zum Ende der Simulation dargestellt. Wie bereits in Abschnitt 2.3 beschrieben beträgt die Anzahl der Overlay-Nachbarn in einem idealisierten d-dimensionalen CAN unabhängig von der Gesamtzahl der Knoten $2d$. Für die Fälle $d = 2$ und $d = 3$ entspricht das Simulationsergebnis weitgehend diesen Annahmen. Aufgrund einer leicht ungleichmäßigen Verteilung der Zonengrößen ergibt sich eine geringfügig höhere Anzahl als $2d$ Nachbarn.

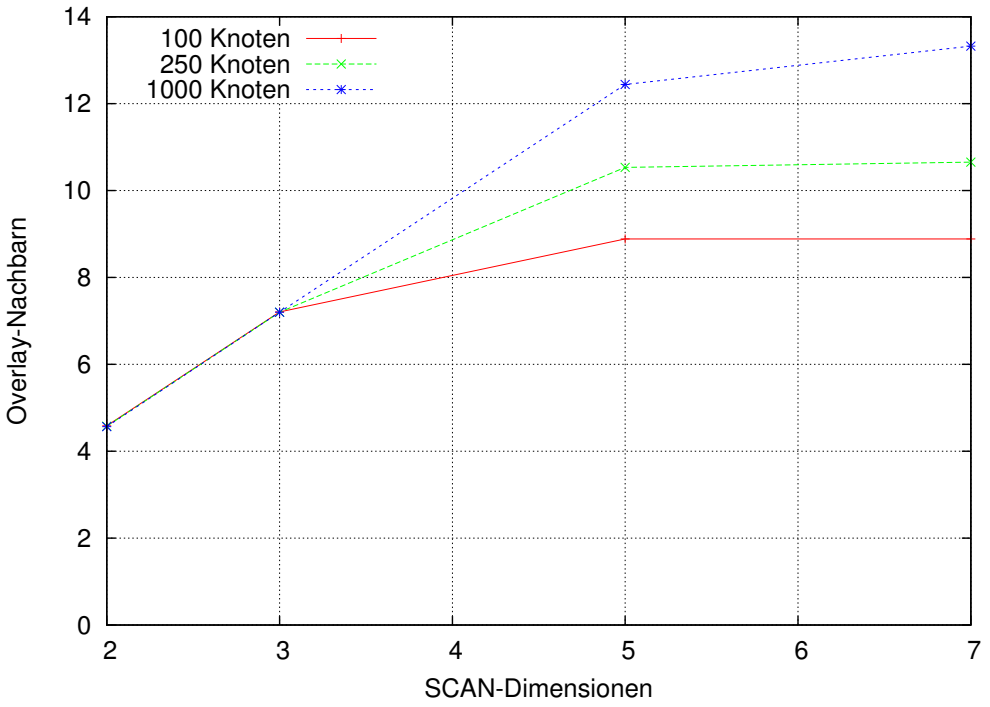

Abbildung 6.1: Durchschnittliche Anzahl der Overlay-Nachbarn eines Knotens zum Ende der Simulation

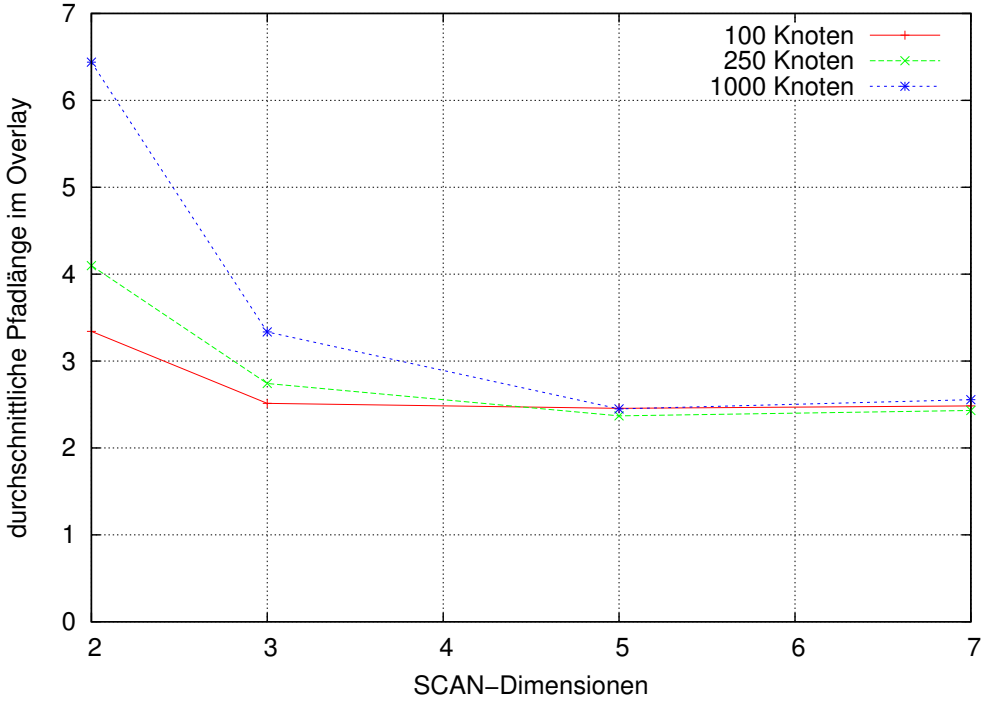

Abbildung 6.2: Durchschnittliche Pfadlänge im Overlay über den gesamten Simulationszeitraum gemessen

In den Simulationsläufen mit $d = 5$ und $d = 7$ Dimensionen liegt die Zahl der Nachbarn jedoch für eine geringe Anzahl von Knoten deutlich unter dem Wert $2d$. In diesen Fällen ist die Gesamtzahl der Knoten nicht ausreichend um jedem Knoten $2d$ disjunkte Nachbarknoten zuzuweisen. Aus topologischer Sicht grenzen zwar weiterhin an eine Zone in jeder Dimension zwei Nachbarzonen an, diese Zonen sind jedoch nicht mehr in allen Fällen verschieden. Dieser Effekt lässt sich auch an den in Abbildung 6.2 dargestellten durchschnittlichen Pfadlängen im Overlay erkennen. Sobald sich durch eine Steigerung der CAN-Dimensionalität die Zahl der Overlay-Nachbarn nicht mehr erhöhen lässt, bleibt auch die Pfadlänge im Overlay unverändert.

Dies wirkt sich auch auf die Verfahren zum sicheren Einfügen und Auffinden von Diensten aus. Durch eine geringere Anzahl von Nachbarn stehen auch weniger disjunkte Pfade zum Austausch eines Sitzungsschlüssels zur Verfügung. In Netzen mit geringer Knotenanzahl kann, selbst in einem hochdimensionalen CAN-Overlay, durch eine größere Anzahl von verteilten Geheimnissen bzw. Replikaten die Sicherheit der Verfahren nicht merklich erhöht werden. Dieser Effekt lässt sich auch deutlich an den in Abschnitt 6.3.3 präsentierten Simulationsergebnissen erkennen. Des Weiteren muss beachtet werden, dass die in Abbildung 6.1 für eine bestimmte Netzgröße wiedergegebene Anzahl an Overlay-Nachbarn erst zum Ende des Netzaufbaus erreicht werden. In der Anfangsphase des Netzaufbaus[2] steht somit eine deutlich geringere Anzahl disjunkter Pfade zum Austausch eines Schlüssels zur Verfügung.

6.3.2 Sicherheit und Kommunikationsaufwand

Im Folgenden werden die Erfolgswahrscheinlichkeit für einen Lookup sowie der benötigte Kommunikationsaufwand für unterschiedliche Szenarien evaluiert. Eine Lookup-Operation zählt als *erfolgreich*, falls ein zum Zeitpunkt der Lookup-Operation im Netz vorhandener Dienst korrekt aufgefunden wird. Wird der gleiche Dienst von mehreren Knoten angeboten, so wird der Lookup-Erfolg für jeden Knoten separat gewertet. Befinden sich beispielsweise zum Zeitpunkt einer Lookup-Operation für den Dienst *Temperatursensor* im Netz 10 Knoten, die diesen Dienst anbieten, und werden davon 6 Knoten aufgefunden, beträgt der Anteil erfolgreicher Lookup-Operationen 0,6. In vielen Anwendungsszenarios ist es ausreichend, falls mindestens ein Knoten oder ein bestimmter Anteil von Knoten mit dem gesuchten Dienst aufgefunden wird. Die hier getroffene Definition einer erfolgreichen Lookup-Operation betrachtet dagegen ein „worst-case“-Szenario, in dem möglichst alle Knoten, die einen Dienst anbieten, auch aufgefunden werden müssen.

Der Einfluss der CAN-Dimensionalität und Zahl der verwendeten Replikate auf den Anteil erfolgreicher Lookup-Operationen ist in Abbildung 6.3 zu sehen. Das Schaubild zeigt den Anteil der erfolgreich durchgeführten Lookup-Operationen in einem Netz mit 250 Knoten bei der Verwendung von SPX mit Replikaten zum Einfügen der Dienste. Bis zu $d = 5$ Dimensionen ist eine deutliche Steigerung der erfolgreichen Lookup-Operationen zu sehen. Eine Erhöhung der Dimensionalität auf $d = 7$ ergibt hingegen keine signifikante Verbesserung. Dies ist auf den im vorherigen Absatz beschriebenen Effekt zurückzuführen und entspricht den Ergebnissen aus Abbildung 6.1.

[2] In vielen Veröffentlichungen im Bereich der Overlay-Netze wird der Aufbau des Netzes nicht weiter betrachtet und von einem bereits bestehenden Netz ausgegangen

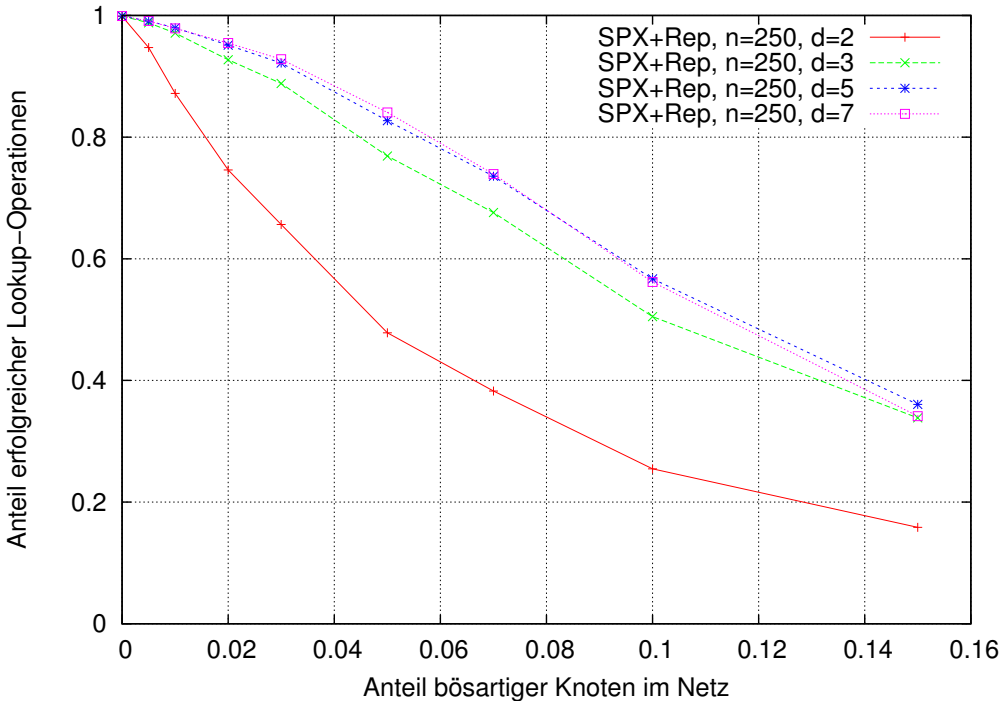

Abbildung 6.3: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem Netz mit n=250 Knoten, falls zum Einfügen der Dienste SPX mit Replikaten verwendet wird

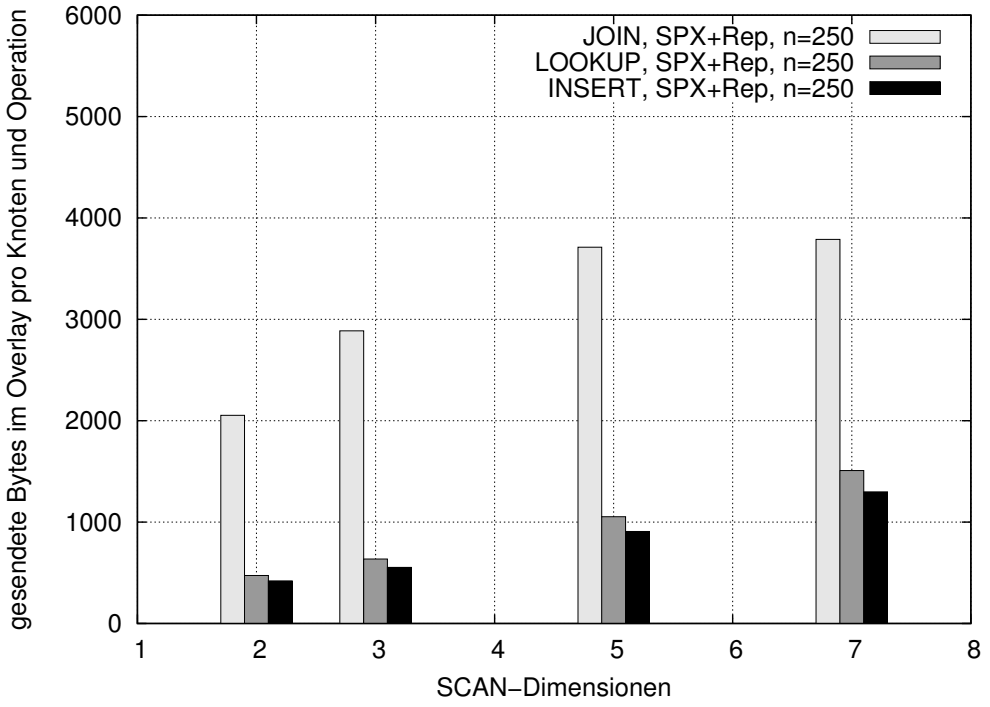

Abbildung 6.4: Kommunikationsaufwand für eine Join-, Lookup- und Insert-Operation pro Overlayknoten in einem Netz mit n=250 Knoten, falls zum Einfügen der Dienste SPX mit Replikaten verwendet wird.

