Michael Roßberg

Skalierbare Autokonfiguration sabotageresistenter virtueller privater Netze

Cuvillier Verlag Göttingen
Internationaler wissenschaftlicher Fachverlag

Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Fachgebiet Telematik / Rechnernetze

Dissertation

zur Erlangung des Akademischen Grades
Doktoringenieur (Dr.-Ing.)

Skalierbare Autokonfiguration sabotageresistenter virtueller privater Netze

---

vorgelegt von: Michael Roßberg
geboren am: 27. Mai 1981 in Karlsburg

Datum der Einreichung: 24. November 2010
Datum der Verteidigung: 18. Mai 2011

Betreuender Gutachter: Prof. Dr.-Ing. Günter Schäfer
Externer Gutachter: Prof. Dr.-Ing. Phuoc Tran-Gia
Externer Gutachter: Prof. Dr. Günter Müller

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.
1. Aufl. - Göttingen : Cuvillier, 2011
Zugl.: (TU) Ilmenau, Univ., Diss., 2011

978-3-86955-775-5

© CUVILLIER VERLAG, Göttingen 2011
Nonnenstieg 8, 37075 Göttingen
Telefon: 0551-54724-0
Telefax: 0551-54724-21
www.cuvillier.de

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen.
1. Auflage, 2011
Gedruckt auf säurefreiem Papier

978-3-86955-775-5

Zusammenfassung

Ein schneller und sicherer Informationsfluss hat sich in den letzten Jahren immer mehr zum Hauptkriterium für den Erfolg von Firmen, Behörden und Organisationen herauskristallisiert. Um dafür öffentliche Netze zu nutzen, werden zur Realisierung von Vertraulichkeit, Authentizität und Integrität in der Regel virtuelle private Netze (VPN) betrieben. Deren manuelle Konfiguration und Betrieb ist jedoch aufwändig und anfällig für potentielle Fehler.
Dieser Umstand hat zwar zur Entwicklung zahlreicher Autokonfigurationsansätze geführt, die allerdings bisher weder Sabotageresistenz noch Robustheit adressieren. Die wenigen skalierbaren Systeme erfüllen selbst einfache funktionale Anforderungen nicht, wie etwa den Einsatz privater IP-Adressen innerhalb des VPN. Zwei Ansätzen konnten sogar eklatante Sicherheitsmängel nachgewiesen werden.
Ausgehend davon ist im Rahmen dieser Arbeit ein neuartiges Konzept für einen IPsec-basierten VPN-Autokonfigurationsansatz entstanden. Durch den Einsatz von Peer-to-Peer-Prinzipien und den damit verbundenen Verzicht auf exponierte Systeme stellt er die Grundlage für ein skalierbares und hochgradig verfügbares System dar. Wesentlich ist dabei die Unterstützung indirekt angebundener VPN-Teilnehmer, also solcher, die ausschließlich über andere erreichbar sind. Dabei werden die Vertraulichkeit, die Integrität und die Authentizität der übertragenen Daten immer Ende-zu-Ende garantiert, und es wird in Bezug auf Verfügbarkeit beispielsweise eine Umleitung von Verkehr im Falle partieller Kommunikationsstörungen über dritte Standorte erlaubt. Durch den Autokonfigurationsmechanismus ist es ferner möglich, auf Sabotageangriffe durch Veränderung der Infrastruktur schnell zu reagieren, indem beispielsweise die externen IP-Adressen betroffener Systeme gewechselt werden. Beim Entwurf des Systems konnte gezeigt werden, dass auch ohne den Broadcast von Routing-Informationen in gebräuchlichen Transportnetzen optimale Pfade gefunden werden, und wie bei der Einbettung strukturierter Overlay-Netze in diese Transportnetze Inkonsistenzen und Partitionierungen vermieden werden können.
Durch die Möglichkeit indirekte Sicherheitsbeziehungen zu konfigurieren, können direkte Kommunikationsvorgänge nahezu beliebig administrativ erlaubt oder verboten werden. Erfolgt so ein Verzicht auf direkte Sicherheitsbeziehungen zu gefährdeten Systemen, wird es möglich, die nach außen sichtbaren IP-Adressen bestimmter VPN-Knoten zu verbergen, und sie so effektiv vor DoS-Angriffen zu schützen. Die analytische Konstruktion von optimal sabotageresistenten Topologien erwies sich im allgemeinen Fall als NP-schwer. Daher ist im Rahmen der Arbeit ein System entwickelt worden, welches mit geringem administrativen Aufwand dennoch Topologien mit nachweisbaren Sicherheitseigenschaften konstruieren kann. Dabei wird unter anderem jedem VPN-Knoten eine Verfügbarkeitszone zugeordnet und nur zwischen bestimmten Zonen eine direkte Kommunikation erlaubt.

Das Gesamtkonzept wird mithilfe einer umfangreichen Implementierung evaluiert, welche plattformunabhänige Teile so kapselt, dass eine parallele Untersuchung mittels Simulationen und in einem Prototyp erfolgen kann. Auch die Implementierung geht, beispielsweise durch die Einführung von IP-Adressattestationen zur Sicherung gegen interne Angreifer, über den Stand der Technik hinaus.

Abstract

Over the last years, a fast and secure flow of information has crystalized as major criterion for the success of companies, authorities, and other organizations. In order to exploit untrustworthy networks for this, virtual private networks (VPNs) are deployed to guarantee data confidentiality, authentication, and integrity. Nonetheless, the associated manual deployment and operation is time-consuming and error-prone.
This led to the development of numerous proposals for VPN auto-configuration, but none of them addresses DoS resilience or robustness. The few scalable systems do not fulfill basic objectives, such as the use of private IP address ranges within the VPN. Two approaches were even shown to have significant security issues.
Thus, within this thesis a novel concept for the automatic configuration of IPsec infrastructures has been developed. By utilizing peer-to-peer principles and hence relinquishing exposed central systems, the approach provides a basis for scalable and highly available VPNs. A key feature is the integration of indirectly connected VPN nodes, i.e., nodes that are only accessible via others, while guaranteeing end-to-end data confidentiality, integrity, as well as authentication, at the same time. This allows for redirecting traffic flows over a third VPN node in the case of partial network outages, for example. Furthermore, due to the auto-configuration aspect it is possible to quickly react to DoS attacks, e.g., by reintegrating affected nodes with different IP addresses. As a result of this thesis, it has been shown how – at least within common substrate networks – optimal forwarding paths can be found within VPNs without utilizing broadcast for routing, and how inconsistencies and partitions can be avoided when embedding structured overlay networks into these substrate networks.
Additionally, the possibility to deploy indirect security associations allows administrators to artificially reduce the number of direct connections as the nodes will still be able to communicate via others. This can be utilized to reduce the number of outer IP addresses that can be gathered by an attacker, e.g., by wiretapping, and hence reduce the number of possible targets for DoS attacks. And while the construction of optimally resilient topologies can be shown to be NP hard, heuristics are presented to still allow for planning topologies with calculable risks. In doing so, VPN nodes are classified into so-called availability zones and direct associations will only be allowed between neighboring zones.
The overall concept is evaluated by an extensive implementation, which can be deployed within a network simulator or – utilizing the same code – within a real prototype. Also the implementation extends the state of the art, e.g., by introducing cryptographic address attestations for VPN nodes in order to cope with internal attackers.

Coram iudice et in alto mare in manu dei soli sumus.

(Römische Weisheit)

Danksagung

Dass diese Arbeit in dieser Form so erfolgreich abgeschlossen wurde, ist sicherlich nicht nur mein Verdienst, sondern auch der von Prof. Dr.-Ing. Günter Schäfer, der als mein Betreuer immer wieder offene Probleme und lohnenswerte Zwischenziele identifiziert hat. In diesem Kontext bin ich auch den zwei weiteren Gutachtern Prof. Dr.-Ing. Phuoc Tran-Gia und Prof. Dr. Günter Müller unmittelbar zu Dank verpflichtet, die sich nicht nur durch die umfangreiche Arbeit gekämpft, sondern auch hilfreiche Anmerkungen gegeben haben.

Darüber hinaus möchte ich Prof. Dr.-Ing. Thorsten Strufe und Dr. Michael Brinkmeier für ihre offenen Ohren gerade in der Anfangszeit der Arbeit danken. Aber auch meinen anderen Kollegen Martin Aumüller, Mathias Fischer, Franz Girlich, René Golembewski, Sascha Grau, Michael Grey, Mario Holbe, Anja Pölck, Dr. Jens Schreyer und Dr. Werner Horn gebührt Dank für die Zeit, die sie selbstlos mit Diskussionen und Probelesen verbracht haben.

Keine erfolgreiche Dissertation in der Informatik ohne sie: die gewissenhaften, fleißigen und klugen Studenten (kein Witz), die durch ihre Abschlussarbeiten und Hiwi-Tätigkeiten letztlich ein Projekt in die richtige Richtung treiben lassen. Mein Dank gebührt daher Wolfgang Steudel, Alexander Heinlein, Jens Dönhoff, Florian Schüttler, Thomas Köllmer, 2 $\times$ (Michael Braun), Mathias Hübscher, Daniel Stadie und Markus Trapp.

Ein (interessantes) Ergebnis der Arbeit ist außerdem eine Bestätigung der These, dass hinter jedem erfolgreichen Mann eine starke Frau stehen soll. Vielen lieben Dank Melli – insbesondere für das stoische Ertragen meiner schlechten Launen und der Zerstreutheit! Die Wurzeln der Arbeit werden sicher im Fundament der Familie gehalten: Sie hat mich immer zu Neugier und Kritik ermutigt, auch wenn es Lebenssituationen nach sich zog, die (für sie) nicht immer leicht waren.

Ich danke Euch allen, und wahrscheinlich findet Ihr an der einen oder anderen Stelle ja eine Phrase oder sprachliche Nuance, die von Euch auf die Arbeit durchgefärbt hat...

Inhaltsverzeichnis

1 Einleitung und Motivation **13**
1.1 Defizite bei der Konfiguration virtueller privater Netze (VPN) 13
1.2 Beiträge der Arbeit 16
1.3 Übersicht 16

2 Grundlagen: VPN–Konfiguration und strukturierte Peer–to–Peer–Systeme **19**
2.1 Sicherheitsziele und Bedrohungen in paketorientierten Netzen 19
2.2 VPN–Szenarien 21
2.2.1 VPN von Firmen & Nichtregierungsorganisationen 21
2.2.2 VPN von Behörden und Militär 23
2.2.3 Privat eingesetzte VPN 24
2.3 Kryptographische Protokolle zur Absicherung von VPN 24
2.3.1 SSL, TLS und DTLS 24
2.3.2 Die IPsec–Protokollfamilie 25
2.3.3 Weitere kryptographische Protokolle 27
2.4 Strukturierte Peer–to–Peer Systeme 27
2.4.1 Verteilte Hash–Tabellen 28
2.4.2 SkipNet und Skip–Graphen 29
2.4.3 Peer–to–Peer–basierter Multicast 30
2.4.4 Netzwerkkoordinatensysteme 31
2.4.5 Spezielle Angriffe auf Peer–to–Peer–Systeme 32

3 Stand von Wissenschaft und Technik **35**
3.1 Anforderungen an VPN–Autokonfigurationssysteme 35
3.1.1 Funktionale Anforderungen 35
3.1.2 Nichtfunktionale Anforderungen 37
3.1.3 Sicherheitsanforderungen 37
3.1.4 Anforderungen in Bezug auf die Verfügbarkeit 39
3.2 Existierende Systeme zur Autokonfiguration von VPN 41
3.2.1 Zentralisierte VPN–Konfigurationssysteme 42
3.2.2 Dezentrale VPN–Konfigurationssysteme 46
3.2.3 Vollständig verteilte VPN–Konfigurationssysteme 53
3.3 Overlay–Netze zur Steigerung von Robustheit und Verfügbarkeit 60
3.3.1 Robuste Overlay–Netze 60
3.3.2 Sabotageschutz von Netzwerkdiensten durch Overlay–Netze . . 61
3.4 Strukturierte Overlay–Netze für den Datentransport 61
3.4.1 Transport in Netzen mit garantierter, direkter Kommunikationsmöglichkeit 62

3.4.2 Indirekte Knotenanbindung durch Supernodes 62
3.4.3 Ansätze für Mobile Ad–hoc–Netze (MANET) 62
3.5 Fazit 64

4 Secure OverLay for IPsec Discovery (SOLID) 67
4.1 Grundlegende Prinzipien und Annahmen der VPN–Autokonfiguration 67
4.1.1 Netzstruktur 67
4.1.2 Annahmen 69
4.1.3 Kernfunktionen 69
4.1.4 Angreifermodell 70
4.2 Bootstrapping 71
4.3 Adresszuweisung 73
4.4 Discovery 74
4.4.1 Verteilte Datenbank oder direkte Abbildung 74
4.4.2 Struktur des Peer–to–Peer–Netzes 75
4.4.3 Querverbindungen auf Basis von Stichproben 76
4.5 Routing und Paketweiterleitung 82
4.5.1 Routing–Metrik 83
4.5.2 Routing–Mechanismen 84
4.5.3 Abkürzungsalgorithmen 85
4.5.4 Mechanismen zur Paketweiterleitung 89
4.5.5 Fazit 91
4.6 Topologiekontrolle 91
4.6.1 Divergenz zwischen Transportnetz– und VPN–Routing 92
4.6.2 Abhören von Kontrollinformationen zur Beschleunigung der Routing–Konvergenz 93
4.6.3 Overlay–Wiederherstellung nach Netzpartitionierungen 94
4.6.4 Anpassung an die Transportnetzumgebung 97
4.6.5 Fazit 99
4.7 Multicast–Routing 99
4.7.1 Topologien für die Verteilung von Multicast–Daten 99
4.7.2 Optimierungsmaßnahmen 101
4.8 Fazit 102

5 Betrachtungen zur Sabotageresistenz 103
5.1 Modellierung von VPN 103
5.2 Abschätzung der Bedrohung durch externe DoS–Angriffe 104
5.3 Schutz vor externen DoS–Angriffen 110
5.3.1 Optimale VPN–Topologien 110
5.3.2 Schutz durch Verfügbarkeitszonen 110
5.4 Schutz gegen interne DoS–Angriffe 118
5.4.1 Garantierte Vorhersagen der Grenzen interner DoS–Angriffe . . 119
5.4.2 Reaktive Regelung von Datenflüssen 120
5.5 Verfügbarkeitszonen in SOLID-VPN 121
5.5.1 Bootstrapping und Verbindungsaufbau 121
5.5.2 Topologiekontrolle und Routing 122

5.6 Fazit . . . 123

6 Implementierung 125
6.1 Zusätzliche funktionale Anforderungen . . . 125
6.2 Einbettung in Simulator & Prototyp . . . 126
6.3 Überblick über die Software–Architektur . . . 127
6.4 Bootstrapping . . . 128
6.5 Adresszuweisung . . . 131
6.6 Discovery und Topologiekontrolle . . . 132
6.7 Routing und Paketweiterleitung . . . 133
6.7.1 Eingesetzte Protokollarchitektur . . . 134
6.7.2 Routing–Regeln und –Tabellen . . . 135
6.7.3 Weiterleitung von Multicast–Verkehr . . . 137
6.7.4 NAT–Traversal mit IKE–Mediation . . . 137
6.8 Fazit . . . 138

7 Evaluierung & Diskussion 141
7.1 Qualitative Diskussion . . . 141
7.1.1 Umsetzung funktionaler Anforderungen . . . 141
7.1.2 Nichtfunktionale Eigenschaften . . . 143
7.1.3 Erfüllung der Sicherheits– und Verfügbarkeitsanforderungen . . . 146
7.2 Quantitativ evaluierte Topologien und Szenarien . . . 155
7.2.1 Struktur des Labornetzes . . . 156
7.2.2 VPN mit direkter Konnektivität . . . 156
7.2.3 Verschachtelte VPN–Topologien . . . 157
7.2.4 Gewählte Standardparameter . . . 157
7.3 Evaluierung des Prototypen . . . 159
7.3.1 Reaktiver Aufbau von Sicherheitsbeziehungen . . . 159
7.3.2 Erzeugter Verwaltungsaufwand . . . 161
7.4 Netzwerksimulationen . . . 162
7.4.1 Statische Overlay–Eigenschaften . . . 162
7.4.2 Einfügen und Entfernen von Knoten . . . 167
7.4.3 Verhalten bei Transportnetz–Partitionierungen . . . 171
7.4.4 Verhalten bei partiellen Ausfällen . . . 174
7.4.5 Sabotageresistenz . . . 176
7.5 Fazit . . . 183

8 Resümee 187
8.1 Zusammenfassung . . . 187
8.2 Ausblick . . . 189

A Weiterführende Messungen 193
A.1 Verzögerung der IPsec–Schlüsselaushandlung . . . 193
A.2 Häufigkeit der Verletzung der Dreiecksungleichung bei Verzögerungszeiten . . . 194

Literaturverzeichnis 195

Abkürzungsverzeichnis **213**

Stichwortverzeichnis **215**

I have always wished for my computer to be as easy to use as my telephone; my wish has come true because I can no longer figure out how to use my telephone

(Bjarne Stroustrup)

Kapitel 1

Einleitung und Motivation

Eine hauptsächliche Motivation bei der Entwicklung paketvermittelnder Kommunikationsnetze war deren im Vergleich zu verbindungsorientierten Netzen höhere Robustheit gegen zufällige Ausfälle und Sabotage [Bar64]. Auf dem Gipfel des kalten Krieges sollte in den USA so eine Plattform für den militärischen und behördlichen Nachrichtenaustausch geschaffen werden, welche selbst nach atomaren Angriffen funktionsfähig bleibt. Die dabei entwickelten Prinzipien haben über das ARPANET auch Eingang in die Funktionsweise des heutigen Internets gefunden.

Als Resultat sind die Routing-Mechanismen des Internets daher in der Lage auch größere Ausfälle ohne signifikante Probleme zu kompensieren. So war beispielsweise im New Yorker World Trade Center eine Vielzahl von Internet Providern angesiedelt, um die dort terminierenden Transatlantikkabel in ihre Netze zu integrieren. Trotz der Zerstörung dieser Verbindungen im September 2001 waren weiterhin 99% der Adresspräfixe global erreichbar, und auch die restlichen Netze standen innerhalb weniger Stunden wieder zur Verfügung [Com03].

Im Gegensatz zu den bis dahin aufgebauten Telefonnetzen und anfänglichen Regierungsnetzen, stellt das Internet allerdings kein geschlossenes Netz dar. Diese Offenheit bewirkt zwar, dass ein Zugang nahezu überall auf der Welt preiswert möglich ist; jedoch muss so auch von Angreifern innerhalb des Internets ausgegangen werden. Insbesondere in Bezug auf die Sabotageresistenz, in diesem Kontext teilweise auch als Verfügbarkeit oder Resistenz gegen Denial-of-Service (DoS)-Angriffe bezeichnet, sind die Mechanismen des Internet Protocol (IP) hier noch nicht ausreichend.

1.1 Defizite bei der Konfiguration virtueller privater Netze (VPN)

Um das Internet trotz seiner offenen Architektur zur Realisierung eines schnellen und sicheren, internen Informationsflusses nutzen zu können, schotten Firmen, Behörden, Organisationen aber auch Privatleute ihre Systeme vom Rest des Netzes ab und verbinden einzelne Teile über virtuelle Kanäle. Zur Sicherung von Datenvertraulichkeit, –integrität und –authentizität, sowie zur Realisierung von kontrolliertem

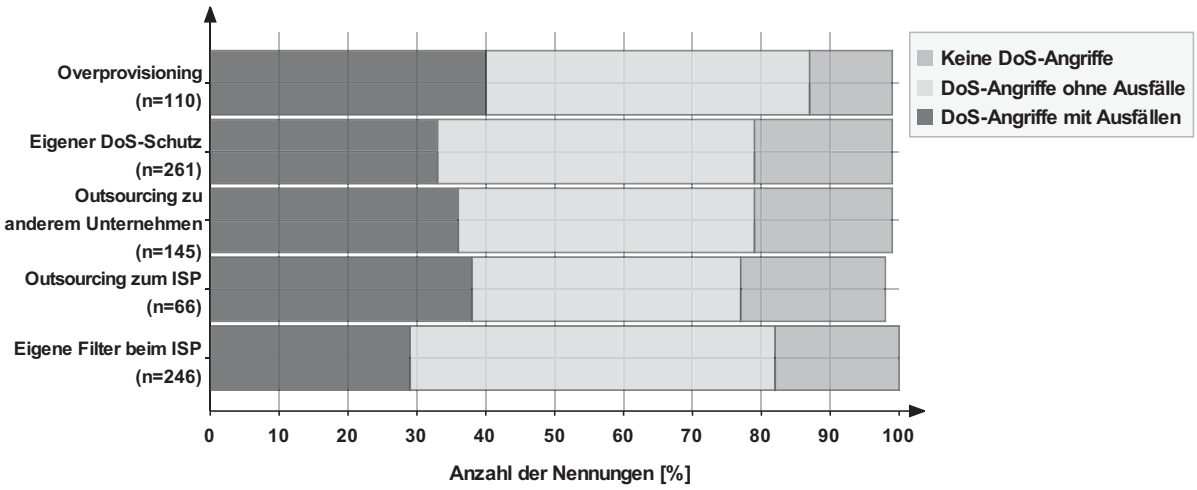


Abbildung 1.1: Umfrage unter 400 IT-Führungskräften zu DoS-Angriffen [For09]: Aufschlüsselung von Gegenmaßnahmen und Auswirkungen auf die IT-Infrastruktur über einen Zeitraum von 12 Monaten

Zugang werden die über die virtuellen Kanäle ausgetauschten Daten kryptographisch gesichert, sodass oberhalb des Internets ein virtuelles privates Netzwerk (VPN) entsteht.

Ein in diesem Zuge häufig realisierter Ansatz setzt ein oder mehrere zentrale VPN-Konzentratoren ein, welche Verbindungen von Außenstandorten oder mobilen Teilnehmern entgegennehmen und zwischen ihnen Daten weiterleiten. Diese Architektur ist leicht zu verwalten, und die Realisierung von Vertraulichkeit, Authentizität, Integrität und Zugriffskontrolle wird in diesem Kontext hinreichend gut beherrscht. Skalierbarkeit in Bezug auf Teilnehmer und Bandbreite, Robustheit sowie Sabotageresistenz stellen hingegen Probleme dar, da diese VPN mit den Konzentratoren einen dedizierten Ausfallpunkt und Flaschenhals besitzen. Auch wenn zurzeit relativ wenig über erfolgreiche Sabotageangriffe auf VPN berichtet wird, liegt dies vermutlich nicht an deren Abwesenheit sondern an den folgenden drei Gründen:

1. Die großen VPN wichtiger Institutionen werden mittels kostenintensiver und relativ unflexibler Service Level Agreements (SLAs) durch einen Internet Service Provider (ISP) vor ungezielten Angriffen geschützt (beispielsweise Verizon DOS Defense [Ham09, Pas06] oder Verisign Internet Defense Network [Ver10]).

2. Angriffe auf VPN erzeugen ferner in der Regel keinen messbaren *Backscatter*, also Pakete die von VPN-Teilnehmern an die Inhaber gefälschter IP-Adressen zurückgesendet werden. Dadurch können DoS-Angriffe auf VPN nicht von dritten Parteien gemessen werden.

3. Aufgrund des mit erfolgreichen Angriffen verbundenen Image-Schadens und der Angst vor Trittbrettfahrern, werden von DoS-Angriffen betroffene Organisationen diese von sich aus nicht an die Öffentlichkeit bringen.

Trotz fehlender Statistiken ist aber gerade bei VPN eine gewisse Immunität gegen DoS-Angriffe extrem wichtig, da sonst mit relativ geringem Aufwand sofort organisationsinterne Arbeitsabläufe vollständig verhindert werden können. Ein Schutz gegen Sabotageangriffe durch SLA funktioniert auch nur begrenzt, da zum einen kleine Institutionen kein solches Abkommen haben. Zum anderen besitzt nur der Betreiber eines VPN die nötigen kryptographischen Schlüssel und kann so legitime VPN-Pakete sicher von Angriffsverkehr unterscheiden. Soll ein VPN potentiell aus dem gesamten Internet erreichbar sein oder kennt der Angreifer die IP-Adressen der Gegenstellen, können die Filter des Internet Provider vergleichsweise leicht umgangen werden. Das Ermitteln von Filterregeln ist bereits für herkömmliche DoS-Angriffe sehr schwer, wie die in Abbildung 1.1 dargestellte Umfrage zeigt. Keine der aufgeführten Methoden zum Schutz gegen DoS-Angriffe kann Ausfälle auf ein akzeptables Niveau senken.