In Abbildung 6.4 ist der Kommunikationsaufwand, der pro Knoten und Operation für dieses Szenario im Overlay entsteht, dargestellt. Mit zunehmender Dimensionalität steigt auch der Aufwand für das Hinzufügen eines neuen Knotens. Dies resultiert aus der Tatsache, dass bei einer Zonenteilung eine größere Anzahl von Nachbarknoten kontaktiert und mit neuem Schlüsselmaterial versorgt werden müssen. Da ab $d = 5$ die Zahl der Overlay-Nachbarn nicht weiter zunimmt, stagniert der Aufwand für eine Join-Operation hier ebenfalls. Der Aufwand für eine Lookup- bzw. Insert-Operation steigt kontinuierlich mit der Zahl der verwendeten Replikate. Da sich die Implementierung bei der Festlegung der Zahl der verwendeten Replikate nur anhand der Dimensionalität des CAN orientiert und die Zahl der Nachbarn unberücksichtigt lässt, steigt der Kommunikationsaufwand auch für $d = 7$ weiter an, ohne für ein größeres Maß an Sicherheit zu sorgen. Die Kosten für eine Lookup-Operation sind geringfügig höher als für eine Insert-Operation, obwohl für beide Operationen SPX mit Replikaten eingesetzt wird. Der Grund für den zusätzlichen Kommunikationsaufwand einer Lookup-Operation liegt in der Übertragung der Liste von Netzwerkadresse der Knoten, die den gesuchten Dienst anbieten. Zum Einfügen eines Dienstes muss hingegen nur die eigene Netzwerkadresse des Knotens übertragenen werden.

6.3.3 Vergleich verschiedener Szenarien

In diesem Abschnitt wird die Effizienz der Sicherheitsmechanismen SPX sowie SPX und MPX mit Replikaten in Netzen mit 100, 250 und 1000 Knoten evaluiert. Die Simulationsergebnisse dazu sind in den Abbildungen 6.5 bis 6.11 dargestellt.

Davon zeigen die Abbildungen 6.5 und 6.6 den Anteil erfolgreicher Lookup-Operation sowie den Kommunikationsaufwand für ein Netz mit $n = 100$ Knoten und $d = 5$ Dimensionen. Es lässt sich deutlich erkennen, dass durch die Verwendung von Replikaten der Anteil erfolgreicher Lookup-Operationen signifikant erhöht werden kann. Die zusätzliche Verwendung des aufwändigeren Schlüsselaustauschverfahrens MPX führt jedoch nicht zu einer Verbesserung der Ergebnisse. Ist der Anteil der bösartigen Knoten groß, werden durch das Verfahren MPX sogar weniger korrekte Lookup-Operationen ermöglicht. Dieses Ergebnis war jedoch aufgrund der in Abschnitt 4.1.4 angestellten Überlegungen zu erwarten. Anhand des in Abbildung 6.6 dargestellten Kommunikationsaufwands werden die Kosten, die für eine höheres Maß an Sicherheit erforderlich sind, deutlich. Durch die Verwendung von fünf Replikaten pro Dienst, steigt auch der Aufwand für eine Lookup- bzw. Insert-Operation um annähernd das Fünffache. Dem nochmals wesentlich höheren Aufwand zum Einfügen der Replikate mittels MPX steht kein entsprechender Sicherheitsgewinn gegenüber, so dass die Verwendung von MPX mit Replikate in diesem Szenario nicht sinnvoll ist.

Wird die Zahl der Sensorknoten auf $n = 250$ erhöht, fällt der Anteil erfolgreicher Lookup-Operationen wie in Abbildung 6.7 zu sehen ist bei einem hohen Anteil bösartiger Knoten geringer aus als in dem kleineren Netz mit 100 Knoten. Dies deckt sich mit den Berechnungen aus Abschnitt 4.1.4. Angesichts der Tatsache, dass die absolute Zahl der bösartigen Knoten gegenüber dem Szenario mit 100 Knoten um den Faktor 2,5 größer ist, ist dies ein gutes Ergebnis. Allerdings kann auch in diesem Szenario durch Verwendung von MPX die Wahrscheinlichkeit für eine korrekte Lookup-Operation nicht erhöht werden.

Beim Vergleich des in Abbildung 6.8 dargestellten Kommunikationsaufwands mit dem des kleineren Netzes in Abbildung 6.6 fällt auf, dass der Aufwand für eine

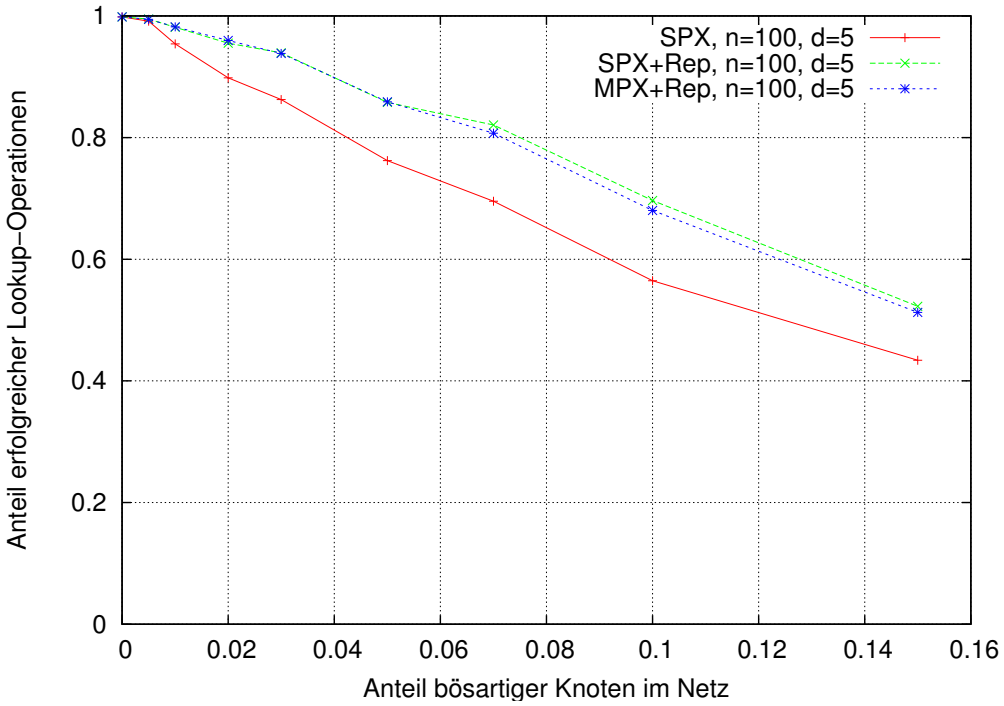

Abbildung 6.5: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem 5-dimensionalen SCAN mit n=100 Knoten abhängig vom verwendeten Sicherheitsmechanismus zum Einfügen der Dienste.

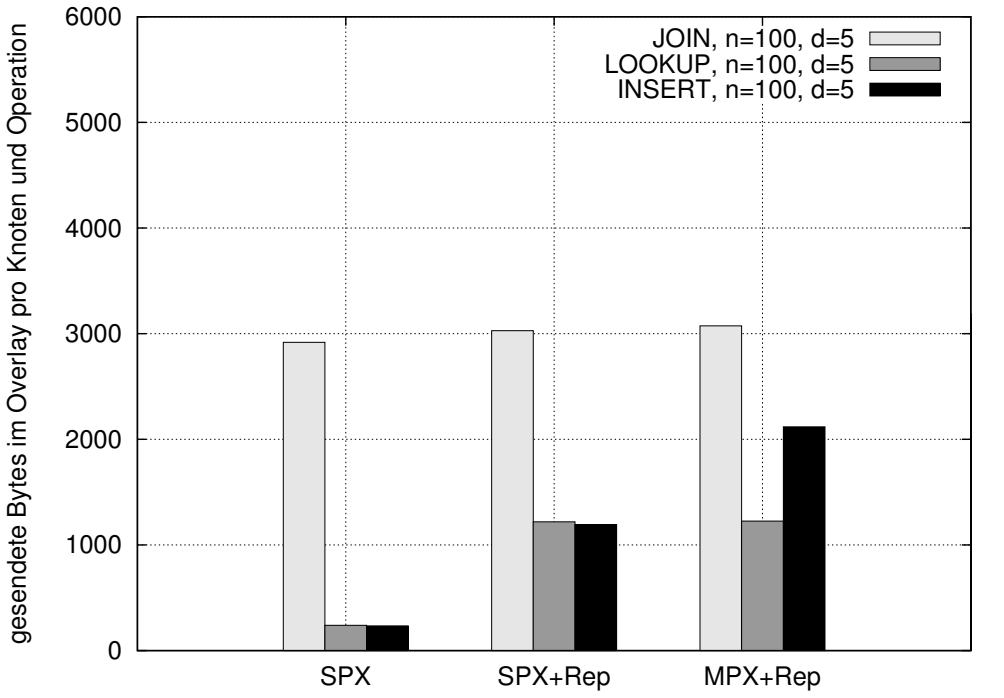

Abbildung 6.6: Kommunikationsaufwand für eine Join-, Lookup- bzw. Insert-Operation pro Overlayknoten in einem Netz mit n=100 Knoten abhängig vom verwendeten Sicherheitsmechanismus zum Einfügen der Dienste.

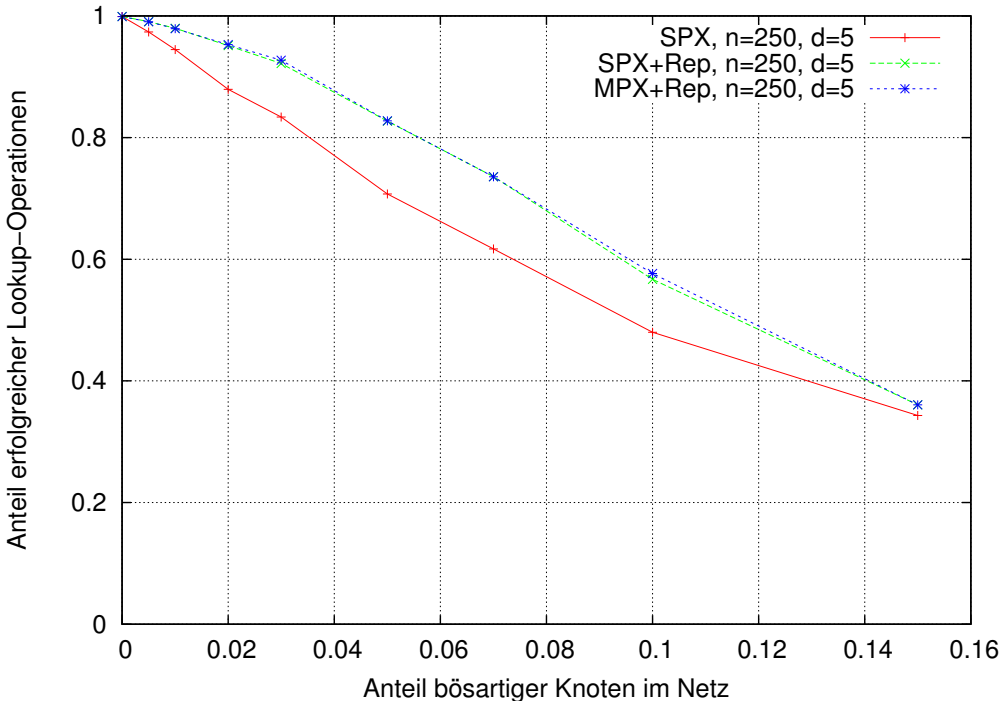

Abbildung 6.7: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem 5-dimensionalen SCAN mit n=250 Knoten abhängig vom verwendeten Sicherheitsmechanismus zum Einfügen der Dienste.

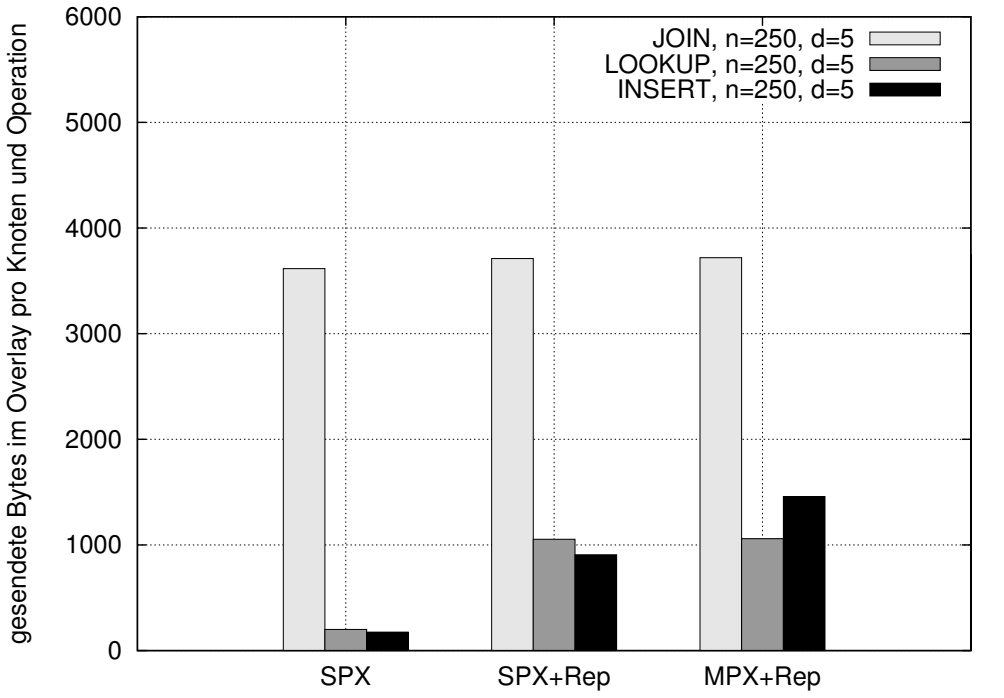

Abbildung 6.8: Kommunikationsaufwand für eine Join-, Lookup- bzw. Insert-Operation pro Overlayknoten in einem Netz mit n=250 Knoten abhängig vom verwendeten Sicherheitsmechanismus zum Einfügen der Dienste.

	100 Knoten	250 Knoten	1000 Knoten
Simulationszeit [s]	12000s	30000s	120000s
Lookup-Operationen pro Knoten	21,7	49,2	188,9
Insert-Operationen pro Knoten	15,1	35,4	135,3

Tabelle 6.2: Durchschnittliche Anzahl durchgeführter Insert- und Lookup-Operationen

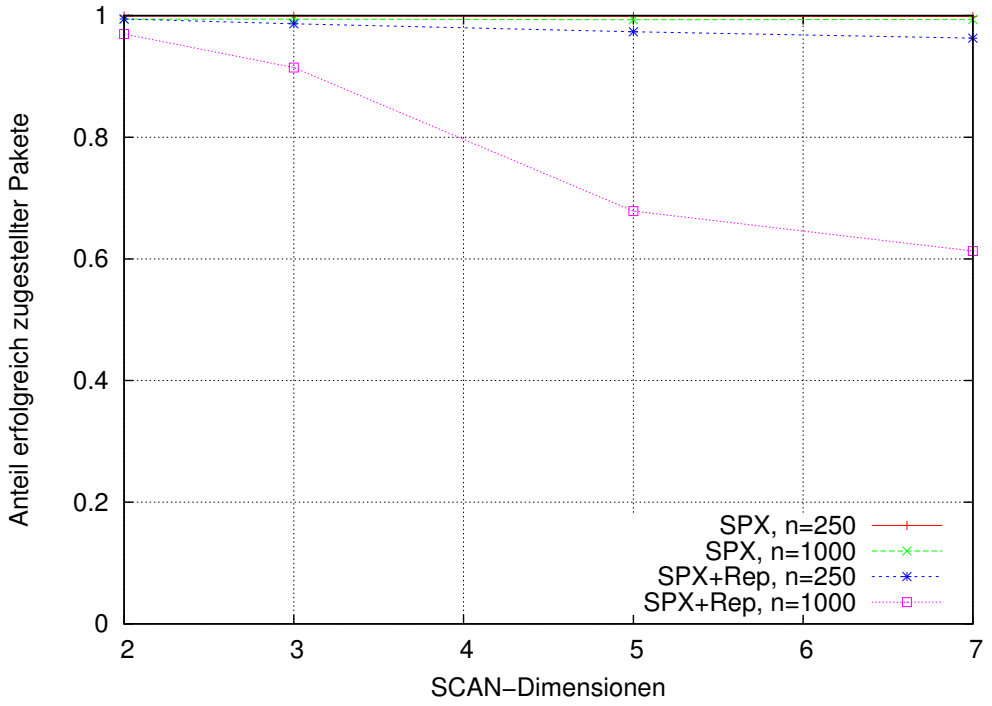

Abbildung 6.9: Anteil der Overlay-Nachrichten, die erfolgreich zugestellt wurden und nicht aufgrund von Paketverlusten im Underlay verloren gegangen sind

Lookup- bzw. Insert-Operation gesunken ist. Dies ist darauf zurückzuführen, dass in der Implementierung für das erneute Einfügen oder Auffinden eines Dienstes in vielen Fällen ein bereits vorhandener Sitzungsschlüssel wiederverwendet wird (siehe Abschnitt 4.3) und ein erneuter Schlüsselaustausch entfällt. Lediglich in Fällen, in denen die entsprechende Zone geteilt wurde und der bisher zuständige Knoten einen Dienst nicht mehr verwaltet, muss erneut ein Schlüssel ausgetauscht werden. Da die Simulationszeit abhängig von der Knotenanzahl gewählt wurde und Diensteinträge alle 1800s erneuert werden, ist die Zahl der pro Knoten durchgeführten Lookup- und Insert-Operationen in einem Netz mit vielen Knoten höher. In Tabelle 6.2 ist aufgeführt wie viele Operationen abhängig vom gewählten Szenario im Schnitt durchgeführt wurden. In den Simulationsläufen mit großer Knotenzahl sind somit die durchschnittlichen Kosten für eine Lookup- oder Insert-Operation geringer, da für viele dieser Operationen kein Schlüsselaustausch stattfinden musste.

Bei den Simulationen mit $n = 1000$ Knoten ergaben sich bei Verwendung vieler Replikate durch Überlastsituationen im Underlay zum Teil hohe Paketverlustraten (siehe Abbildung 6.9). Diese sind auf die hohe Anzahl von Lookup- und Insert-Operationen zurückzuführen, die durch die periodische Erneuerung von Dienstein-

träge gegen Simulationsende hervorgerufen werden. Fallen diese periodischen Erneuerungen vieler Knoten zeitlich zusammen, führt dies durch die Summe der gesendeten Nachrichten zu einer hohen Auslastung des Mediums und vermehrt zu Kollisionen beim Medienzugriff. Aufgrund der daraus resultierenden Pufferüberläufe werden viele Overlay-Pakete verworfen.

Um dennoch einen Vergleich der Sicherheitsmechanismen für solche Netze zu ermöglichen, wurden die Simulationen der Mechanismen SPX mit Replikaten und MPX mit Replikaten für 1000 Knoten mit einer Datenübertragungsrate von 11 Mbit/s durchgeführt. Da sich die Simulation eines Netzes dieser Größenordnung mit GloMoSim als sehr rechenaufwändig[3] erwiesen hat, wurden die Szenarien mit 1000 Knoten nur mit jeweils 3 Seeds berechnet, während für die übrigen Simulationen 20 Seeds verwendet wurden. Da die Abweichungen der einzelnen Simulationsläufe mit 1000 Knoten jedoch hinreichend gering ausgefallen sind, sollen die Ergebnisse hier dennoch wiedergegeben werden.

Die Abbildungen 6.10 und 6.11 zeigen die Ergebnisse der Simulationen mit 1000 Knoten. Im Gegensatz zu den Simulationen mit weniger Knoten ist hier erstmals durch die Verwendung von MPX zum Einfügen der Dienste eine Verbesserung der Lookup-Ergebnisse im Vergleich zu SPX zu sehen. Dies legt die Vermutung nahe, dass die größere Anzahl der disjunkten Pfade, die sich aus der größeren Anzahl von Overlay-Nachbarn ergibt (siehe Abbildung 6.1), der Grund dafür ist. Vergleicht man in einem Netz mit 250 Knoten die Verfahren SPX mit Replikaten und MPX mit Replikaten mit $d = 3$ anstelle von $d = 5$ Dimensionen (siehe Abbildung A.9 und Abbildung A.11 im Anhang), stellt man fest, dass in diesem Fall ebenfalls durch Verwendung von MPX das Lookup-Ergebnis verbessert werden kann. In diesem Fall ist ebenfalls die durchschnittliche Anzahl der Overlay-Nachbarn im Verhältnis zur Anzahl der verwendeten Replikate bzw. verteilten Geheimnisse höher. In einem Netz mit 100 oder 250 Knoten überlappen bei $d = 5$ Dimensionen die von MPX und Replikaten verwendeten Pfade. In diesen Fällen kann durch zusätzliche Verwendung von MPX die Sicherheit nicht erhöht werden. Allerdings kann dann der alleinige Einsatz von MPX ohne die Verwendung von Replikaten sinnvoll sein, da auf diese Weise der Speicher für die Replikate eingespart und gleichzeitig die Sicherheit erhöht wird.

Ein weiteres Problem mit MPX entsteht durch die Verwendung des Challenge-Response-Verfahrens um die Gültigkeit der verteilten Geheimnisse zu überprüfen. Um den Kommunikationsaufwand zu reduzieren, verwendet die Implementierung nach drei erfolglosen Versuchen mit MPX das einfachere Verfahren SPX. Falls ein Mechanismus wie VSS (siehe Abschnitt 4.1.2) zur Verfügung steht, der manipulierte Geheimnisteile erkennt, kann diese Maßnahme entfallen und in jedem Fall ein Schlüssel mit MPX aufgebaut werden. Simulationen haben ergeben, dass in einem 5-dimensionalen CAN mit 250 Knoten bei einem Anteil von 5% bösartiger Knoten durch Verwendung von MPX mit VSS der Anteil erfolgreicher Lookup-Operationen beispielsweise von 77.2% auf 79,4% gesteigert werden konnte.

[3]Die Simulationen dieser Arbeiten wurden verteilt auf rund 40 CPUs durchgeführt und benötigten einen Gesamtrechenaufwand von über 15000 Stunden.

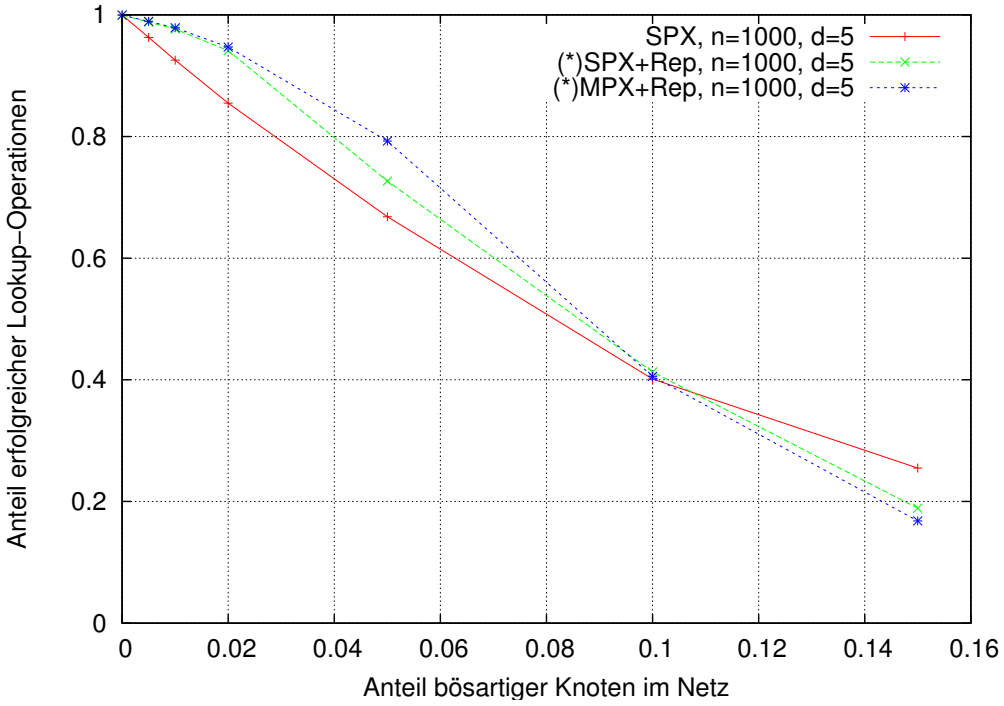

Abbildung 6.10: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem 5-dimensionalen SCAN mit n=1000 Knoten abhängig vom verwendeten Sicherheitsmechanismus zum Einfügen der Dienste.
(*) Simulation wurde mit 11Mbit/s Datenübertragungsrate durchgeführt

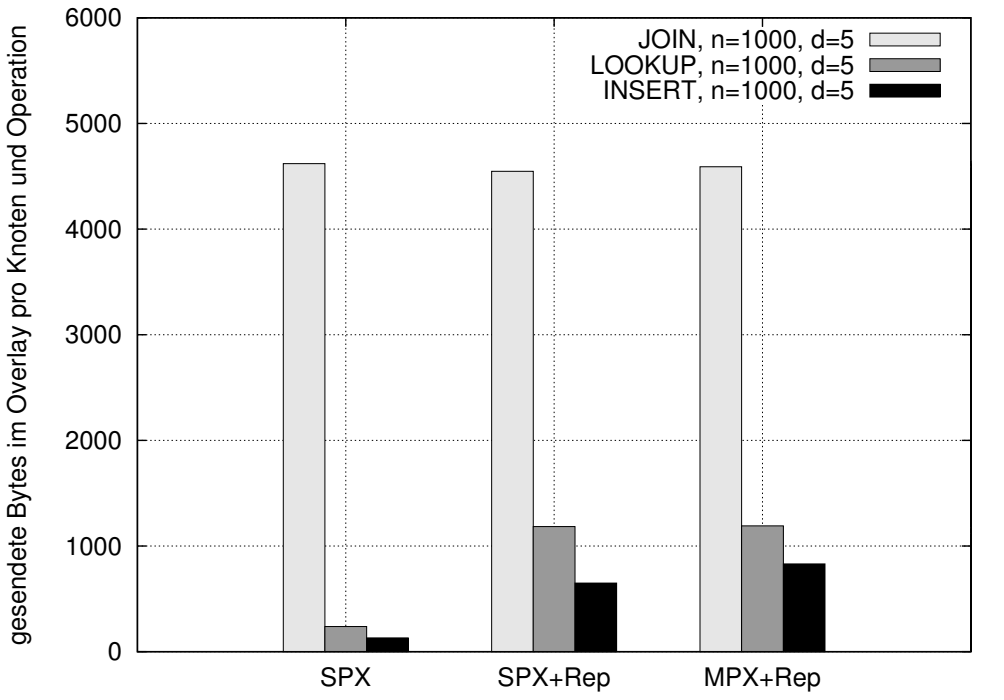

Abbildung 6.11: Kommunikationsaufwand für eine Join-, Lookup- bzw. Insert-Operation pro Overlayknoten in einem Netz mit n=1000 Knoten abhängig vom verwendeten Sicherheitsmechanismus zum Einfügen der Dienste.