Alternativ können im Kontext von VPN nur sabotageresistentere Topologien erzeugt werden. Ein Beispiel sind die vollvermaschten Topologien, bei denen jeder Standort mit allen anderen verbunden wird, und Angreifer so keinen zentralen Punkt sabotieren können. Die Konfiguration ist jedoch aufwendig und damit potentiell fehleranfällig, erlaubt keine mobilen Teilnehmer, und die Skalierbarkeit in Bezug auf die Anzahl der VPN-Endpunkte ist problematisch, da die Anzahl der einzurichtenden Sicherheitsbeziehungen quadratisch mit ihnen wächst. Wird ein Teilnehmer eines vollvermaschten VPN angegriffen und erhält durch alternative Internet-Anbindung eine andere IP-Adresse, müsste nun in allen Geräten der anderen VPN-Teilnehmer eine Rekonfiguration vorgenommen werden. Die Angreifer können ferner die Kommunikation zwischen zwei Punkten stören, wobei durch die statische Konfiguration in diesem Fall keine Umleitungen von Verkehr innerhalb des VPN möglich sind, obwohl so unter Umständen eine Kommunikation über andere Standorte noch zu realisieren wäre.

Die einzige Möglichkeit auf eine statische Konfiguration zu verzichten, stellt der Einsatz eines Systems zur automatischen Konfiguration dynamischer VPN dar, wobei die existierenden Ansätze allerdings weder Sabotageresistenz noch Robustheit adressieren. Darüber hinaus erfüllen die wenigen skalierbaren Systeme selbst einfache funktionale Anforderungen nicht, wie etwa den Einsatz privater IP-Adressen innerhalb des VPN. Einige Ansätze schwächen zusätzlich die Sicherheit des VPN signifikant, indem beispielsweise unsichere kryptographische Verfahren oder Gruppenschlüssel eingesetzt werden (siehe Abschnitt 3.2).

Insgesamt erweist sich somit die manuelle Konfiguration großer VPN als zu aufwendig, fehleranfällig, unflexibel und bietet keinen ausreichenden Schutz vor bandbreitenerschöpfenden DoS-Angriffen. Die bestehenden VPN-Autokonfigurationssysteme vereinfachen zwar die Einrichtung und sind potentiell weniger anfällig gegen menschliche Fehler, berücksichtigen aber in der Regel lediglich wenige Szenarien, und sehen keine Mechanismen zur Realisierung von Robustheit und Sabotageresistenz vor.

1.2 Beiträge der Arbeit

Ausgehend von dieser Situation wurden im Rahmen der Dissertation und der vorangegangen Diplomarbeit [Ros07] zunächst umfangreiche Anforderungen für die automatische Konfiguration von VPN ermittelt. Wie bereits angedeutet, ergab ein systematischer Vergleich des Standes der Wissenschaft und Technik mit den Anforderungen eine relativ große Lücke [RS11].

Infolgedessen ist ein eigener Ansatz mit dem Namen Secure OverLay for IPsec Discovery (SOLID) entstanden, der von beherrschten Peer-to-Peer-Prinzipien ausgehend, ein vollständiges System zur Konfiguration von VPN mit folgenden Eigenschaften bereitstellt:

- Im Gegensatz zu bestehenden Ansätzen zeichnet sich SOLID durch die Realisierung umfangreicher funktionaler Anforderungen aus, wie der Möglichkeit verschachtelte Topologien oder VPN mit privaten IP-Adressbereichen zu konfigurieren.
- Durch die Einrichtung verschachtelter kryptographischer Tunnel wird die Sicherheit der übertragenen Daten selbst bei Anwesenheit interner Angreifer gewährleistet.
- Der konsequente Verzicht auf exponierte VPN-Teilnehmer und die Nutzung der VPN-Topologie zur Ableitung von Routing-Informationen ermöglichen einen robusten Umgang mit Ausfällen, aber auch eine schnelle Reaktion auf Sabotageangriffe.

Um die Resistenz gegen mögliche Sabotageangriffe weiter zu erhöhen, wurden im Kontext dieser Arbeit Ansätze zur Steigerung der Verfügbarkeit von VPN entwickelt und in SOLID integriert. Hierbei werden VPN-Teilnehmer in so genannte Verfügbarkeitszonen eingeteilt und eine direkte Kommunikation nur zwischen vordefinierten Zonen erlaubt. Auf diese Weise wird es für potentielle Angreifer schwieriger, Ziele für Sabotageangriffe zu identifizieren, und so die Resistenz des Netzes insgesamt erhöht.

SOLID ist ferner im Rahmen einer Simulationsumgebung implementiert worden und kann so auch für große Netze untersucht werden. Zusätzlich werden große Teile des Quellcodes wiederverwendet, um einen Prototypen auf Linux-Basis zu realisieren. Die vollautomatische Einrichtung von Kommunikationsinfrastrukturen auf Basis von IP security (IPsec) kann so simulativ und experimentell mit nahezu identischen Ergebnissen untersucht werden.

1.3 Übersicht

Die vorliegende Arbeit gliedert sich in acht Kapitel. Je nach Wissensstand und Interesse empfiehlt es sich, einige Passagen zu überspringen oder genauer zu lesen. Abbildung 1.2 gibt einen Überblick über einige mögliche Reihenfolgen.

Im folgenden zweiten Kapitel werden zunächst einige grundlegende Fakten zu Zielen der Netzwerksicherheit sowie aktuell eingesetzten VPN-Architekturen und Protokollen gegeben. Anschließend erfolgt ein Überblick über strukturierte Peer-to-Peer-Systeme und ihren möglichen Einsatz zur Weiterleitung von Unicast- und Multicast-Verkehr. Zusätzlich wird auf Netzwerkkoordinatensysteme eingegangen, dort im Speziellen auf den Vivaldi-Ansatz, und es werden mögliche Angriffe auf Peer-to-Peer-Systeme skizziert.

Kapitel 3 beginnt mit einem Überblick über funktionale und nichtfunktionale Anforderungen an VPN-Konfigurationssysteme. Sicherheitsanforderungen, im Speziellen in Bezug auf Verfügbarkeit, werden in eigenen Abschnitten detailliert dargelegt. Die erarbeiteten Anforderungen werden in Abschnitt 3.2 mit den 17 existierenden Ansätzen zur VPN-Autokonfiguration verglichen. Im Anschluss erfolgt ein kurzer Überblick über Overlay-Netze, die explizit konzipiert wurden, um die Sabotageresistenz angebotener Dienste zu erhöhen. Als dritte Komponente des Standes der Wissenschaft werden verschiedene Verfahren zur Realisierung von Routing in strukturierten Overlay-Netzen in Abschnitt 3.4 vorgestellt.

In Kapitel 4 wird SOLID — als Hauptansatz der Dissertation — vorgestellt. Beginnend mit einem kurzen Abriss über Prinzipien, Annahmen, potentielle Angreifertypen und zu erfüllenden Aufgaben, wird anschließend auf die Realisierung einzelner Dienste, wie etwa Bootstrapping und Topologiekontrolle eingegangen. Den letzten Teil des Kapitels bildet die Beschreibung von Multicast-Routing in SOLID-Netzen.

Die in SOLID durch eine dynamische Topologiekontrolle gewonnene Flexibilität kann zum Schutz vor DoS-Angriffen eingesetzt werden. Kapitel 5 geht dazu zunächst auf die Modellierung von VPN und DoS-Angriffen mittels Graphen ein. Anschließend werden optimale, externe DoS-Angriffe vorgestellt, sowie das Problem auf diese bestmöglich zu reagieren. Das Kapitel schließt mit einer Betrachtung interner DoS-Angriffe, sowie der Übertragung der vorgestellten Techniken auf SOLID.

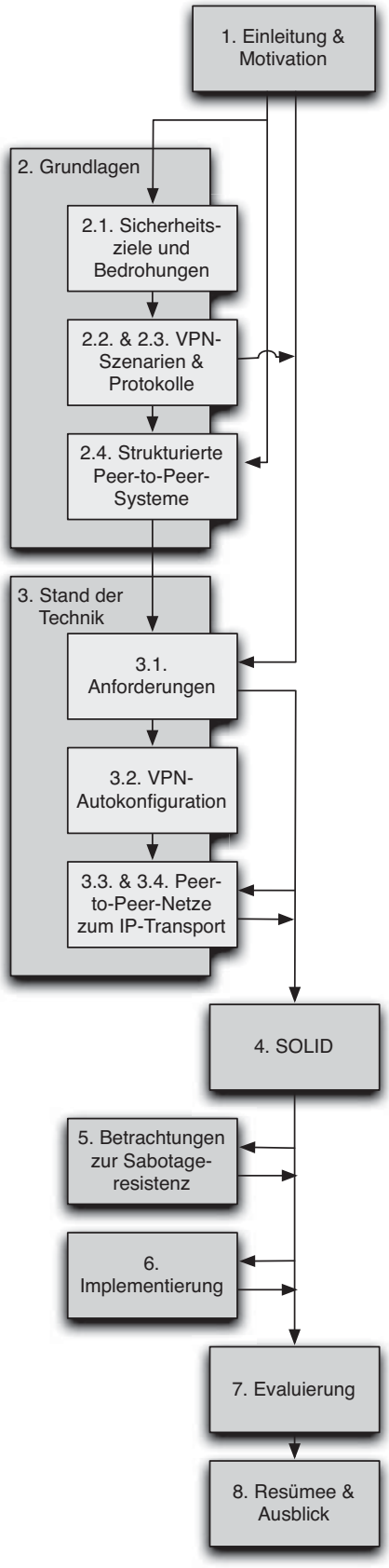


Abbildung 1.2: Empfohlene Lesereihenfolgen

Das sechste Kapitel beschäftigt sich mit der Umsetzung des vorgestellten Systems in Simulationsumgebung und Prototyp. In einem ersten Abschnitt werden dazu weitere die Implementierung betreffende Anforderungen hergeleitet, bevor die Umsetzungsstrategie und Software-Architektur vorgestellt werden. Die Realisierung ausgewählter Funktionen, wie Bootstrapping und Routing, wird in einzelnen Unterkapiteln detaillierter beschrieben.

Das letzte Hauptkapitel 7 beschäftigt sich mit dem Nachweis erreichter Anforderungen und Ziele. Im ersten Unterkapitel wird SOLID dazu systematisch in Bezug auf die in Abschnitt 3.1 aufgestellten Anforderungen diskutiert. Anschließend wird auf verschiedene Simulations- und Experimentalumgebungen eingegangen, in welchen SOLID quantitativ untersucht und bewertet wurde. Die folgenden Abschnitte beschreiben die eigentlichen Experimente und Resultate in Prototyp und Netzwerksimulator.

Die Arbeit schließt mit einer kurzen Zusammenfassung und einem Ausblick über mögliche, fortsetzende Arbeiten in Kapitel 8. Dabei wird sowohl auf zukünftige forschungsorientierte Aspekte als auch auf eventuelle Produktisierungsschritte eingegangen.

Повторение - мать учения
(Wiederholung - Mutter des Lernens)

(Russisches Sprichwort)

Kapitel 2

Grundlagen: VPN–Konfiguration und strukturierte Peer–to–Peer–Systeme

In diesem Kapitel werden zunächst kurz die allgemeinen Sicherheitsziele und Bedrohungen in Datennetzwerken, sowie Grundlagen und Einsatzszenarien IP-basierter VPN vorgestellt, bevor im dritten Teil auf konkrete, kryptographische Protokolle zur Absicherung von VPN eingegangen wird. Abschnitt 2.4 gibt einen Überblick über die wichtigsten Vertreter strukturierter Peer-to-Peer-Systeme. Hierbei werden sowohl Ansätze zur Realisierung von Unicast- als auch Multicast-Kommunikation behandelt. Das Kapitel schließt mit einer Vorstellung von Netzwerkkoordinatensystemen und bekannten Angriffsstrategien auf Peer-to-Peer-Systeme.