6.4 Bewertung der Ergebnisse

Die Simulationen haben gezeigt, dass das im Simulator implementierte Protokoll SCANv3 geeignet ist um ein vollständiges CAN-Overlay aufzubauen und Nachrichten im Overlay zu vermitteln. Ebenso ist das Einfügen und Auffinden von Diensten in einem Netz bestehend aus gutartigen Knoten problemlos möglich.

Die Sicherheit der vorgeschlagenen Verfahren zum Austausch eines Sitzungsschlüssels sowie zum sicheren Einfügen und Auffinden von Diensten wurde durch Simulation eines Insider-Angreifers evaluiert. Der simulierte Angreifer versucht dabei das Auffinden eines Dienstes zu verhindern und, falls genug kooperierende Angreifer zur Verfügung stehen, manipulierte Datensätze zurückzuliefern (siehe Abschnitt 5.3.3). Die durch die Simulation gewonnenen Ergebnisse entsprechen weitgehend dem, was anhand der theoretischen Überlegungen aus Kapitel 4 zu erwarten war. Durch aufwändigere Sicherheitsmechanismen wie das Schlüsselaustauschprotokoll MPX und die Verwendung von Replikaten kann der Anteil der erfolgreichen Lookup-Operationen deutlich erhöht werden. In den theoretischen Berechnungen wurde nicht berücksichtigt, dass durch den Aufbau des Overlays ein Knoten anfangs wenige Overlay-Nachbarn besitzt und daher auch nur wenige disjunkte Pfade zur Verfügung stehen. In Netzen mit geringer Knotenzahl und hoher CAN-Dimensionalität ergeben sich auch noch nach dem vollständigen Aufbau des Netzes weniger als $2d$ disjunkte Pfade. In diesen Fällen führt die Verwendung von MPX und Replikaten zu keiner Verbesserung der Sicherheit des Dienstverzeichnisses und sollte daher aus Effizienzgründen vermieden werden. Die Verwendung von MPX stellt jedoch ein geeignetes Verfahren dar, um zwischen zwei Overlayknoten auf möglichst sichere Weise einen gemeinsamen Schlüssel zu etablieren. Ein solcher Schlüssel wird beispielsweise für das in Abschnitt 4.4 vorgeschlagene Verfahren zur Behandlung von Knotenausfällen benötigt. Damit der Kommunikationsaufwand für einen Schlüsselaustausch mit MPX in Gegenwart eines Angreifers möglichst gering gehalten werden kann, empfiehlt sich wie in Abschnitt 4.1.2 vorgeschlagen die Verwendung von VSS.

Der Kommunikationsaufwand von SCANv3 hängt von einer Vielzahl von Parametern ab. Da das Dienstverzeichnis ein Overlay-Netz darstellt, spielt die durchschnittliche Pfadlänge im Underlay für den Kommunikationsaufwand eine entscheidende Rolle. In den simulierten Szenarien beträgt diese im Schnitt 2-3 Hops. Um den Kommunikationsaufwand im Underlay zu berechnen, muss die Zahl der gesendeten Bytes im Overlay daher um diesen Faktor multipliziert werden. In Underlay-Netzen mit längeren Pfadlängen ist der Aufwand somit entsprechend größer. Durch die Wahl eines geeigneten Schlüsselaustauschverfahrens und einer geeigneten Anzahl von Replikaten lässt sich das Sicherheitsniveau sowie der benötigte Kommunikationsaufwand individuell an die Erfordernisse eines Anwendungsszenarios anpassen. Das Sicherheitsniveau, dass durch die Verwendung von SPX mit drei bis fünf Replikaten in Netzen bis 1000 Knoten erzielt wird, bietet bei einem vertretbaren Kommunikationsaufwand in vielen Fällen ausreichend Schutz vor Insider-Angreifern. Ein Vergleich des Kommunikationsaufwands mit anderen Dienstverzeichnissen konnte mangels frei verfügbarer Implementierungen nicht durchgeführt werden. Die einzige momentan verfügbare Implementierung eines Dienstverzeichnis für Ad-hoc-Netze [Nuev04] scheint nicht für den Einsatz in Netzen mit 100 bis 1000 Knoten geeignet zu sein und führte bei mehreren durchgeführten Simulationsläufen mit einer großen Anzahl von Knoten zum Absturz des Simulators.

Bei der Bewertung der Sicherheitsarchitektur wurden die in Abschnitt 2.4.1 beschriebenen Angriffsszenarien betrachtet. Die Klasse der DoS-Angriffe wurde von den Sicherheitsbetrachtungen jedoch weitgehend ausgeschlossen. Beispielsweise sieht SCANv3 keine Verfahren vor, mit dem überprüft werden kann, ob Dienstbeschreibungen, die von einem Knoten eingetragen werden, gültig sind und der Dienst von dem Knoten auch erbracht werden kann. Dennoch bietet das Dienstverzeichnis SCANv3 mit seiner umfassenden Sicherheitsarchitektur ein hohes Maß an Sicherheit ohne auf den Sensorknoten aufwändige kryptographische Verfahren einzusetzen und eignet sich daher zum sicheren Auffinden von Diensten in Sensornetzen.

7. Zusammenfassung und Ausblick

Sensornetze werden in Zukunft in vielen Bereichen des täglichen Lebens die Erfassung von Daten grundlegend verändern. Aufgrund der geringen Kosten eines Sensorknotens können Hunderte dieser Knoten zur sensorischen Erfassung der Umwelt eingesetzt werden und liefern somit kontinuierlich ein exaktes Abbild ihrer Umgebung.

In dienstorientierten Sensornetzen werden durch die Verknüpfung verschiedener Sensoren und Aktoren neue, komplexere Dienste erschaffen. Typischerweise sind diese Netze heterogen und erfordern von den Sensorknoten die selbständige Bewältigung komplexer Aufgaben. Dienstorientierte Sensornetze können beispielsweise zur Gebäudeautomation oder zur Patientenüberwachung in der Altenpflege eingesetzt werden. Durch den Einsatz von Sensornetzen können anfallende Arbeiten erleichtert sowie Fehler besser erkannt werden. Da insbesondere der Einsatz im Gesundheitswesen einen sensiblen Umgang mit Patientendaten erfordert, müssen die eingesetzten Verfahren ausreichend Schutz vor Angreifern bieten.

Damit durch die Verknüpfung verschiedener Dienste dynamisch ein neuer Dienst geschaffen werden kann, wird ein Verfahren benötigt, das den Sensorknoten erlaubt, vorhandene Dienste aufzufinden sowie eigene Dienste zu veröffentlichen. Diese Leistungen können durch Dienstverzeichnisse erbracht werden.

Sensorknoten haben aufgrund ihrer geringen Größe und niedrigen Herstellungskosten stark beschränkte Ressourcen. Bei dem Entwurf von Protokollen für Sensornetze muss daher der Rechen- und Kommunikationsaufwand auf ein Minimum reduziert werden. Aus diesem Grund kann ein Großteil der vorhandenen Lösungen aus dem Festnetz- und Ad-hoc-Bereich nicht sinnvoll in Sensornetzen verwendet werden.

Aufgabenstellung dieser Diplomarbeit war die Analyse, Optimierung und Evaluierung des sicheren verteilten Dienstverzeichnisses SCANv2 [HoBZ04]. Um die im Rahmen der Diplomarbeit durchgeführten Verbesserungen des Dienstverzeichnisses evaluieren zu können, wurde das Protokoll in einem Simulator implementiert.

7.1 Ergebnisse der Arbeit

Im ersten Teil der Arbeit wurde die Sicherheit von SCANv2 untersucht. Die Bewertung der Sicherheit erfolgte durch Aufstellung geforderter Schutzziele sowie möglicher Angriffsszenarien auf das Dienstverzeichnis. Um eine umfassende Liste möglicher Angriffe zu erhalten, wurde diese durch Zusammenstellung aktueller Veröffentlichungen aus den Bereichen Sicherheit in Sensornetzen und Sicherheit in DHTs gewonnen.

Die Sicherheitsarchitektur von SCANv2 soll das Dienstverzeichnis vollständig vor Angriffen durch Außenstehende schützen. Wird einer der Sensorknoten physisch manipuliert, so soll der verursachte Schaden durch einen solchen Insider-Angreifer zumindest lokal begrenzt bleiben. Die durchgeführte Sicherheitsanalyse brachte einige Schwachstellen in SCANv2 zu Tage, die eine Verletzung dieser Anforderung ermöglichen. Zum Beispiel sieht das Protokoll zum Schutz vor Replay-Angriffen die Verwendung von Zeitstempel vor. Diese Maßnahme wird jedoch nicht konsequent genug verwendet, so dass ein Outsider-Angreifer dennoch durch das Wiedereinspielen von Protokollpaketen Einfluss auf die Overlay-Topologie nehmen kann. Ein weitere Schwachstelle entsteht durch die unsichere Verwendung von Gruppenschlüsseln nach dem Beitritt neuer Knoten. Dadurch kann ein Knoten unberechtigterweise nach einem Knotenausfall die Wahl eines Nachfolgerknotens für die ausgefallene Zone beeinflussen. Das vorgeschlagene Verfahren zu Behandlung von Knotenausfällen bietet jedoch an sich noch weitere Schwachstellen und ist insgesamt zur sicheren Behandlung von Knotenausfällen nicht geeignet. Schließlich fehlt in [HoBZ04] die Spezifikation eines Mechanismus, der das sichere Einfügen und Auffinden von Diensten erlauben würde.

Im zweiten Teil der Arbeit wurden einige der Schwachstellen von SCANv2 beseitigt und die verbesserte Version SCANv3 spezifiziert. SCANv3 bietet einen verbesserten Schutz vor Replay-Angriffen. Zudem wird auf die Verwendung von Zeitstempeln verzichtet, so dass der aufwändige Einsatz von synchronen Uhren auf den Sensorknoten entfallen kann. Das Problem der unsicheren Gruppenschlüssel in SCANv2 wird durch die Generierung und Verteilung neuer Gruppenschlüssel nach der Teilung einer Zone gelöst. Des Weiteren wurde ein neues Verfahren zur Behandlung von Knotenausfällen skizziert, das mit Hilfe eines Schlüsselaustauschverfahren dynamisch Schlüssel zwischen den Nachbarn der ausgefallenen Zone etabliert und dadurch die Sicherheit des Verfahrens verbessert. Durch Umstellung und Zusammenfassung einzelner Protokollschritte ist insgesamt der Protokollablauf von SCANv3 im Vergleich zur Vorgängerversion übersichtlicher und weniger aufwändig.

Ein wesentlicher Bestandteil der Arbeit war die Spezifikation des eigentlichen Protokolls zum sicheren Einfügen und Auffinden von Diensten im Overlay. Das Dienstverzeichnis SCANv3 verwendet die Schlüsselaustauschprotokolle SPX und MPX um zwischen zwei Overlayknoten einen gemeinsamen symmetrischen Sitzungsschlüssel zu etablieren. Durch den Sitzungsschlüssel kann sich der Verwalter einer Zone gegenüber einem anfragenden Knoten authentisieren und die weitere Kommunikation mit dem anfragenden Knoten durch Verschlüsselung schützen. Die eigentliche Übermittlung der Diensteinträge kann dadurch effizient außerhalb des Overlays erfolgen.

Die verbesserte Version des Dienstverzeichnisses wurde im Simulator GloMoSim implementiert und evaluiert. Die Implementierung beinhaltet die Simulation eines An-

greifers, der Sensorknoten übernimmt und dadurch möglichst hohen Schaden verursacht. Auf diese Weise kann die Effizienz der verbesserten Sicherheitsarchitektur in einem Netz mit simulierten Angreifern beurteilt werden. Anhand der Simulationsergebnisse wurde gezeigt, dass die vorgeschlagenen Verfahren geeignet sind um in einem Netz mit bösartigen Knoten das Einfügen und Auffinden von Diensten zu ermöglichen. Der Anteil der erfolgreichen Lookup-Operationen, die in einem simulierten Sensornetz mit einem bestimmten Anteil bösartiger Knoten erzielt werden entspricht dabei weitgehend den Erwartungen, die sich aufgrund der theoretischen Überlegungen zur Sicherheit der Verfahren ergeben. Das erzielte Sicherheitsniveau von SCANv3 kann durch eine Reihe von Parametern an die Erfordernisse eines bestimmten Anwendungsszenarios angepasst werden. Ein höheres Maß an Sicherheit wird dabei durch einen höheren Kommunikationsaufwand beim Einfügen und Auffinden der Dienste erkauft.