Weitere, ergänzende Informationen zum Thema Netzwerksicherheit können beispielsweise [For94], [Sch03] oder [Eck09] entnommen werden. Details zu Peer-to-Peer-Systemen finden sich unter anderem in [SW05, VLO09, BYL08].

2.1 Sicherheitsziele und Bedrohungen in paketorientierten Netzen

Normalerweise sollen bei der IT-Sicherheit und damit auch in paketorientierten Netzen fünf allgemeine Schutzziele realisiert werden [Sch03]:

- **Vertraulichkeit:** Beispielsweise durch kryptographische, physische oder organisatorische Maßnahmen soll für sensible Informationen garantiert werden, dass lediglich bestimmte Instanzen Kenntnis von diesen erhalten können.
- **Integrität:** Daten sollen nicht unbemerkt modifiziert werden können.
- **Zurechenbarkeit:** Die Herkunft von Datensätzen soll zu jedem Zeitpunkt nachvollziehbar (**Authentizität**) beziehungsweise Dritten gegenüber glaubhaft nachweisbar (**Nichtabstreitbarkeit**) sein. Die Kenntnisnahme und Modifikation von Daten sowie die Nutzung von Diensten (zum Beispiel Übertragungsdienste) soll lediglich durch autorisierte Instanzen erfolgen.
- **Verfügbarkeit:** Es soll nicht möglich sein, bereitgestellte Dienste zu stören.

Die Arbeit folgt an dieser Stelle der Definition aus [Sch03]. Auch wenn der kontrollierte Zugang nach anderen Lehrmeinungen oft lediglich als Sicherheitsdienst zum Erreichen von Vertraulichkeit, Integrität, Authentizität oder Verfügbarkeit aufgefasst wird, ist es bei Kommunikationsnetzen oft üblich diesen davon unabhängig zu gewährleisten. Beispielsweise kann so wegen rechtlicher Rahmenbedingungen erzwungen werden, dass Weiterleitungspunkte im Netz nur von Berechtigten benutzt werden oder bestimmte Nutzergruppen nur einen eingeschränkten Zugang erhalten [Bun10]. In [For94] wird dieses Sicherheitsziel auch als *Legitimate Use* bezeichnet.

Auf der anderen Seite versuchen potentielle Angreifer ihrerseits Ziele zu erreichen, indem sie zumindest gegen eines der Schutzziele verstoßen. In paketvermittelnden Netzen, wie dem Internet, können sie dazu folgende prinzipiellen Strategien verfolgen:

- **Abhören und Verkehrsflussanalyse:** Angreifer, die beispielsweise über einen Zugang zu übermittelnden Leitungen oder Zwischensystemen verfügen, beziehungsweise drahtlose Übertragungen auswerten, können übertragene Pakete mitlesen und analysieren. Ohne weitere Sicherungsmaßnahmen ist die Vertraulichkeit übertragener Informationen nicht länger gegeben.

- **Grayhole- und Blackhole-Angriffe:** Neben dem Mitlesen kann die Weiterleitung von Paketen auch teilweise oder vollständig unterdrückt werden. Erstere werden je nach Schreibstil als Grayhole- oder Greyhole-Angriffe bezeichnet. Die selektive Unterdrückung von Paketen kann bei diesen Attacken auch gezielt erfolgen, beispielsweise wenn anhand der Paketgrößen einen Rückschluss auf deren Typ möglich ist. Je nach System und Angriff kann dadurch die Integrität der übertragenen Daten verletzt werden, beziehungsweise eine *Sabotage* erfolgen, welche die Verfügbarkeit negativ beeinflusst.

- **Paketverzögerung:** Kontrolliert der Angreifer die Weiterleitung von Datenpaketen, kann er dies mit einer selektiven Verzögerung tun. Die Attacke kann zum einen zum Reduzieren der Leistungsfähigkeit angebotener Dienste führen (Sabotage). Im Gegensatz zu Gray- oder Blackhole-Angriffen kann zum anderen bei der Messung von Verzögerungszeiten die Integrität gefährdet werden, wie etwa beim Einsatz des Network Time Protocol (NTP).

- **Wiedereinspielen von Paketen:** Durch das wiederholte Senden vormals legitimer Pakete können Integrität und Verfügbarkeit negativ beeinflusst werden, indem beispielsweise Aktionen erneut ausgelöst (Angriff auf Authentizität) oder Überlastsituationen erzeugt werden (Sabotage).

- **Senden gefälschter Pakete:** Das Erzeugen neuer Pakete ist für einen Angreifer oft das wirkungsvollste Mittel in paketorientierten Netzen Attacken durchzuführen. Dabei existiert eine Vielzahl verschiedener Methoden. Beispielsweise ist es in ungesicherten Systemen möglich, zwischen zwei Knoten zu vermitteln und diesen vorzutäuschen, sie würden direkt miteinander kommunizieren. Diese Man-in-the-Middle-Attacke ermöglicht eine Verletzung von Vertraulichkeit, Integrität, Zurechenbarkeit, kontrolliertem Zugang und Verfügbarkeit. Ferner kann etwa durch das Senden von einer großen Anzahl gültiger Anfragen eine

Überlastsituation erzeugt oder durch das Senden falsch formatierter Anfragen ein Dienst zum Absturz gebracht werden (Sabotage).

Um sich gegen diese Angriffe zu schützen, und so in der Lage zu sein auch sensible Informationen über öffentliche Netze auszutauschen, können innerhalb dieser Netze VPN betrieben werden. Diese sichern durch kryptographische Maßnahmen zu, dass ein möglichst großer Teil der Sicherheitsziele trotzdem erreicht werden kann. Der folgende Abschnitt geht auf die Anwendungsfälle ein, in denen VPN oftmals eingesetzt werden.

2.2 VPN–Szenarien

Verschiedene Organisationen und Individuen setzen VPN teilweise sehr unterschiedlich ein, weshalb im Folgenden einige der vorherrschenden Einsatzszenarien beschrieben werden. Aufgrund ihrer Dominanz bei Endnutzern und der hohen Flexibilität wird dabei ausschließlich auf IP-basierte Ansätze eingegangen.

2.2.1 VPN von Firmen & Nichtregierungsorganisationen

Besonders für kleinere Firmen ist der Haupteinsatzzweck von VPN die Einbindung von mobilen Heim- oder Außendienstmitarbeitern, welche in diesem Kontext oft als *Road Warriors* bezeichnet werden. Die in diesem *Remote-Zugriffsszenario* entstehenden VPN-Topologien sind vergleichsweise einfach: Wie in Abbildung 2.1 illustriert ist, verbindet sich jeder mobile Teilnehmer über ein nicht vertrauenswürdiges, *schwarzes Netz* zu einem zentralen VPN-Konzentrator, der ihm durch einen kryptographischen Tunnel sicheren Zugriff auf das interne Netz, welches oft als *rotes Netz* bezeichnet wird, und die Knoten aller anderen VPN-Teilnehmer gewährt. Der Begriff *VPN-Knoten* wird in dieser Arbeit verwendet, um jegliche Geräte zu bezeichnen, die über kryptographisches Material verfügen und direkt VPN teilnehmen. In Abgrenzung dazu bezeichnen VPN-Gateways lediglich Geräte, die ein rotes Netz transparent in ein VPN einbinden. Innerhalb der roten Netze werden oft private IP-Adressen verwendet, sodass jeder VPN-Knoten mindestens zwei IP-Adressen besitzt: eine *äußere* zur Kommunikation im schwarzen Netz und eine *innere* für den gesicherten Datenaustausch.

In größeren Organisationen werden VPN oft zur Verbindung einzelner Außenstandorte eingesetzt; ein so genanntes *Site-to-Site-VPN* entsteht. Im Vergleich zum Remote-Zugriffsszenario sind die Sicherheitsbeziehungen hier relativ statisch, das heißt sie ändern sich kaum in Bezug auf ihre öffentliche IP-Adresse, Bandbreite oder Verzögerung. Ein weiterer Unterschied ist, dass ganze Netzwerke miteinander verbunden und nicht einzelne Knoten integriert werden. In Bezug auf Sicherheit sind VPN-Gateways in diesen Szenarien durch physische Maßnahmen in der Regel besser gegen Kompromittieren gesichert, sodass in Regel nicht von internen Angreifern ausgegangen wird.

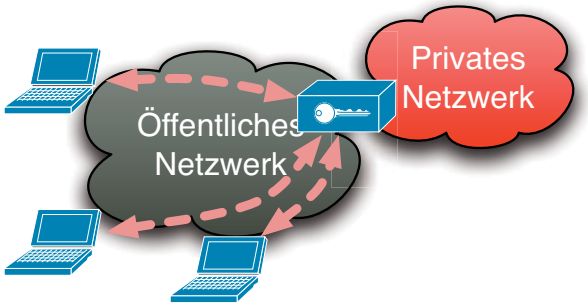


Abbildung 2.1: Beispiel eines Remote-Zugriffsszenarios für die Einbindung mobiler Knoten

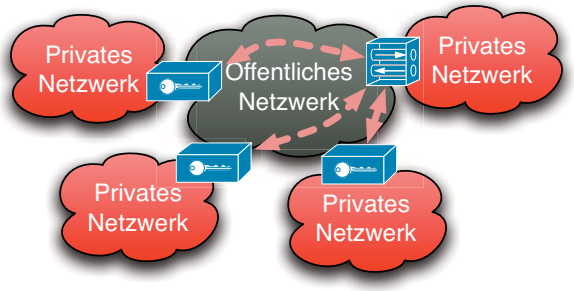


Abbildung 2.2: Hub-and-Spoke-Architektur zur Verbindung von Außenstandorten

Trotz dieser Unterschiede zum Remote-Zugriffsszenario wird laut [RRKC04] bei circa der Hälfte der Site-to-Site-VPN eine ähnlich *zentralisierte* Architektur realisiert. Wie in Abbildung 2.2 illustriert, verbinden sich die einzelnen Außenstandorte sternförmig zu einem VPN-Konzentrator der Hauptniederlassung, der auch die Weiterleitung von Paketen zwischen Außenstandorten übernimmt. Aufgrund der an eine Radnabe erinnernden Topologie, wird oft von einer *Hub-and-Spoke*-Architektur gesprochen. Da für den Aufbau des VPN lediglich eine Sicherheitsbeziehung pro Außenstandort benötigt wird, liegt der Vorteil dieses Ansatzes im geringen Verwaltungsaufwand.

Während Hub-and-Spoke-Architekturen in Bezug auf die niedrige Anzahl der einzurichtenden Sicherheitsbeziehungen ein Extrem darstellen, existiert mit den *vollvermaschten Topologien* das andere Extrem, bei dem jeder Außenstandort mit jedem anderen verbunden ist (siehe auch Abbildung 2.3). Auf diese Weise müssen zwar $\mathcal{O}(n^2)$ Sicherheitsbeziehungen aufgebaut werden, jedoch bieten vollvermaschte Topologien Vorteile in Bezug auf die erreichte Übertragungsbandbreite und -verzögerung. Ferner vermeiden sie durch die *verteilte* Organisation die Schaffung eines Single Point of Failure.

Zwischen diesen Extremen bewegen sich *dezentrale* Architekturen, wie die des Dynamic Multipoint VPN (DMVPN) [Flu07b]. Hier werden mehrere VPN-Konzentratoren zu einem unstrukturierten, unter Umständen vollvermaschten, statischen Kernnetz konfiguriert. Dynamische Knoten oder Gateways verbinden sich zu einem oder mehreren Konzentratoren, welche übergebene Daten umverschlüsseln und zum jeweili-

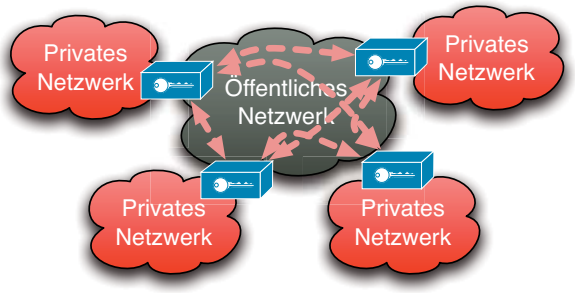


Abbildung 2.3: Ein vollvermaschtes Site-to-Site-VPN

gen Zielkonzentrator weiterleiten. Paket-Routing und -weiterleitung zwischen den Konzentratoren können so mithilfe gewöhnlicher IP-Mechanismen realisiert werden.

Neben diesen Architekturen existieren weitere Spezialfälle. Aus Verfügbarkeits- oder Skalierbarkeitsgründen kann es notwendig werden, mehrere VPN-Gateways zur Anbindung eines roten Netzes einzusetzen. Ferner kann es beispielsweise erforderlich sein, eine Entwicklungsabteilung vom Rest des Intranets durch ein weiteres VPN-Gateway abzugrenzen, wodurch *verschachtelte* VPN entstehen. Außendienstmitarbeiter müssen in diesem Fall eine *indirekte Sicherheitsbeziehung* aufbauen, um zunächst in das allgemeine Firmennetz und von dort aus in die Entwicklungsabteilung zu gelangen.

2.2.2 VPN von Behörden und Militär

Sowohl behördliche als auch militärische Netze unterscheiden sich zunächst nicht sehr von den vorgestellten zivilen VPN-Einsatzszenarien. Allerdings betreiben sie zum Teil eigene Transportnetze [Rei07], innerhalb derer sie relativ abgeschottet sind und beispielsweise natives Multicast-Routing einsetzen können.

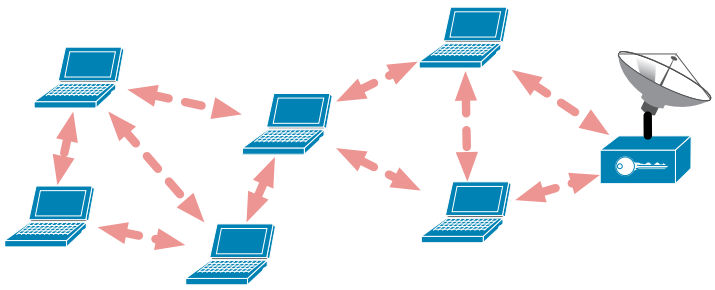


Abbildung 2.4: Taktisches Netzwerk mit Satellitenverbindung

Neben diesen relativ statischen VPN werden zur Erfüllung hoheitlicher Aufgaben auch stark dynamische VPN benötigt, die etwa auf MANET aufbauen [GTEB05, DMP+04, HKE09]. Diese in Abbildung 2.4 beispielhaft dargestellten VPN-Architekturen werden im Militärjargon oft als *taktische Netze* bezeichnet. Die gleichen Anforderungen finden sich aber auch bei humanitären Einsätzen in Katastrophenszenarien. Durch die schwierige Arbeitsumgebung dieser VPN müssen sie unter anderem mit hohen

Fluktuationsraten, schlechten Funkbedingungen und potentiell kompromittierten VPN-Knoten umgehen können.

2.2.3 Privat eingesetzte VPN

Einzelpersonen nutzen VPN häufig, um sich von unterwegs mit Heimnetzen zu verbinden [VPN09] oder Verbindungen zu privaten Spiele-Servern aufzubauen. Dieses Szenario unterscheidet sich zunächst nicht von den in Abbildung 2.1 dargestellten, kommerziellen Remote-Zugriffsszenarien.

Von anhaltender Entwicklung ist die Nutzung von VPN zur Absicherung von Tauschbörsen innerhalb geschlossener Nutzergruppen. Hierbei kommen entweder konventionelle VPN-Techniken zum Einsatz, indem ein sicherer Tunnel zu einem vertrauenswürdigen Provider aufgebaut wird [Kra09b, Rel09], oder es werden einige der im folgenden Kapitel vorgestellten Ansätze zur automatischen Konfiguration von VPN eingesetzt.

2.3 Kryptographische Protokolle zur Absicherung von VPN

Auch wenn die vorgestellten VPN-Architekturen und der jeweils mit ihnen verbundene Konfigurationsaufwand relativ unabhängig von dem für die eigentliche Absicherung gewählten kryptographischen Protokoll sind, so hat dieses jedoch einen erheblichen Einfluss auf die zu erreichende Effizienz und Sicherheit. Da ferner Implementierungsdetails immer wieder auf bestimmte Eigenschaften der kryptographischen Protokolle aufbauen, werden die gebräuchlichsten in den folgenden Teilabschnitten vorgestellt.

2.3.1 SSL, TLS und DTLS

Das wahrscheinlich verbreiteteste kryptographische Protokoll stellt Transport Layer Security (TLS) [DR08] dar, welches eine von der Internet Engineering Task Force (IETF) standardisierte und weiterentwickelte Variante von Netscape's Secure Socket Layer (SSL) Protokoll ist. Da beide Protokolle ursprünglich zur Ende-zu-Ende-Absicherung der Kommunikation auf Applikationsebene entwickelt wurden, bauen sie auf das Transmission Control Protocol (TCP) auf und sind in Softwarebibliotheken implementiert, welche im Sicherheitskontext des jeweiligen Nutzers laufen. Dort werden sie durch die Applikationslogik direkt aufgerufen. Nichtsdestotrotz verwenden Systeme wie CloudVPN [Kra09a] und OpenVPN [Fei06] virtuelle Netzwerkschnittstellen, so genannte *TAP devices*, um generischen Nutzdatenverkehr durch einen Daemon abzufangen und über TLS-Verbindungen sicher an andere VPN-Teilnehmer auszuliefern.

Wie bereits angedeutet, verwenden sowohl TLS als auch SSL TCP-Verbindungen zur Datenübertragung. Daher sind sie zum einen aufgrund der auftretenden Verzögerungen nicht geeignet, Echtzeitdaten zu transportieren. Zum anderen führt bereits die Übermittlung von TCP-Daten innerhalb von TLS-Verbindungen durch die Staukontrolle zu Effizienzverlusten [HOI+04]. Darüber hinaus existiert beim Einsatz von Network Address Translation (NAT) durch den mit TCP eingeführten Verbindungszustand nicht die Möglichkeit, mit der gleichen Zuverlässigkeit *Hole Punching* zu betreiben, wie dies beim User Datagram Protocol (UDP) möglich wäre. Unter letzterem wird die Technik verstanden, zwischen zwei Geräten, welche sich jeweils hinter einem NAT-Gateway befinden, mithilfe eines öffentlichen Servers eine direkte Verbindung auszuhandeln [FSK05].

Aus diesen Gründen wurde das Datagram Transport Layer Security (DTLS) Protokoll spezifiziert [RM04, RM06], welches die Funktionalität von TLS auf UDP als Transportprotokoll portiert. Um mit verworfenen Paketen, Paketdoppelungen und falschen Paketreihenfolgen umgehen zu können, führt DTLS Sequenznummern und ein Schiebefenster ein. Abgesehen davon, dass es im Kontext des Nutzers abläuft, wird DTLS damit von der Struktur vergleichbar zu IPsec, welches im Folgenden vorgestellt wird.

2.3.2 Die IPsec–Protokollfamilie

Im Unterschied zu TLS, wurde IPsec [KS05, Fra01] zur Sicherung des Datenverkehrs auf Internet-Schicht konzipiert und ist spätestens mit IPv6 ein integraler Bestandteil der IP-Familie. Durch die direkte Implementierung im Betriebssystemkern können Daten vergleichsweise schnell verarbeitet und gegebenenfalls weitergeleitet werden. Dabei ist nicht nur der Einsatz in Endsystemen vorgesehen; IPsec kann auch in VPN-Gateways eingesetzt werden, und so eine Verbindung mehrerer roter Netze zu einem Site-to-Site-VPN ermöglichen.

Aus technischer Sicht unterscheidet der IPsec-Standard Schlüsselaushandlungsprotokolle, im Speziellen Internet Key Exchange (IKE) oder das modernere IKEv2 [Kau05], und Protokolle zur Datenübertragung: Encapsulating Security Payload (ESP) und den weniger gebräuchlichen Authentication Header (AH). Im Rahmen dieser Arbeit werden aufgrund ihres deutlich größeren Funktionsumfanges lediglich IKEv2 und ESP eine Rolle spielen. Allein bei letzterem werden drei Modi unterschieden, die für diese Arbeit signifikant sind:

- Durch die Nutzung von ESP im *Transportmodus* können einander direkt erreichende Endsysteme die Nutzdaten von IP-Paketen transparent verschlüsseln und gegebenenfalls authentisieren. Dabei werden zwischen IP-Header und Nutzdaten ein ESP-Feld eingefügt, eine Aktualisierung von Protokollfeld und Prüfsumme im IP-Header vorgenommen, und die Nutzdaten verschlüsselt sowie bei Bedarf authentisiert. Eine Sicherung des eigentlichen Headers wird nicht vorgenommen, ist jedoch oft auch nicht zwingend erforderlich, da Modifikationen am Protokollkopf in der Regel lediglich bewirken können, dass ein Paket verworfen wird.

- Soll mit einem VPN-Gateway kommuniziert werden, muss der *Tunnelmodus* zum Einsatz kommen. Bei diesem wird das vollständige IP-Paket verschlüsselt und gegebenenfalls mit Authentisierungsinformationen versehen. Die entstehenden Daten werden anschließend mit einem neu generierten IP-Paket versendet. Der Nachteil gegenüber dem Transportmodus ist primär der Mehraufwand bei der Übermittlung. Allerdings können im inneren IP-Header bei Bedarf private Adressen verwendet werden, sodass beispielsweise ein Außendienstmitarbeiter auch ein Gerät hinter einem VPN-Gateway adressieren kann.
- Kommunizieren Endsysteme miteinander, sollen semantisch aber private IP-Adressen verwendet werden, kann der noch vergleichsweise unbekannte *Bound End-to-End Tunnel (BEET) Modus* [NM08] eingesetzt werden. Bei diesem werden ähnlich wie im Tunnelmodus während der IKE-Aushandlung virtuelle IP-Adressbereiche für innere Netze ermittelt; sie sind aber von der Größe auf genau eine IP-Adresse beschränkt. Durch diese Festlegung kann der innere IP-Header im Empfänger wiederhergestellt werden, ohne dass eine explizite Übertragung erforderlich ist. Die Pakete ähneln somit Transportmodus-Paketen, bei denen Sender und Empfänger ein NAT auf interne Adressen vornehmen. Durch die strikte Trennung von Identität und Erreichbarkeit wird eine Reihe von Problemen des Transportmodus gelöst. So vereinfacht sich das Zusammenspiel mit nicht-vertrauenswürdigen NAT-Gateways, und der Wechsel der externen IP-Adressen von IPsec-Knoten führt nicht zwangsläufig zu einem Bruch der Sicherheitsbeziehung.

Neben diesen Modi existiert noch die *IIPtran* genannte Möglichkeit, Tunnel zu betreiben, bei der IPIP-Tunnel durch Sicherheitsbeziehungen im Transportmodus abgesichert werden [TEW04]. Die gesicherten Pakete sind dabei vollständig äquivalent zu denen, die von IPsec im Tunnelmodus erzeugt würden. Der Unterschied zwischen IIPtran und Tunnelmodus besteht allein in der Art der Verarbeitung im Betriebssystem.

Der angedeutete große Funktionsumfang von IPsec führt aber auch zu dem viel zitierten Nachteil der hohen Komplexität von Spezifikation und Implementierungen [FS00]. Allerdings spricht eine Reihe von Punkten für zunehmende Nivellierung dieses Argumentes im Vergleich zu den anderen Verfahren:

- Mit der Einführung von IKEv2 wurde das Schlüsselaushandlungsprotokoll von IPsec stark vereinfacht.
- Die Implementierungen in Betriebssystemen und IKE-Daemons haben eine Entwicklung von über zehn Jahren durchlaufen, sodass sie trotz ihrer Komplexität als relativ stabil und interoperabel gelten.
- Innerhalb dieser zehn Jahre seit der initialen Standardisierung sind keine praktisch relevanten Sicherheitslücken bekannt geworden (beispielsweise im Gegensatz zu SSL, TLS [RRDO10] und vielen proprietären Protokollen).
- Nicht zuletzt hat die anhaltende Kritik für die Entwicklung der in den folgenden Kapiteln vorgestellten Autokonfigurationsmechanismen gesorgt, die zumindest versprechen, einen Teil der Komplexität vor dem Nutzer zu verbergen.