7.2 Ausblick

Die Sicherheitsarchitektur von SCANv3 ist bisher unabhängig von den Sicherheitsmerkmalen des Underlays. Durch eine schichtenübergreifende Sicherheitsarchitektur könnte der Aufwand, der pro Schicht durch kryptographische Operationen und die Verwaltung von Schlüsseln entsteht, reduziert werden. Dies erfordert den Vergleich möglicher sicherer Routing- und Transportprotokolle für Sensornetze hinsichtlich deren Sicherheitsmerkmale.

Des Weiteren konnte im Rahmen dieser Arbeit nicht abschließend geklärt werden, ob die vorgeschlagenen Verfahren zur Behandlung von Knotenausfällen geeignet sind, um auf effiziente und sichere Weise die Struktur des Overlays zu erhalten. In weiteren Arbeiten müssen daher verschiedene Ansätze zur Lösung dieses Problems hinsichtlich der Sicherheit und des Kommunikationsaufwands miteinander verglichen und beurteilt werden.

A. Weitere Simulationsergebnisse

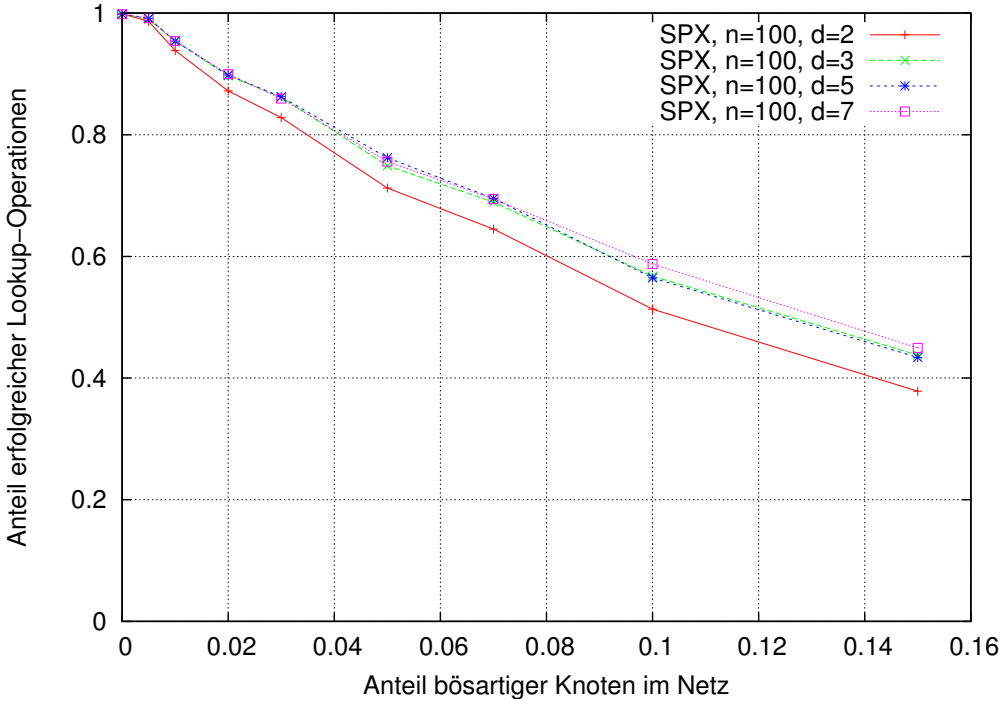

Abbildung A.1: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem Netz mit n=100 Knoten, falls zum Einfügen der Dienste SPX verwendet wird.

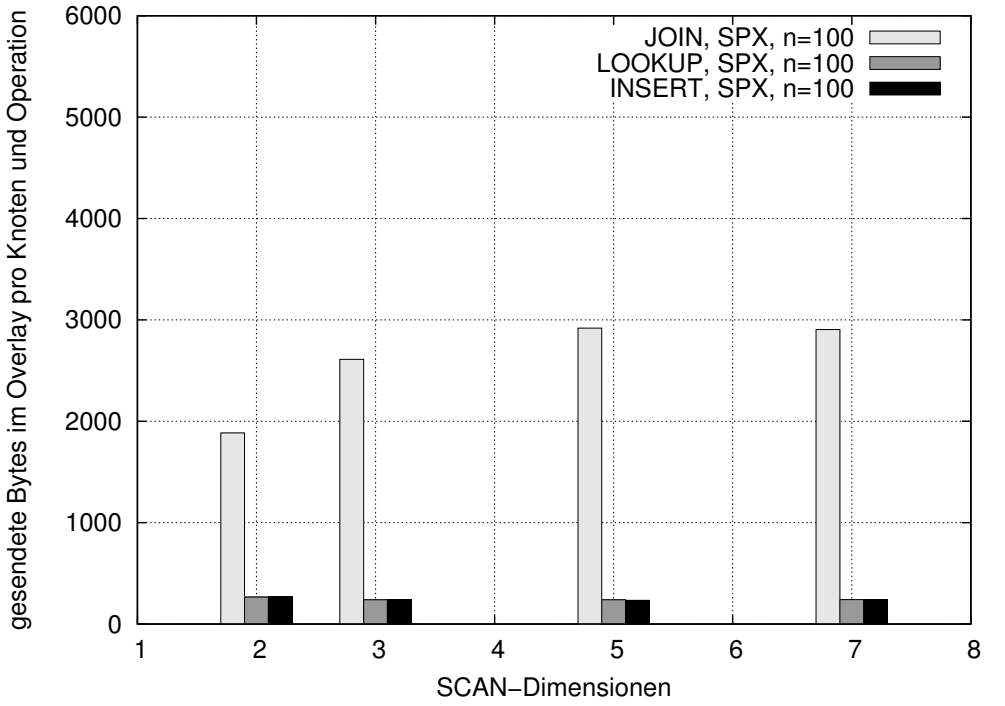

Abbildung A.2: Kommunikationsaufwand für eine Join-, Lookup- und Insert-Operation pro Overlayknoten in einem Netz mit n=100 Knoten, falls zum Einfügen der Dienste SPX verwendet wird.

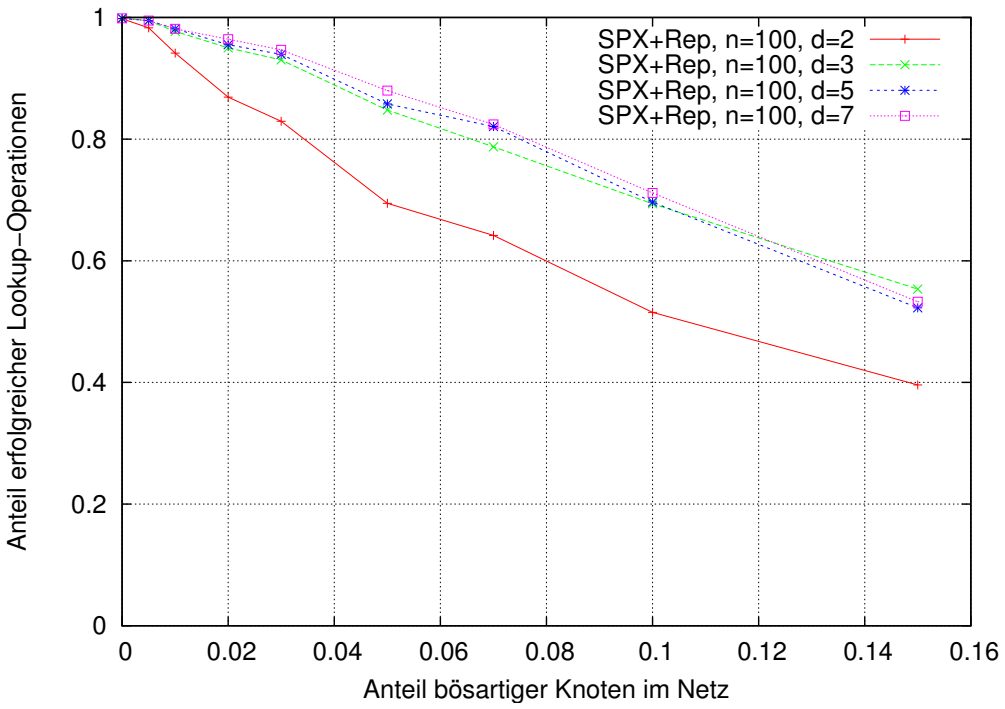

Abbildung A.3: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem Netz mit n=100 Knoten, falls zum Einfügen der Dienste SPX mit Replikaten verwendet wird.

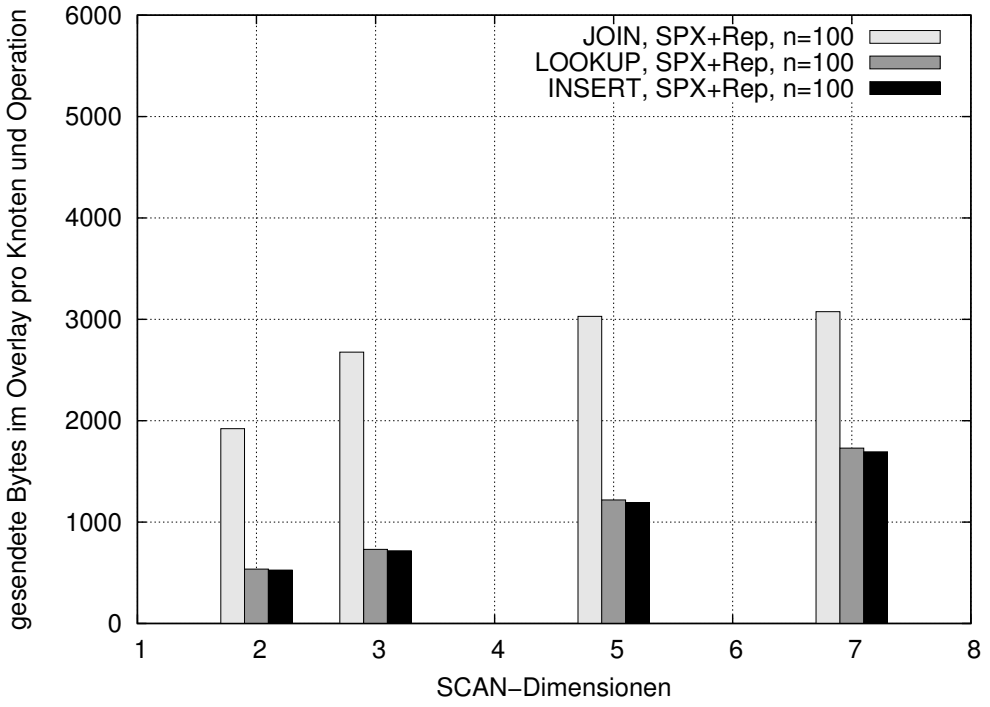

Abbildung A.4: Kommunikationsaufwand für eine Join-, Lookup- und Insert-Operation pro Overlayknoten in einem Netz mit n=100 Knoten, falls zum Einfügen der Dienste SPX mit Replikaten verwendet wird.

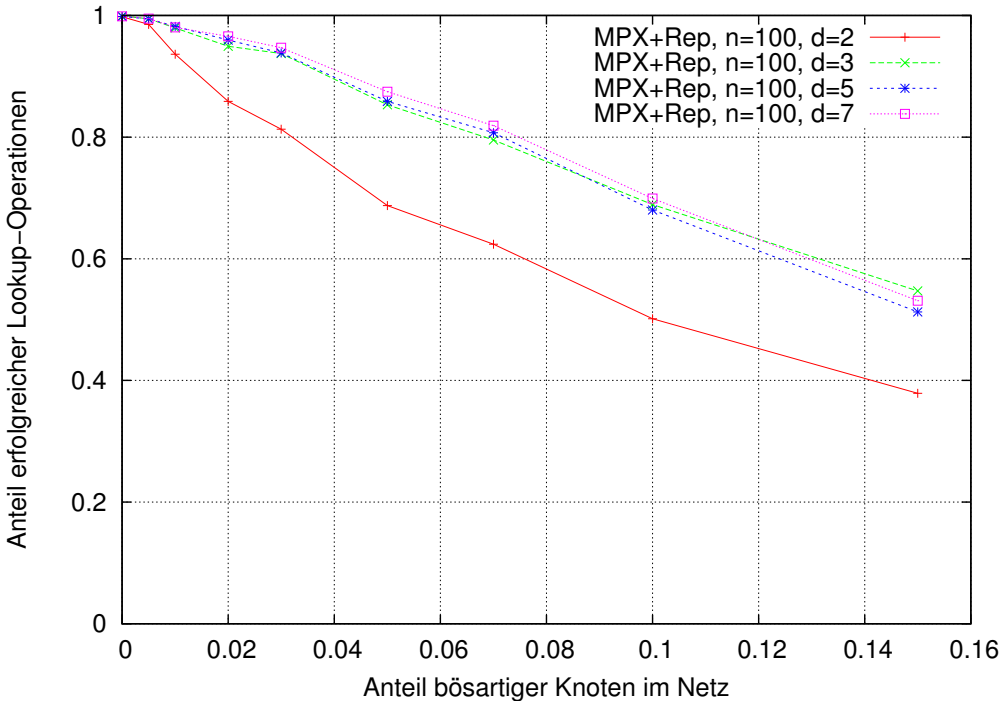

Abbildung A.5: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem Netz mit n=100 Knoten, falls zum Einfügen der Dienste MPX mit Replikaten verwendet wird