2.3.3 Weitere kryptographische Protokolle

Neben TLS und IPSec existieren eine ganze Reihe weiterer Protokolle zur Absicherung von VPN. Am bekanntesten ist das Point-to-Point Tunneling Protocol (PPTP) [HPV+99], welches Ende der neunziger Jahre von einer Gruppe Netzwerkhersteller (unter anderem 3Com und Microsoft) entwickelt wurde. Trotz bekannter, teils erheblicher Sicherheitsprobleme [Lev98] wird PPTP, aufgrund seiner einfachen Konfiguration, noch in vielen Infrastrukturen eingesetzt. Für sicherheitskritische Entwicklungen spielt es jedoch keine Rolle mehr.

Layer Two Tunneling Protocol (L2TP) [TVR+99] ist ein Protokoll, welches wie PPTP ebenfalls zum Aufbau von Tunneln für Daten der Schicht 2 und die passwortbasierte Nutzerauthentisierung verwendet wird. Im Unterschied zu PPTP wird beim Einsatz über IP-Netze jedoch ein zusätzlicher IPsec-Schutz gefordert, sodass es im Bereich der VPN-Overlays in der Regel nur zu Abrechnungszwecken als Erweiterung zum IPsec-Schutz verwendet wird (beispielsweise [gem09]).

Einige der im nächsten Kapitel vorgestellten Ansätze zur VPN-Autokonfiguration verwenden eigene, proprietäre kryptographische Protokolle (beispielsweise Wippien, WASTE, P2PVPN, N2N und Hamachi[2]). Diese sind jedoch nicht vollständig offengelegt oder wenn Teile offengelegt sind, führen sie zu teils schweren Sicherheitsbedenken. Ohne Standardisierungsbemühungen werden sie Nischenprodukte bleiben und können nicht für sicherheitskritische VPN eingesetzt werden.

2.4 Strukturierte Peer–to–Peer Systeme

Völlig unabhängig von ihrem Einsatzzweck werden verteilte Systeme in den letzten Jahren verstärkt so entworfen, dass sie durch die Struktur der aufgebauten Overlays besonders effizient oder skalierbar bezüglich der Anzahl der Teilnehmer werden. Darüber hinaus zeichnen sich die, aufgrund ihres Verzichts auf exponierte Knoten als *Peer-to-Peer*-Netze bezeichneten, Systeme durch Selbstorganisation und Fehlertoleranz aus. Das bedeutet, sie adaptieren ihre Topologie an die Gegebenheiten im Transportnetz und Overlay selbst. Gerade dieser Selbstorganisationsaspekt macht sie zu einer idealen Basis für die verteilte VPN-Autokonfiguration, daher werden im Folgenden verschiedene Aspekte von Peer-to-Peer-Systemen näher erläutert. Da die aufgebauten Topologien je nach Anwendungsfall völlig unterschiedlich ausfallen, wird dabei zwischen Systemen zur Realisierung verteilter Hash-Tabellen und Systemen zur Gruppenkommunikation unterschieden. Anschließend wird auf Netzwerkkoordinatensysteme zur Abschätzung von Transportnetzverzögerungen innerhalb von Peer-to-Peer-Systemen eingegangen. Der Abschnitt schließt mit einer Zusammenfassung bekannter Angriffe auf strukturierte Peer-to-Peer-Netze.

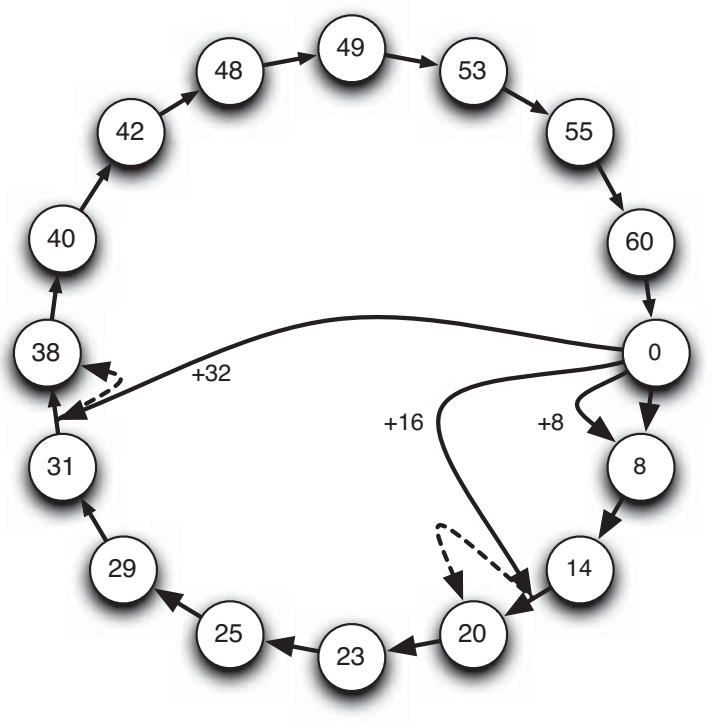


Abbildung 2.5: Chord-Overlay mit 64 möglichen Hash-Werten und Querverbindungen für Knoten 0

2.4.1 Verteilte Hash–Tabellen

Ähnlich wie bei den VPN sind auch verteilte Hash-Tabellen beziehungsweise Distributed Hash Table (DHT) über die letzten zehn Jahre gut untersucht worden. Die Kernidee der meisten Ansätze ist dabei zunächst die gleiche: Die teilnehmenden Knoten wählen sich eine zufällige Kennung und suchen sich auf dieser Basis sowie einer durch das System vorgegeben Struktur ihre Nachbarknoten, zu denen sie Verbindungen aufbauen. Jeder Teilnehmer des Overlay-Netzes kann nun Daten unter einem Schlüsselwert ablegen oder abrufen, indem er den jeweiligen Knoten kontaktiert, dessen Kennung dem Hash-Wert des Schlüssels am ähnlichsten ist. Je nach Ansatz werden teils komplexe Replikationsmechanismen verwendet, um Knotenausfälle und Inkonsistenzen zu kompensieren.

Stärkstes Unterscheidungsmerkmal der entwickelten Systeme ist dabei die Struktur des Overlay-Netzes und die damit verbundenen Such- und Einfügealgorithmen. Eines der ersten DHT ist Chord [SMK+01]. Wie in Abbildung 2.5 illustriert, bauen Knoten hier basierend auf ihren Kennungen eine Ringtopologie auf. Um nicht bei jeder Suche in einem Overlay mit n Knoten $\mathcal{O}(n)$ viele Nachrichten entlang der Ringstruktur senden zu müssen, werden zusätzlich unidirektionale Querverbindungen mit logarithmischem Abstand durch den Ring platziert. Aus der Konstruktion resultiert ein Overlay-Netz, bei dem jeder Knoten $\mathcal{O}(\log n)$ viele Verbindungen aufbaut, aber gleichzeitig alle anderen Knoten in $\mathcal{O}(\log n)$ vielen Suchschritten finden kann.

Pastry [RD01] nutzt eine ähnliche Ring- und Querverbindungstopologie wie Chord; es erlaubt allerdings ein Routing in beide Ringrichtungen. Dadurch ist die zu minimieren-

de Overlay-Metrik als absolute Differenz zwischen der Kennung eines Knotens und des Hash-Wertes der Suchanfrage definiert. Ferner baut Pastry zusätzliche Verbindungen auf, die in Bezug auf das Transportnetz in der Nähe des Knotens liegen. Die Nähe des nächsten Knotens fließt ferner in die Routing-Entscheidung mit ein, sodass aufgrund der lokaleren Suche, diese im Mittel schneller erfolgt.

Unter anderem durch Koorde [KK03] und Symphony [MBR03] wurde der Chord-Ansatz verfeinert. Durch das effizientere Platzieren von Querverbindungen werden hier nur konstant viele Verbindungen pro Knoten benötigt, um Suchen in $\mathcal{O}(\log n)$ Overlay-Schritten zu garantieren. Beide Ansätze sind jedoch darauf angewiesen, dass in jedem Schritt optimale Entscheidungen getroffen werden. Inkonsistenzen in der Routing-Struktur haben somit viel stärkere Auswirkungen.

Neben den vorgestellten ringbasierten Verfahren existieren eine Reihe weiterer Topologien mit ähnlichen Aufwänden und Geschwindigkeiten. Viceroy [MNR02] setzt beispielsweise ein so genanntes Butterfly-Netzwerk ein, um ähnlich wie Koorde und Symphony in logarithmischer Zeit zu routen, obwohl nur konstant viele Verbindungen aufgebaut werden. Auch hier ist es extrem wichtig, dass jeder Knoten eine genaue Sicht seiner Umgebung besitzt.

Kademlia [MM02] stellt ein System dar, bei dem sich Knoten auf den Ecken eines Hyperwürfels befinden. Die Suchen entlang der Kanten werden in diesem Ansatz so weitergeleitet, dass die XOR-Verknüpfung von Suchschlüssel und Kennung des nächsten Knotens in jedem Schritt minimiert wird.

Im Content Addressable Network (CAN) [RFH+01] wählt sich jeder Knoten beispielsweise einen Punkt in einem d-dimensionalen Hyperwürfel und speichert Informationen, deren Koordinaten (ebenfalls aus d Hash-Funktionen gewonnen) sich im Hyperraum um ihn herum befinden. Das Routing basiert auf euklidischer Distanzminimierung und benötigt bei $\mathcal{O}(d)$ Nachbarn $\mathcal{O}(\sqrt[d]{n})$ Overlay-Schritte.

Auch wenn die vorgestellten verteilten Systeme zunächst nur zur effizienten Speicherung und Suche in großen Datenmengen konzipiert wurden, ergibt sich ein Zusammenhang zu Routing-Verfahren in VPN. So existieren Verfahren, wie beispielsweise die im nächsten Kapitel vorgestellte Internet Indirection Infrastructure ($i3$) [SAS+02, ZLS+05], welche die nach den vorgestellten Ansätzen aufgebauten Overlays zur Punkt-zu-Punkt-Kommunikation nutzt. Dabei können an Stelle der Suchnachrichten Datenpakete über das Overlay weitergeleitet werden.

2.4.2 SkipNet und Skip–Graphen

Ebenso wie die bereits vorgestellten DHT-Ansätze dienen auch SkipNet [HJS+03] und die parallel entwickelten Skip-Graphen [AS07, JRS+09] der Indizierung verteilter Datensätze. Im Gegensatz zu den DHT werden die in SkipNet abgelegten Daten jedoch nicht gestreut, sondern Teilnehmer anhand ihrer strukturierten Schlüssel direkt in eine Ringstruktur eingebunden. Das Verfahren erlaubt so komplexere Suchanfragen, wie beispielsweise Longest prefix matches. Um dennoch vergleichsweise schnell suchen zu können, wird die Zugehörigkeit des Knotens in SkipNet auf eine Hierarchie von

Ringen mit logarithmisch abfallenden Größen abgebildet. Die ebenfalls logarithmische Tiefe der Hierarchie ergibt sich aus der Konstruktion: Jeder Knoten enthält neben seinem Schlüssel eine zufällige Kennung, die definiert, in welchen Ringen er partizipiert. Neben dem Ring, der alle Knoten umfasst, ist beispielsweise ein Knoten mit der Kennung `101000` Teil des Rings `1*`, welcher im statistischen Mittel lediglich die Hälfte der Knoten enthält. Wie in Abbildung 2.6 auf die aus Chord bekannte Art der Illustration übertragen, beteiligt er sich darüber hinaus in den Ringen `10*` und `101*`. Ausreichend lange Kennungen vorausgesetzt, können so alle Knoten – trotz lexikographischer Ordnung – statistisch in einer logarithmischen Anzahl von Overlay-Schritten erreicht werden.

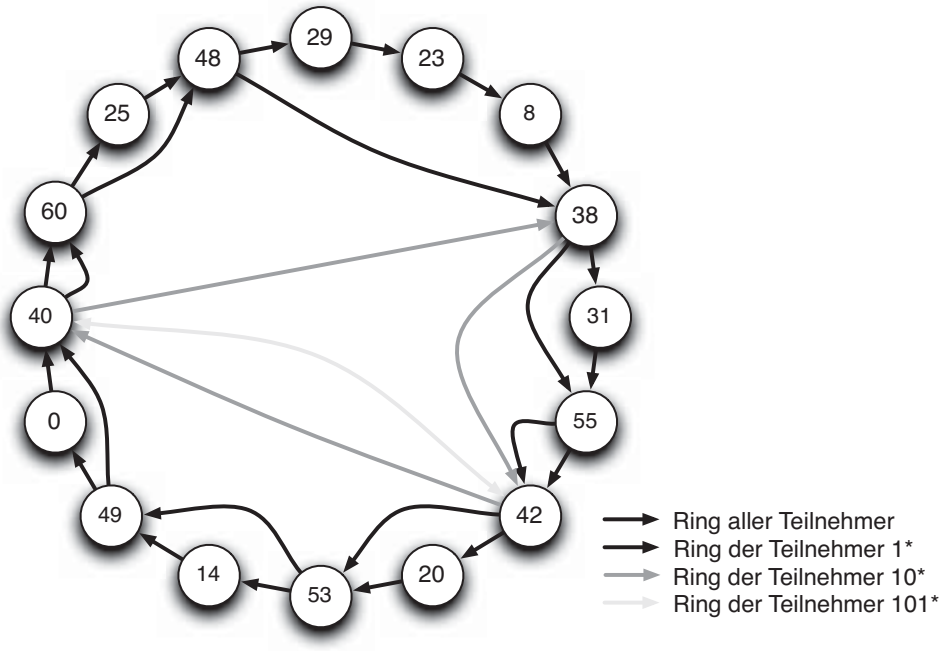


Abbildung 2.6: Beispiel für die von einem Skip-Graph-Algorithmus induzierten Ringstrukturen, an denen der Knoten mit der Zufallskennung 40 (`101000`) teilnehmen

2.4.3 Peer–to–Peer–basierter Multicast

Neben den vorgestellten Peer-to-Peer-Systemen zur Indizierung von Daten, wurden, bedingt durch die schleppende Einführung eines global verfügbaren IP-Multicast-Dienstes, so genannte *Application Layer Multicast (ALM)* Ansätze stark durch die Wissenschaft untersucht [HASG07]. Dabei leiten Endsysteme Multicast-Ströme über einzelne Unicast-Verbindungen untereinander weiter, um zu vermeiden, dass die Quelle die Daten an jeden einzelnen Teilnehmer versenden muss. Von Anfang an wurden dabei auch Ansätze entwickelt, die auf den bereits vorgestellten, strukturierten DHT basieren (beispielsweise [CDKR02, RHKS01]). Für VPN bietet sich durch ALM die Möglichkeit, Multicast über Unicast-Verbindungen zu realisieren und so eine Reihe von Sicherheitsproblemen, die mit der Nutzung von Gruppenschlüsseln einhergehen, zu umgehen [HD03].

Im Allgemeinen werden heute zwei grundlegende ALM-Prinzipien unterschieden. Zum einen existieren *Tree first*- oder auch *Push*-Systeme, bei denen von der Quelle weg ein expliziter Verteilbaum aufgebaut wird. Diese Baumstruktur bleibt grundsätzlich über einen längeren Zeitraum bestehen und wird nur durch sukzessive Optimierungen oder Knotendynamik verändert. Zum anderen werden Ansätze entwickelt, die als *Mesh-first*, *Gossip basierte* oder *Pull*-Systeme bezeichnet werden [KRAV03, ZLLY05]. Das Konzept ist hier geringfügig anders: Es wird zunächst eine vermaschte Topologie zwischen den Overlay-Teilnehmern aufgebaut. Die einzelnen Overlay-Teilnehmer suchen nun regelmäßig nach anderen Teilnehmern, die sowohl einen neuen Datenblock als auch freie Übertragungskapazitäten besitzen, und fordern gegebenenfalls die gefundenen Blöcke an. Je nach Ansatz kann die eingesetzte Topologie entweder strukturiert oder unstrukturiert sein, die Konsequenz ist lediglich, dass die Suche bei ersteren gerichtet erfolgt. In jedem Fall wird die zur Übertragung eines Datenblocks genutzte Topologie wieder baumförmig, wobei die Baumstrukturen für jeden Block unterschiedlich seien können.

Welche der beiden Prinzipien sich besser zur Realisierung von Multicast-Strömen eignet ist umstritten. Während Pull-Verfahren potentiell robuster sind und Bandbreiten besser ausnutzen, können die Topologien von Push-Ansätzen in der Regel besser bezüglich Verzögerungszeiten optimiert werden. In [HLL+07] werden beispielsweise für das real existierende Pull-System PPLive Verzögerungen von teilweise über zwei Minuten angegeben. Dies mag zwar zum einen auf eine nicht ausreichende Topologieoptimierung zurückzuführen sein, zum anderen erfordert die Aufteilung der Daten in Blöcke eine nicht unerhebliche Pufferung in jedem Übertragungsschritt.

2.4.4 Netzwerkkoordinatensysteme

Unabhängig von den Aufgaben der Overlay-Netze besteht häufig das Problem, für einen Knoten den Abstand zwischen zwei anderen Knoten abzuschätzen. Systeme wie Global Network Positioning (GNP) [NZ02] oder Practical Internet Coordinates for Distance Estimation (PIC) [CCRK03] schlagen zur Lösung die Nutzung künstlicher Koordinatensysteme vor. In beiden Systemen werden einige Knoten des Overlays mit festen Positionen konfiguriert; sie bilden so genannte Landmarken. Alle anderen Knoten berechnen eigene Koordinaten, indem sie die Verzögerung zu diesen Landmarken messen und diese korrelieren.

Die vorgeschlagenen Ansätze funktionieren zunächst recht gut. Allerdings schränken sie Robustheit, Skalierbarkeit und Verfügbarkeit ein, da sie exponierte Punkte einführen. Gerade bei VPN sind Verfügbarkeitseigenschaften von signifikanter Wichtigkeit, sodass der neuere verteilte Vivaldi-Algorithmus eine bessere Alternative darstellt. Dieser wurde zunächst in [CDK+03] vorgeschlagen und basiert auf einem Feder-Masse-Modell. Knoten messen zu ihren Nachbarn periodisch Verzögerungszeiten und korrigieren ihre Koordinaten, indem sie sich je nach gemessenem Fehler entweder anziehen oder abstoßen. In diesem ersten Vorschlag wurde dazu ein relativ hochdimensionaler euklidischer Raum verwendet, welcher aber aufgrund des „Fluchs der Dimensionalität“

vergleichsweise wenig Aussagekraft bietet. Letzterer beschreibt den Umstand, dass Werte mit sehr vielen Dimensionen aufgrund des größeren Wertebereichs schlechter zu vergleichen sind.

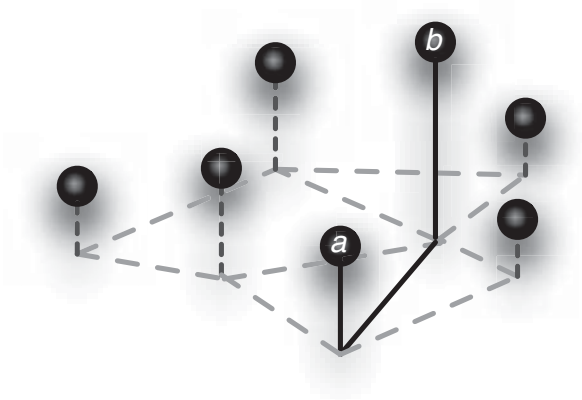


Abbildung 2.7: Zusammengesetzte Distanzmetrik in Vivaldi: zweidimensionale euklidische Grundfläche und auf Manhattan-Distanz basierende Höhenkoordinate

Wie in Abbildung 2.7 dargestellt, wurde in [DCKM04] eine Verfeinerung vorgenommen: An Stelle des mindestens fünfdimensionalen euklidischen Raumes wird eine zusammengesetzte Metrik aus einem zweidimensionalen euklidischen Abstand und einer Manhattan-Distanz gebildet. Letztere dient als Höhenkoordinate zur Abbildung der Verzögerung bis zum Kernnetz, während die euklidische Komponente den Abstand der jeweilig zuständigen Router modelliert. Auf diese Weise lassen sich Internet-Verzögerungen effektiv verteilt abschätzen.

2.4.5 Spezielle Angriffe auf Peer–to–Peer–Systeme

Im Zuge der Entwicklung strukturierter Peer-to-Peer-Systeme sind auch eine Reihe potentieller Sicherheitsbedrohungen diskutiert worden [UPS09], die beim Entwurf von Overlay-Netzen berücksichtigt werden müssen. Es wird in drei wichtige Angriffsstrategien unterschieden:

- Unter einem *Sybil-Angriff* versteht man die mehrfache Teilnahme eines Angreifers unter verschiedenen Identitäten [Dou02]. Auf diese Weise sollen in der Regel Mehrheitsentscheidungen beeinflusst oder einfach nur die Wahrscheinlichkeit des Erfolges anderer Angriffe erhöht werden.
- *Eclipse-Angriffe* bezeichnen einen Vorgang bei dem ein Angreifer versucht, einen Großteil oder die gesamte Nachbarschaft bestimmter Overlay-Knoten zu kontrollieren [SNDW06]. In Verbindung mit dem Sybil-Angriff kann diese Strategie auch von einem einzelnen Angreifer verfolgt werden.
- Sowohl Sybil- als auch Eclipse-Angriffe dienen in der Regel zur Vorbereitung von *Routing-Angriffen*, bei denen Angreifer Suchanfragen falsch, verzögert oder gar

nicht beantworten. Sie können auch durch verstärkten Aufbau von Verbindungen oder durch falsches Beantworten von Suchanfragen besonders viel Verkehr auf sich ziehen (*Sinkhole-Angriffe*). Werden keine weiteren Sicherungsmechanismen eingesetzt, können die dadurch abgefangenen Nachrichten auch mitgelesen oder manipuliert werden.

In a world of uncertainty in which nobody knows the right answer, you need to try out a lot of things and hope you will find one that works.

(Douglass North)

Kapitel 3

Stand von Wissenschaft und Technik

In diesem Kapitel werden zunächst die Anforderungen an VPN und ihre Autokonfigurationssysteme strukturiert dargestellt, um in den weiteren Abschnitten bekannte VPN-Konfigurationssysteme und Overlay-Netze zur Steigerung von Robustheit und Verfügbarkeit, sowie strukturiert routende Netze auf dieser Grundlage zu analysieren. Die festgestellten Diskrepanzen zwischen Anforderungen und existierenden Systemen werden abschließend in einem Fazit überblicksartig zusammengefasst.

3.1 Anforderungen an VPN–Autokonfigurationssysteme

Die im Folgenden vorgestellten Anforderungen an selbstorganisierende VPN sind wiederum in vier Unterabschnitte untergliedert: Zunächst wird auf funktionale und wesentliche nichtfunktionale Anforderungen eingegangen, um anschließend Anforderungen in Bezug auf Sicherheit im Allgemeinen und Verfügbarkeit im Speziellen detailliert aufzuschlüsseln.

3.1.1 Funktionale Anforderungen

VPN-Konfigurationssysteme sollen folgende Ansprüche erfüllen:

1. **Selbstkonfiguration:** Sowohl Aufwand als auch Komplexität der manuellen Aspekte einer Konfiguration von VPN sollen minimiert werden [Gen01]. Im besten Fall bedeutet dies, dass jeder VPN-Knoten lediglich mit einem Zertifikat ausgestattet wird und sich anschließend vollautomatisch zu anderen Teilnehmern des VPN verbindet.

2. **Konfiguration in gebräuchlichen Transportnetzen:** Von VPN-Konfigurationssystemen wird erwartet, VPN-Knoten mit ihren gegebenenfalls angeschlossenen Teilnetzen zu verwalten und sichere Tunnel durch Internet-ähnliche Netze aufzubauen. Dabei sind folgende Punkte zu berücksichtigen:

a) **Gateway-Funktionalität:** Wie bereits angedeutet sollen nicht nur einzelne Endpunkte zu konfigurieren sein, sondern auch VPN-Gateways mit ihren privaten Teilnetzen [KS05]. Einzelne Endpunkte stellen dann nur einen Sonderfall dar, da sie VPN-Gateway und ein „Teilnetz“ in einem Gerät vereinen.

b) **Verschachtelte VPN-Strukturen:** Die VPN-Knoten müssen dabei nicht zwangsläufig direkt miteinander kommunizieren können, da sie sich zum einen in verschiedenen Transportnetzen befinden können, welche lediglich über andere VPN-Knoten gekoppelt sind [KS05]. Zum anderen kann die Kommunikation zwischen einzelnen VPN-Knoten innerhalb des Transportnetzes durch Ausfälle oder DoS-Angriffe gestört sein. Daher soll zumindest in diesen Fällen eine Weiterleitung über andere VPN-Knoten erfolgen.

c) **Verwendung von Unicast-Kommunikation im Transportnetz:** Das Konfigurationssystem soll einen Aufbau von VPN-Tunneln über generische Transportnetze erlauben. Daher kann lediglich von einer Unicast-Unterstützung ausgegangen werden, und es kann insbesondere keine Any-, Multi- oder Broadcast-Unterstützung vorausgesetzt werden. Letztere sind in globalen Netzen in der Regel nicht verfügbar [DLL+00].

d) **Umgang mit Network Address Translation (NAT):** Mobile Teilnehmer sind oft nur über NAT-Gateways mit dem Internet verbunden; ein VPN-Konfigurationssystem sollte auch die Einbindung solcher Teilnehmer und im Idealfall auch die Aushandlung direkter Verbindungen zwischen ihnen unterstützen [AD04].