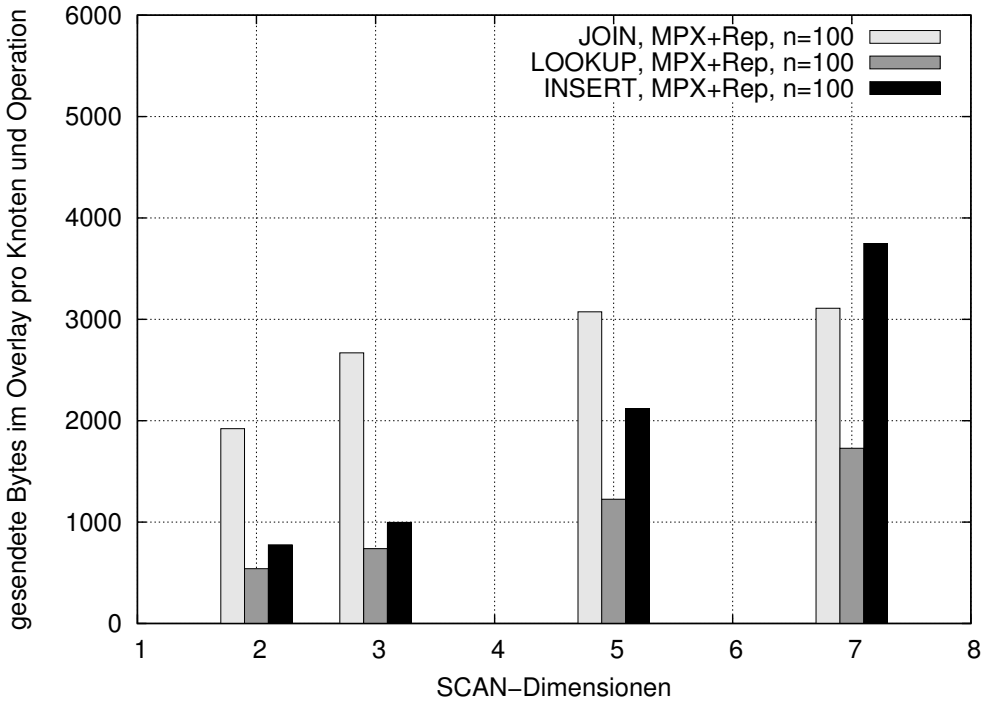

Abbildung A.6: Kommunikationsaufwand für eine Join-, Lookup- und Insert-Operation pro Overlayknoten in einem Netz mit n=100 Knoten, falls zum Einfügen der Dienste MPX mit Replikaten verwendet wird.

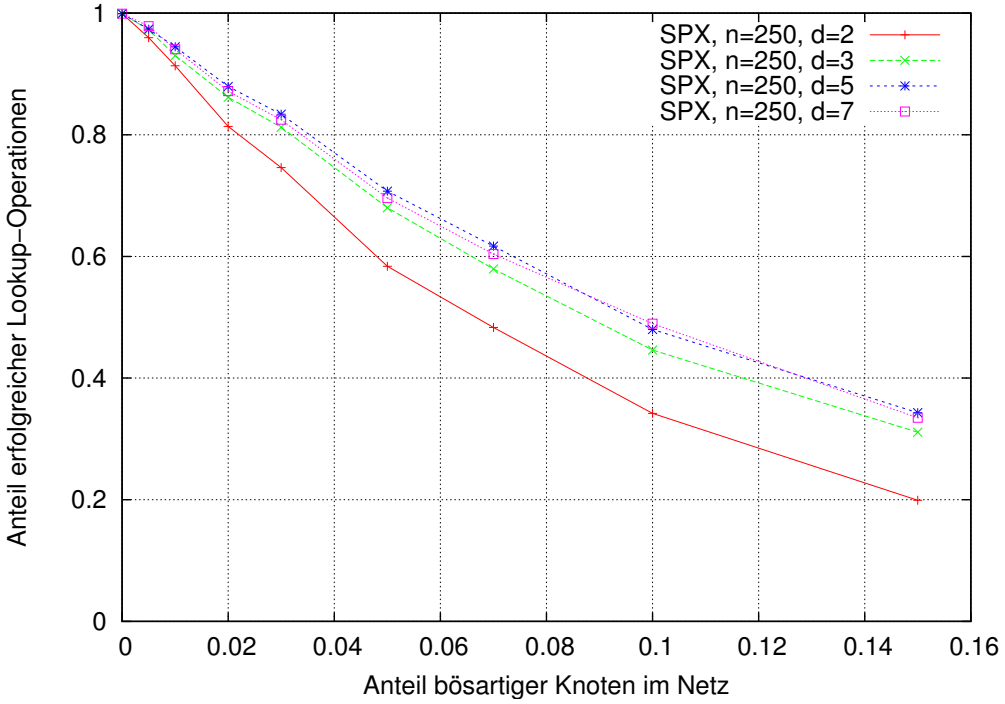

Abbildung A.7: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem Netz mit n=250 Knoten, falls zum Einfügen der Dienste SPX verwendet wird

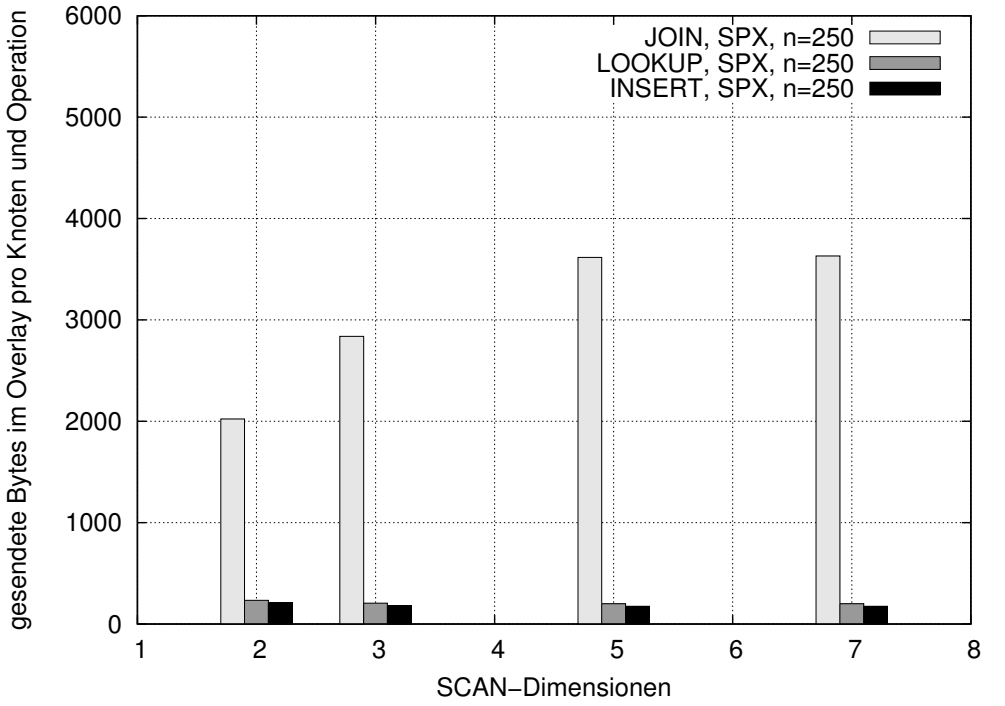

Abbildung A.8: Kommunikationsaufwand für eine Join-, Lookup- und Insert-Operation pro Overlayknoten in einem Netz mit n=250 Knoten, falls zum Einfügen der Dienste SPX verwendet wird.

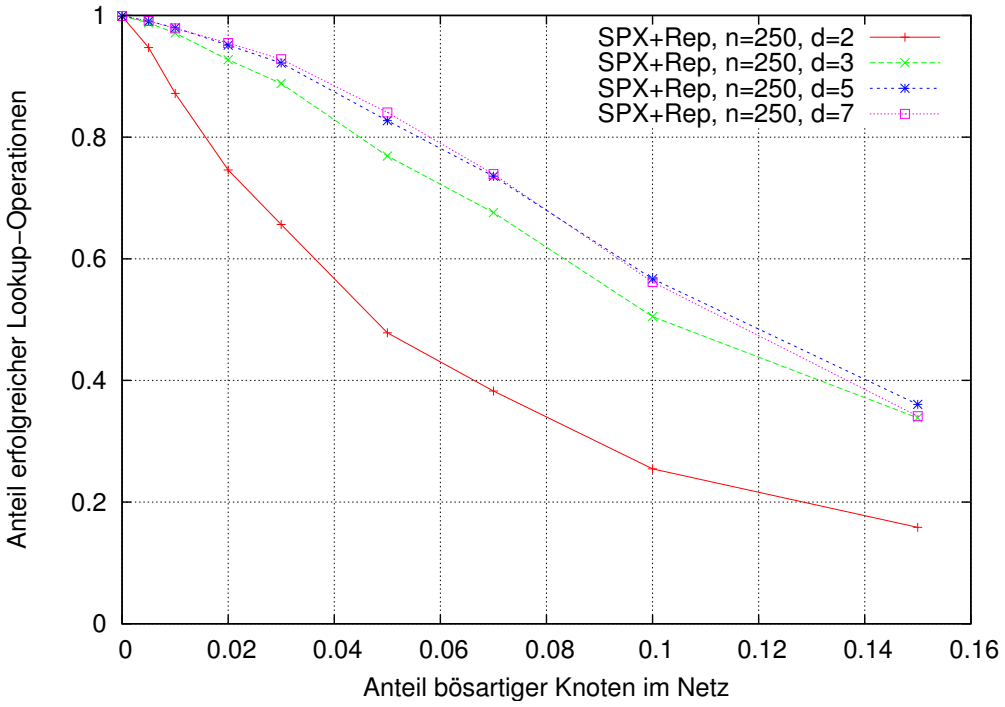

Abbildung A.9: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem Netz mit n=250 Knoten, falls zum Einfügen der Dienste SPX mit Replikaten verwendet wird

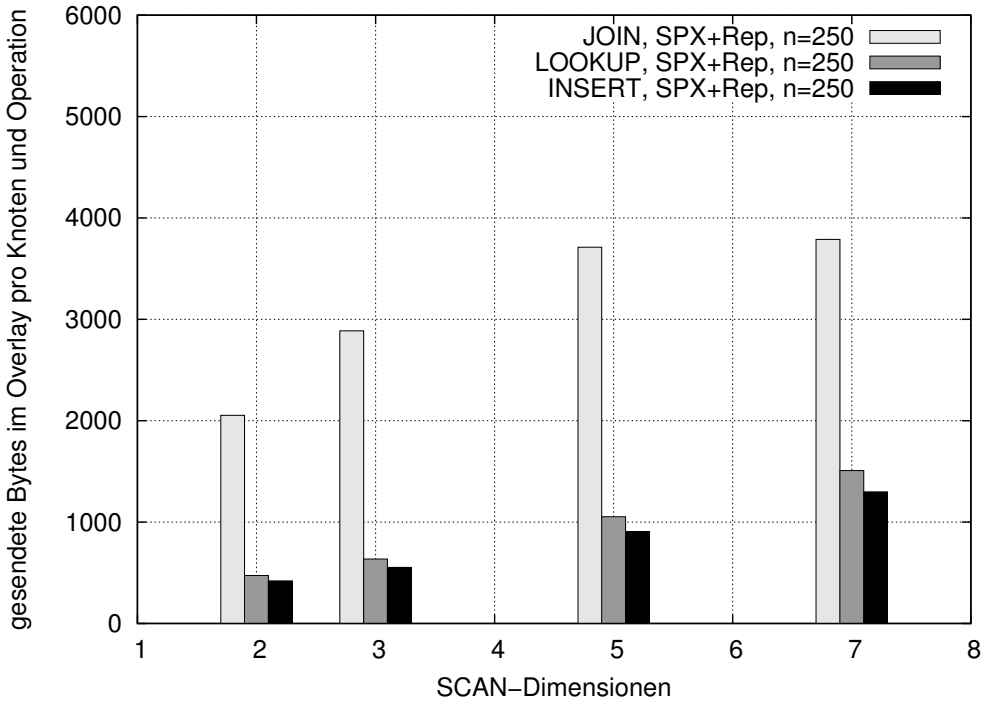

Abbildung A.10: Kommunikationsaufwand für eine Join-, Lookup- und Insert-Operation pro Overlayknoten in einem Netz mit n=250 Knoten, falls zum Einfügen der Dienste SPX mit Replikaten verwendet wird.

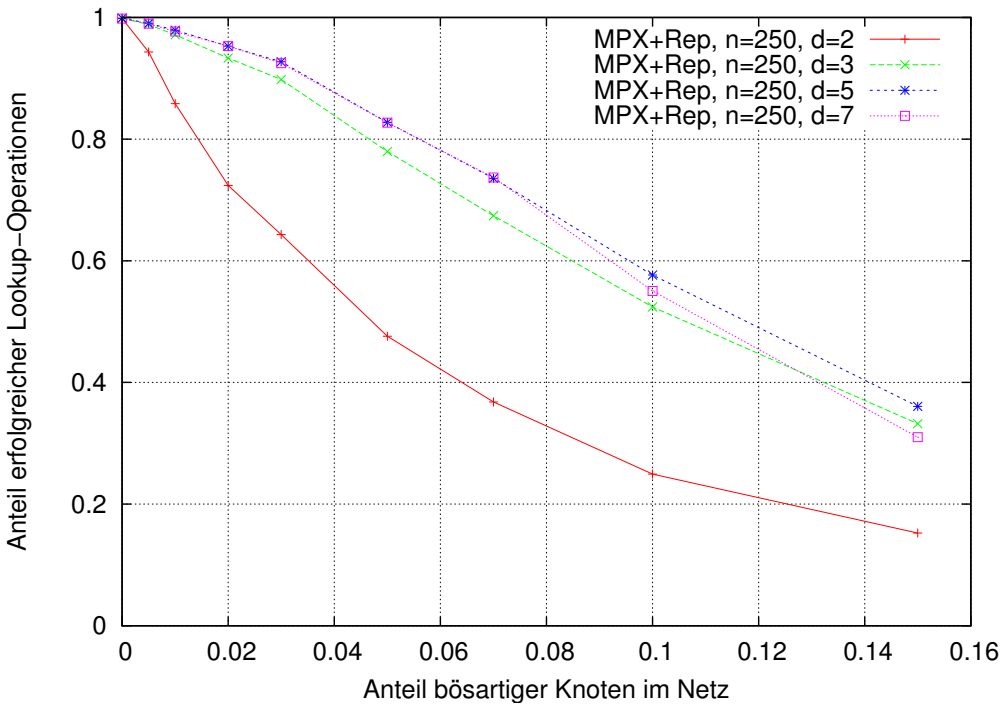

Abbildung A.11: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem Netz mit n=250 Knoten, falls zum Einfügen der Dienste MPX mit Replikaten verwendet wird

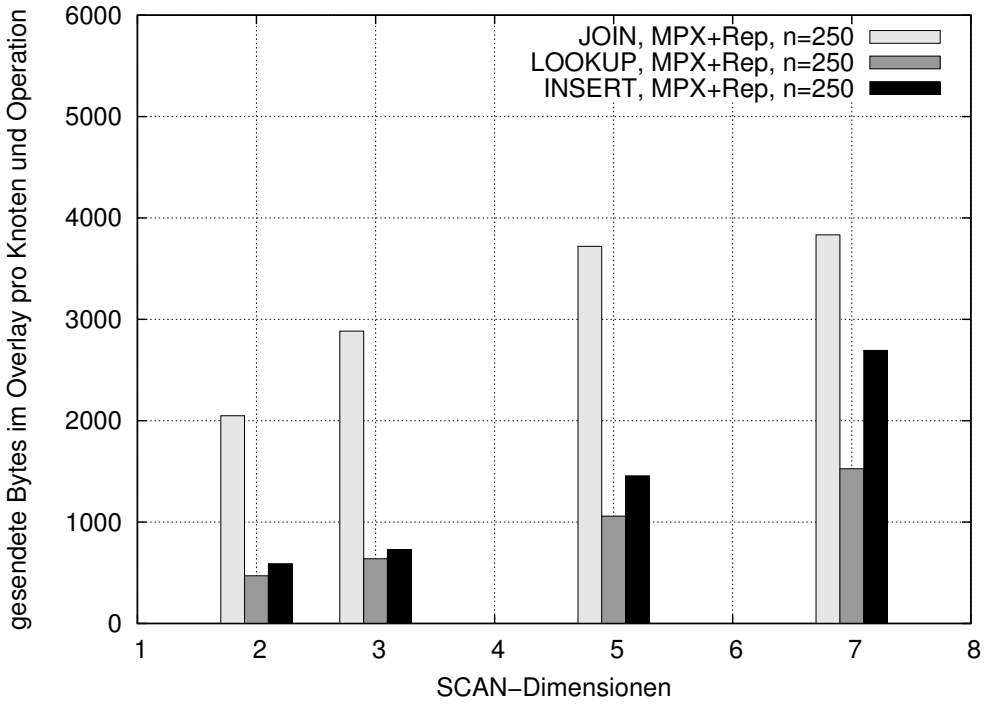

Abbildung A.12: Kommunikationsaufwand für eine Join-, Lookup- und Insert-Operation pro Overlayknoten in einem Netz mit n=250 Knoten, falls zum Einfügen der Dienste MPX mit Replikaten verwendet wird.

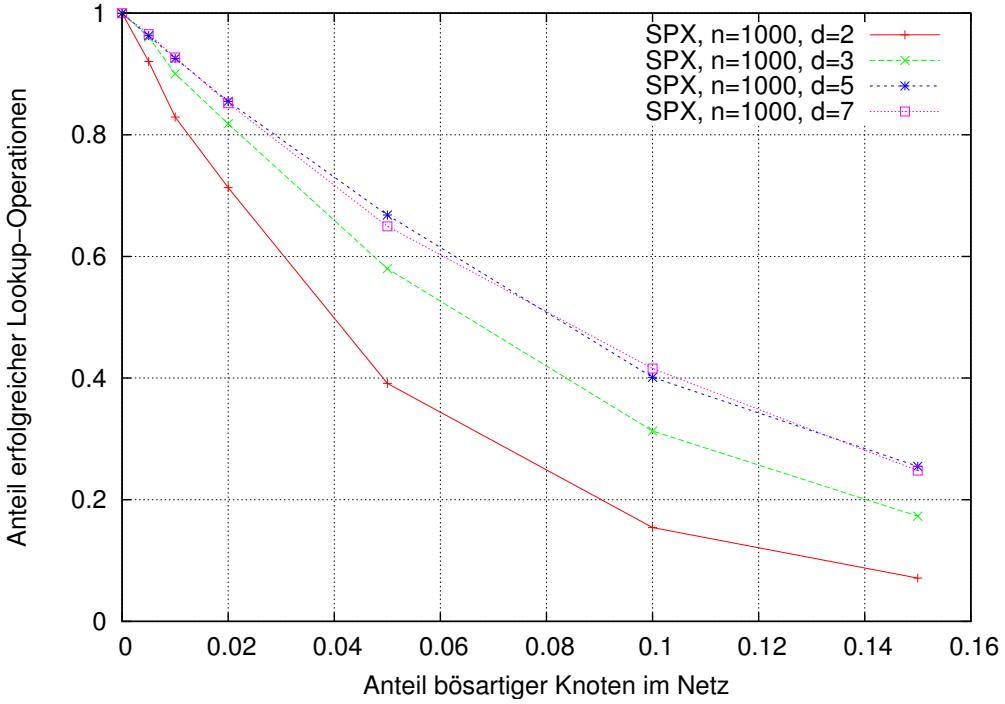

Abbildung A.13: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem Netz mit n=1000 Knoten, falls zum Einfügen der Dienste SPX verwendet wird

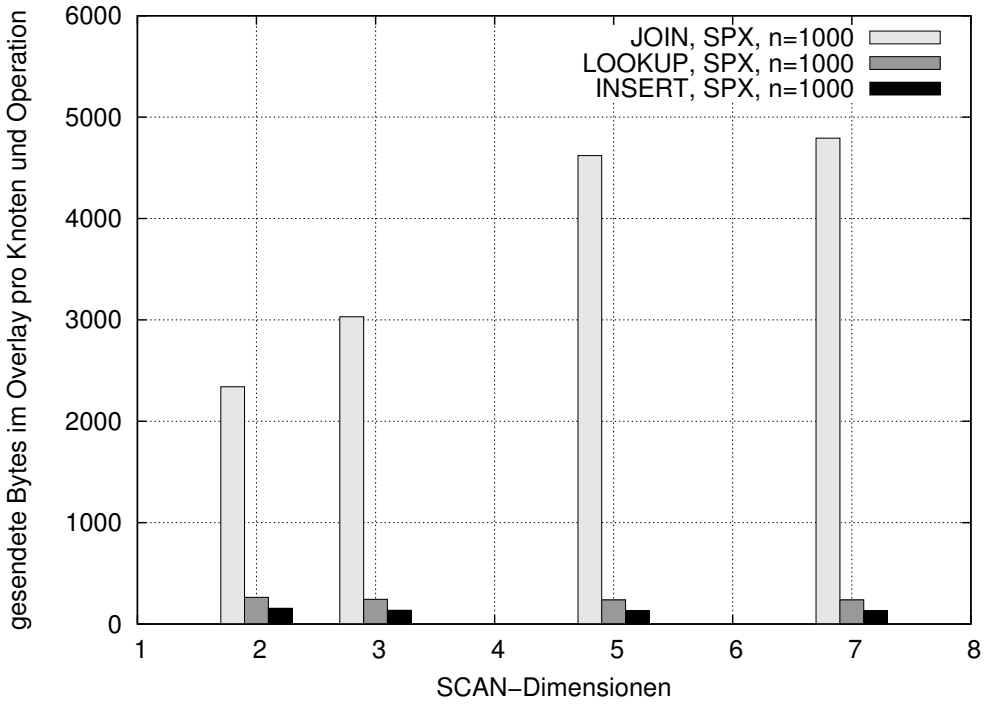

Abbildung A.14: Kommunikationsaufwand für eine Join-, Lookup- und Insert-Operation pro Overlayknoten in einem Netz mit n=1000 Knoten, falls zum Einfügen der Dienste SPX verwendet wird.

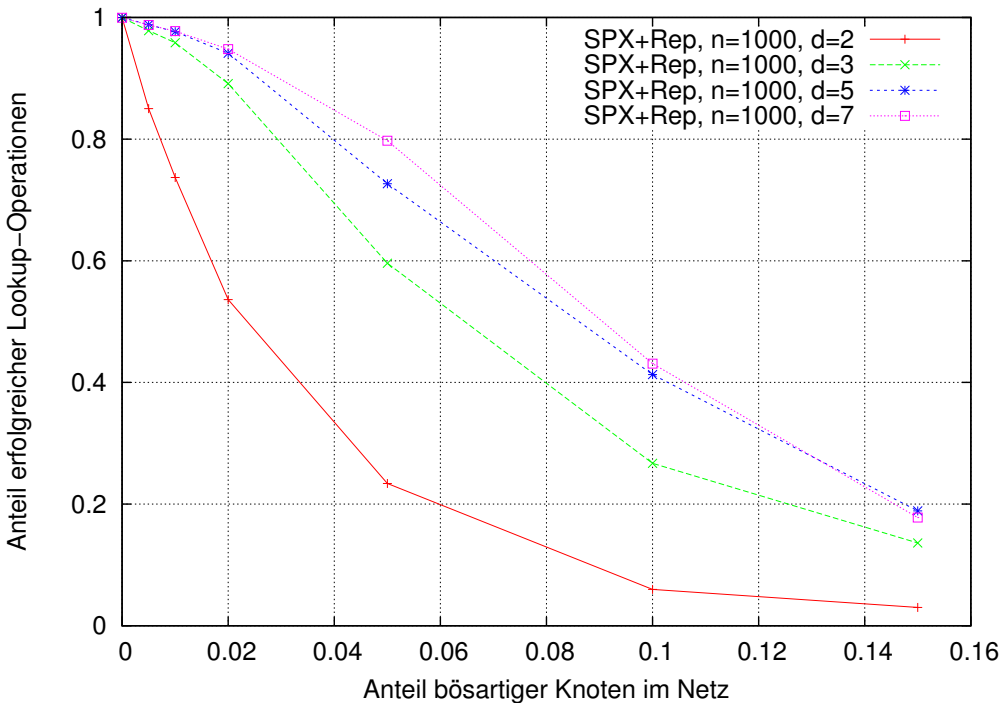

Abbildung A.15: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem Netz mit n=1000 Knoten, falls zum Einfügen der Dienste SPX mit Replikaten verwendet wird

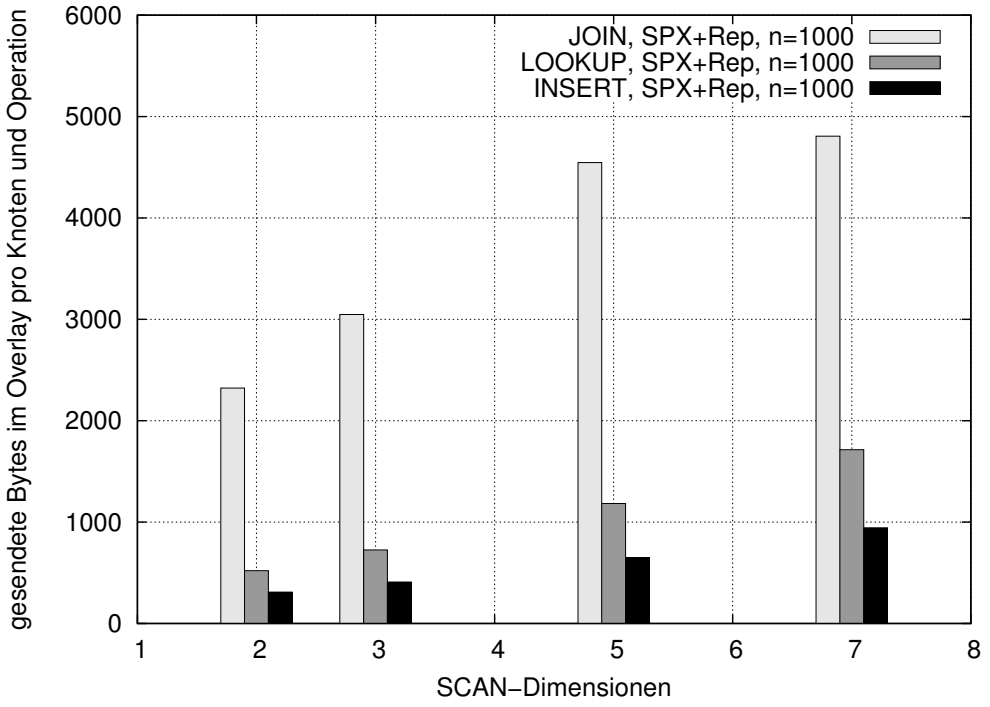

Abbildung A.16: Kommunikationsaufwand für eine Join-, Lookup- und Insert-Operation pro Overlayknoten in einem Netz mit n=1000 Knoten, falls zum Einfügen der Dienste SPX mit Replikaten verwendet wird.

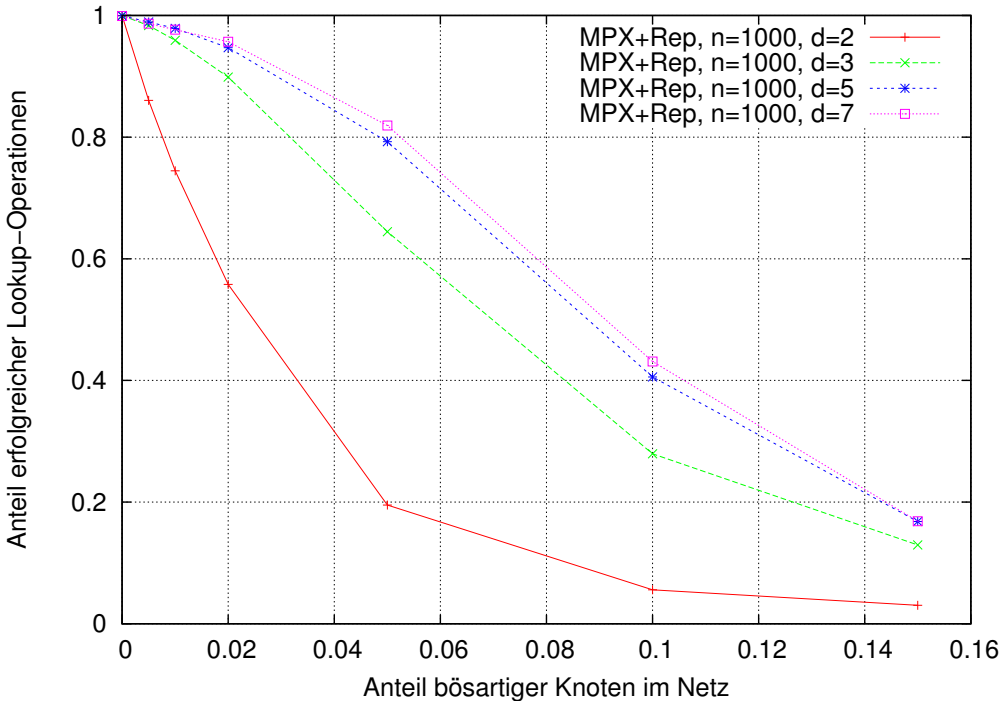

Abbildung A.17: Anteil der erfolgreich durchgeführten Lookup-Operationen in einem Netz mit n=1000 Knoten, falls zum Einfügen der Dienste MPX mit Replikaten verwendet wird

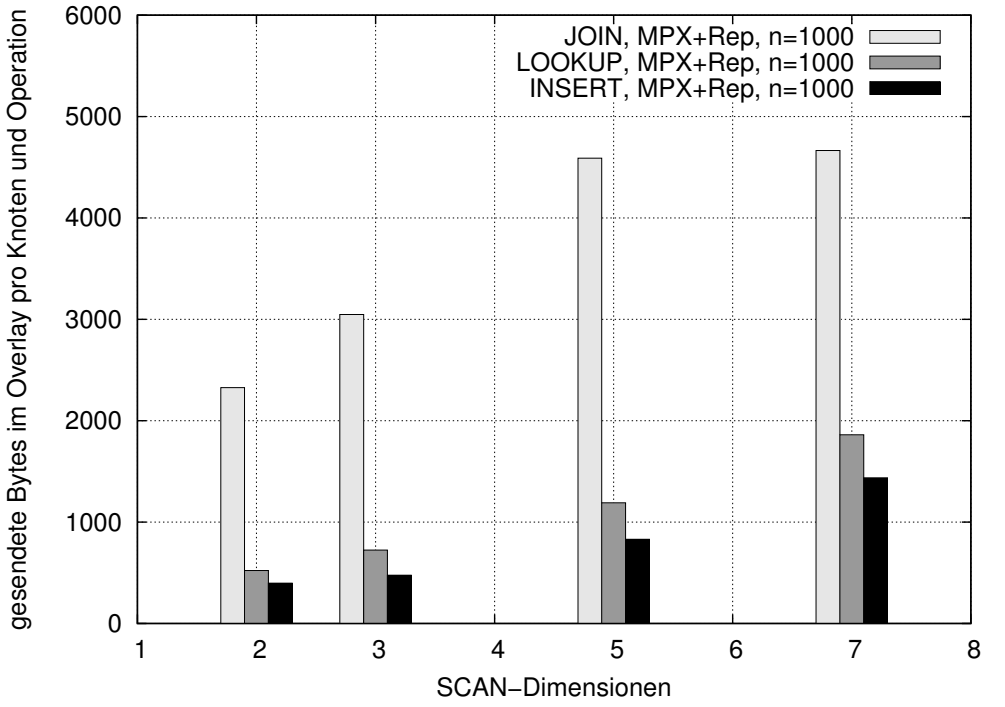

Abbildung A.18: Kommunikationsaufwand für eine Join-, Lookup- und Insert-Operation pro Overlayknoten in einem Netz mit n=1000 Knoten, falls zum Einfügen der Dienste MPX mit Replikaten verwendet wird.

Literatur

[AlCa03] Florina Almenárez und Celeste Campo. SPDP: A Secure Service Discovery Protocol for Ad-hoc Networks. In *9th Open European Summer School and IFIP Workshop on Next Generation Networks, Budapest*, 2003.

[BeRe00] C. Bettstetter und C. Renner. A Comparison of Service Discovery Protocols and Implementation of the Service Location Protocol. In *Proceedings of the Sixth EUNICE Open European Summer School: Innovative Internet Applications*, 2000.

[BMTC+98] Rajive Bagrodia, Richard Meyer, Mineo Takai, Yu an Chen, Xiang Zeng, Jay Martin und Ha Yoon Song. Parsec: A Parallel Simulation Environment for Complex Systems. *Computer* 31(10), 1998, S. 77–85.

[BNWA+03] B. Blum, P. Nagaraddi, A. Wood, T. Abdelzaher, S. Son und J. Stankovic. An Entity Maintenance and Connection Service for Sensor Networks, 2003.

[CDGR+02] Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron und Dan S. Wallach. Secure routing for structured peer-to-peer overlay networks. *SIGOPS Oper. Syst. Rev.* 36(SI), 2002, S. 299–314.

[ChPS03] Haowen Chan, Adrian Perrig und Dawn Song. Random Key Predistribution Schemes for Sensor Networks. In *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*, Washington, DC, USA, 2003. IEEE Computer Society, S. 197.

[CJLM+01] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum und L. Viennot. Optimized Link State Routing Protocol. In *IEEE INMIC Pakistan 2001*, 2001.

[DiHe76] Whitfield Diffie und Martin E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory* IT-22(6), 1976, S. 644–654.

[Douc02] John R. Douceur. The Sybil Attack. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, London, UK, 2002. Springer-Verlag, S. 251–260.

[EsGl02] Laurent Eschenauer und Virgil D. Gligor. A key-management scheme for distributed sensor networks. In *CCS '02: Proceedings of the*

9th ACM conference on Computer and communications security, New York, NY, USA, 2002. ACM Press, S. 41–47.

[Gama85] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in cryptology*, New York, NY, USA, 1985. Springer-Verlag New York, Inc., S. 10–18.

[GeRR98] Rosario Gennaro, Michael O. Rabin und Tal Rabin. Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In *PODC '98: Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, New York, NY, USA, 1998. ACM Press, S. 101–111.

[Gutt99] Erik Guttman. Service Location Protocol: Automatic Discovery of IP Network Services. *IEEE Internet Computing* 3(4), 1999, S. 71–80.

[HiCu02] Jason L. Hill und David E. Culler. Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro* 22(6), 2002, S. 12–24.

[HoBZ04] H.-J. Hof, E.-O. Blaß und M. Zitterbart. Secure Overlay for Service Centric Wireless Sensor Networks. ESAS 2004, 2004.

[HSLA03] Tian He, John A. Stankovic, Chenyang Lu und Tarek Abdelzaher. SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks. In *ICDCS '03: Proceedings of the 23rd International Conference on Distributed Computing Systems*, Washington, DC, USA, 2003. IEEE Computer Society, S. 46.

[HuHZ04] Bernhard Hurler, Hans-Joachim Hof und Martina Zitterbart. A General Architecture for Wireless Sensor Networks: First Steps. In *4th International Workshop on Smart Appliances and Wearable Computing*, Tokyo Japan, Mar 2004. S. 442–444.

[JoMa96] David B. Johnson und David A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Thomasz Imielinski und Hank Korth (Hrsg.), *Mobile Computing*, Band 353, Kapitel 5, S. 153–181. Kluwer Academic Publishers, 1996.

[KaSW04] Chris Karlof, Naveen Sastry und David Wagner. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. In *Second ACM Conference on Embedded Networked Sensor Systems (SensSys 2004)*, November 2004.

[KaWa03] Chris Karlof und David Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. In *First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003, S. 113–127.

[LiNi03] Donggang Liu und Peng Ning. Establishing pairwise keys in distributed sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, New York, NY, USA, 2003. ACM Press, S. 52–61.

[MaRö03] Friedemann Mattern und Kay Römer. Drahtlose Sensornetze. *Informatik-Spektrum* 26(3), Juni 2003, S. 191–194.

[MaWS04] David J. Malan, Matt Welsh und Michael D. Smith. A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography. In *First IEEE International Conference on Sensor and Ad Hoc Communications and Networks SECON04*, Santa Clara, California, October 2004.

[MCPS+02] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk und John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM Press, 2002, S. 88–97.

[MeOV01] Alfred J. Menezes, Paul C. van Oorschot und Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press. http://www.cacr.math.uwaterloo.ca/hac/, 2001.

[Nuev03] Jorge Nuevo. A Comprehensible GloMoSim Tutorial, 2003. http://externe.inrs-emt.uquebec.ca/users/nuevo/glomoman.pdf.

[Nuev04] Jorge Nuevo. A Hierarchical Service Distribution Architecture for Ad Hoc Networks based on the Weighted Clustering Algorithm, 2004. http://externe.inrs-emt.uquebec.ca/users/nuevo/glomopatch.htm/.

[PeRo99] Charles E. Perkins und Elizabeth M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, Washington, DC, USA, 1999. IEEE Computer Society, S. 90.

[PeSW04] Adrian Perrig, John Stankovic und David Wagner. Security in wireless sensor networks. *Commun. ACM* 47(6), 2004, S. 53–57.

[PlFo03] Universal Plug und Play Forum. UPnP Device Architecture 1.0, Version 1.01, 2003. http://www.upnp.org/.

[PSWC+01] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler und J. D. Tygar. SPINS: security protocols for sensor netowrks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, New York, NY, USA, 2001. ACM Press, S. 189–199.

[RFHK+01] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp und Scott Schenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, New York, NY, USA, 2001. ACM Press, S. 161–172.

[RiSA83] R. L. Rivest, A. Shamir und L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* 26(1), 1983, S. 96–99.

[RoDr01] Antony Rowstron und Peter Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. *Lecture Notes in Computer Science* Band 2218, 2001.

[ScLR] Jochen Schiller, Achim Liers und Hartmut Ritter. Embedded Sensor Board ESB. Webseite des Scatterweb-Projekts: http://www.scatterweb.net/.

[Sham79] Adi Shamir. How to share a secret. *Commun. ACM* 22(11), 1979, S. 612–613.

[SIG99] Bluetooth SIG. Specification of the Bluetooth System - Core, Version 1.0B volume 1, 1999. http://www.bluetooth.org/.

[SiMo02] Emil Sit und Robert Morris. Security Considerations for Peer-to-Peer Distributed Hash Tables. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*. Springer-Verlag, 2002, S. 261–269.

[SMLNK+03] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek und Hari Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. Netw.* 11(1), 2003, S. 17–32.

[SrLi04] Mudhakar Srivatsa und Ling Liu. Vulnerabilities and Security Threats in Structured Overlay Networks: A Quantitative Analysis. In *Proceedings of the 20th Annual Computer Security Applications Conference*, Tucson, Arizona, USA, 2004.

[StAn00] Frank Stajano und Ross J. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *Proceedings of the 7th International Workshop on Security Protocols*, London, UK, 2000. Springer-Verlag, S. 172–194.

[WLLP01] Brett Warneke, Matt Last, Brian Liebowitz und Kristofer S. J. Pister. Smart Dust: Communicating with a Cubic-Millimeter Computer. *Computer* 34(1), 2001, S. 44–51.

[WoSt02] Anthony D. Wood und John A. Stankovic. Denial of Service in Sensor Networks. *Computer* 35(10), 2002, S. 54–62.

[ZeBG98] Xiang Zeng, Rajive Bagrodia und Mario Gerla. GloMoSim: a library for parallel simulation of large-scale wireless networks. *SIGSIM Simul. Dig.* 28(1), 1998, S. 154–161.

[ZhKJ01] B. Y. Zhao, J. D. Kubiatowicz und A. D. Joseph. Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing. Technischer Bericht UCB/CSD-01-1141, UC Berkeley, April 2001.