3. **Verwendung privater Adressbereiche:** Die durch das VPN verbundenen Teilnetze werden oft mit privaten IP-Adressbereichen konfiguriert [KK04, Lip07], da das Verwenden öffentlicher Adressen zu aufwendig, zu teuer oder zu unflexibel ist. Darüber hinaus werden private IP-Adressen oft als eine zusätzliche Sicherheitsbarriere angesehen, da sie helfen, die Größe der privaten Teilnetze nach außen zu verbergen. VPN müssen ein eigenes Routing implementieren, um in solchen Szenarien zu funktionieren.

4. **Multicast-Unterstützung innerhalb des VPN:** Auch wenn noch keine allgemein zugänglichen, globalen Multicast-Netze verfügbar sind, so bietet die Verwendung von Multicast in Netzen, die von einer einzelnen Organisation betrieben werden, einige Vorteile [BKW05]. Neben Video-Streaming-Daten können beispielsweise auch Software-Updates über Multicast effizient verteilt werden. Ein Autokonfigurationsansatz sollte dies daher ebenfalls berücksichtigen.

5. **Transparenz:** Der Einsatz von VPN soll vor den Nutzern und beim Einsatz von VPN-Gateways auch vor den Geräten in deren privaten Teilnetzen so gut wie möglich verborgen bleiben [Lip07]. Insbesondere sollen keine Anpassungen und Rekonfigurationen notwendig sein.

3.1.2 Nichtfunktionale Anforderungen

Neben den vorgestellten Anforderungen, die sich direkt in einem Teilsystem eines Autokonfigurationsansatzes realisieren lassen, existieren weitere Anforderungen, welche die Handhabbarkeit und damit letztlich die Qualität des Ansatzes erfassen. Dies sind in Bezug auf VPN-Autokonfiguration:

1. **Skalierbarkeit:** Da die Anwendung automatisierter Konfigurationssysteme vor allem in großen VPN sinnvoll ist, müssen auch Netze mit mehreren tausend VPN-Knoten unterstützt werden [Gen01, Lip07]. Aus diesem Grund sollten Engpässe vermieden und die Anzahl der proaktiv aufgebauten Kommunikationsbeziehungen so gering wie möglich gehalten werden.
2. **Robustheit:** Damit VPN unter jeglichen Umständen bestmöglich funktionieren, müssen auch extreme Ereignisse, wie der gleichzeitige Start oder Ausfall vieler VPN-Knoten berücksichtigt werden. Im Fall von Netzwerkpartitionierungen die Partitionen unabhängig voneinander weiter operieren und anschließend wieder zusammengeführt werden. Ferner soll bei partiellen Ausfällen Verkehr über nicht ausgefallen Punkte umgeleitet werden [CM02].
3. **Agilität:** Automatisierte Konfigurationslösungen bieten die Gelegenheit, selbstständig auf Veränderungen im VPN oder dem darunter liegenden Transportnetz zu reagieren. Diese Rekonfigurationen sollen dabei möglichst schnell erfolgen, sodass beispielsweise mobile Teilnehmer nach einem Handover schnell wieder innerhalb des VPN erreichbar sind [ZF10].
4. **Effizienz:** Schon um die Skalierbarkeit des VPN zu gewährleisten, sollen Konfigurationsmechanismen die Anzahl der versendeten beziehungsweise empfangenen Verwaltungsnachrichten und die Anzahl der proaktiv aufgebauten Sicherheitsbeziehungen minimieren. Die Längen der einzelnen, durch das VPN aufgebauten Pfade müssen ferner minimiert werden, um unnötige Umverschlüsselungen zu vermeiden, die Netzwerkbelastung zu senken und die Ende-zu-Ende-Verzögerung zu minimieren.

3.1.3 Sicherheitsanforderungen

Die durch VPN realisierten Sicherheitseigenschaften dürfen durch eine automatische Einrichtung nicht negativ beeinflusst werden. Im Einzelnen existieren die folgenden Anforderungen, welche sich aus den allgemeinen Sicherheitszielen in Abschnitt 2.1 ableiten:

1. **Vertraulichkeit:** Das wichtigste Ziel beim Aufbau von VPN stellt in der Regel die kryptographische Sicherung der übertragenen Daten vor externer Kenntnisnahme dar [Gen01, Lip07]. Daneben existieren die folgenden Unterziele:
   a) **Ende-zu-Ende-Sicherung:** Besonders bei der Weiterleitung von Daten über mehrere VPN-Knoten hinweg ist es wichtig, die Vertraulichkeit zwischen den Kommunikationsendpunkten herzustellen, sodass auch potentielle interne

Angreifer (zum Beispiel mithilfe kompromittierter VPN-Gateways) keine Kenntnis über indirekt übertragene Daten erhalten können [KS05].

b) **Perfect Forward Secrecy (PFS):** Die für die Verschlüsselung der Daten eingesetzten Sitzungsschlüssel sollen auf eine Art und Weise erzeugt werden, sodass auch ein späteres Kompromittieren der langlebigeren Geheimnisse (wie den privaten Schlüsseln) eines VPN-Knotens keine Rekonstruktion eines Sitzungsschlüssels erlaubt [DOW92]. Auf diese Weise ist die Vertraulichkeit aller Daten, die vorher übertragen wurden, gewährleistet.

c) **Resistenz gegen verdeckte Kanäle:** Vor allem VPN-Gateways, aber möglicherweise auch das Betriebssystem eines einzelnen VPN-Endpunktes, setzen gegenüber dem Nutzer oftmals mittels Zwangsverschlüsselung eine Sicherheitspolitik um. Diese Sicherheitspolitik soll verhindern, dass Daten gewollt oder ungewollt das VPN verlassen. Automatische Konfigurationsansätze für VPN dürfen verdeckte Kanäle nicht begünstigen [GT03].

d) **Infrastructure Hiding:** In vielen Fällen soll von VPN auch die genaue Anzahl und Struktur der Teilnetze, beziehungsweise auch die Struktur des VPN selbst, verschleiert werden [Bro01, Beh06]. Auf diese Weise wird beispielsweise die Planung von DoS-Angriffen erschwert.

e) **Anonymität:** Einige spezielle VPN-Ansätze wie The Onion Router (TOR) [DMS04] oder Freenet [CSWH01] versuchen, auch die Identität der einzelnen Teilnehmer eines VPN untereinander vertraulich zu behandeln. Aufgrund des großen Einflusses auf die Effizienz des VPN, zum Beispiel durch das Erzeugen von Cover Traffic oder den Einsatz von Onion Routing, wird das Teilziel Anonymität im Folgenden allerdings nicht berücksichtigt werden können.

2. **Datenintegrität und –authentizität:** Die übertragenen Nutzdaten müssen im VPN so geschützt werden, dass die Pakete dem VPN zugeordnet werden und gleichzeitig potentielle Manipulationen Dritter ausgeschlossen werden können [Lip07]. In vielen VPN ist es darüber hinaus ein Ziel, dass Datenpakete einem bestimmten VPN-Knoten zugeordnet werden können, wobei bei indirekter Kommunikation dann auch eine Ende-zu-Ende-Sicherung erfolgen muss.

3. **Instanzenauthentisierung:** Schon zur Gewährleistung der Datenauthentizität müssen sich Kommunikationspartner innerhalb eines VPN sicher gegenseitig identifizieren können [Gen01]. Beim Einsatz von IP-Netzwerken schließt eine Authentisierung die private IP-Adresse eines VPN-Knotens, beziehungsweise bei Gateways dessen gesamte privaten IP-Adressbereiche, mit ein. Diese stellen für Zwischensysteme, Endgeräte und letztendlich für die Nutzer die einzige Möglichkeit dar, kryptographische Endpunkte zu differenzieren. Bei VPN, die auf Datensicherungsschicht arbeiten, muss analog die Adresse dieser Schicht authentisiert werden und zusätzlich die Abbildung auf die Adressen der höheren Schichten.

4. **Kontrollierter Zugang:** Eine erfolgreiche Authentisierung eines Kommunikationspartners vorausgesetzt, soll dessen Berechtigung am VPN teilzunehmen durch zwei Dienste sichergestellt werden:

   - Die **statische Zugriffskontrolle** stellt sichere Verbindungen nur zu VPN-Knoten her, die eine Berechtigung in Form eines gemeinsamen Geheimnisses oder eines Zertifikates einer dedizierten Certificate Authority (CA) besitzen [Lip07].
   - Darüber hinaus wird eine **dynamische Zugriffskontrolle** benötigt, um die Aktualitätsprüfung der Zugriffsberechtigung zu realisieren. Bei Punkt-zu-Punkt-Verbindungen mit paarweisen Schlüsseln wird dazu in der Regel eine periodische Reauthentisierung vorgenommen [Nir06]. Werden jedoch Gruppenschlüssel verwendet, müssen diese Schlüssel bei jedem Beitritt oder Verlassen eines Gruppenmitgliedes geändert werden, um *Forward* und *Backward Access Control* zu gewährleisten [WR09].

   Da IP-basierte VPN auf Netzwerkebene arbeiten und kein Wissen über die ausgeführten Anwendungen besitzen, sind über diese Maßnahmen hinausgehende Zugriffskontrollmechanismen an dieser Stelle nicht umzusetzen. Primärer Grund hierfür ist die Vermischung von Daten- und Kontrollfluss in den darüberliegenden Schichten, die immer eine bidirektionale Kommunikation erforderlich machen. Modelle für die Informationsflusskontrolle, wie beispielsweise [Den76], benötigen allerdings für eine sinnvolle Anwendung unidirektionale Flüsse, die durch Anwendungen oder Betriebssystem auf den Endsystemen realisiert werden müssen, wie etwa in [MJB+06]. In verteilten Szenarien stellt weiterhin der Schutz der Trusted Computing Base (TCB) ein Problem dar; bereits ein einzelner kompromittierter Knoten genügt, um Zusagen über den Zugriffsschutz oder Obligationen, wie in [MAHS10] vorgeschlagen, in Frage zu stellen. Aus diesen Gründen versuchen aktuelle VPN-Autokonfigurationsansätze eine Informationsflusskontrolle, die über ein angestrebtes Non-Interference-Modell [HS04] hinausgeht, nicht zu realisieren.

5. **Minimale sicherheitsrelevante Konfiguration:** Unter anderem in [Del09] werden menschliche Fehler als ein Hauptproblem der IT-Sicherheit identifiiziert. Daher sollen auch VPN-Autokonfigurationssysteme den manuellen Arbeitseinsatz nicht nur aus Kostengründen minimieren, sondern dabei gleichzeitig auch die Sicherheit erhöhen, indem menschliche Fehler vermieden werden.

3.1.4 Anforderungen in Bezug auf die Verfügbarkeit

Wie bereits in den Grundlagen ausgeführt wird, ist Verfügbarkeit in der Regel als ein Unterziel der allgemeinen Sicherheitsziele modelliert [Sch03, Eck09]. Dabei handelt es sich in diesem Kontext um die Eigenschaft, auch in Gegenwart extremer, durch Angreifer verursachter Bedingungen bestmöglich weiter zu operieren. Die einzelnen Verfügbarkeitsanforderungen werden in diesem Abschnitt aufgrund der detaillierten Betrachtungen gesondert behandelt.

Resistenz gegen Sabotageangriffe

Sabotageangriffe [Lip07], im Kontext von IP-Netzwerken in der Regel als DoS-Angriffe bezeichnet, durch Abweichung vom normalen Protokoll oder durch Überlastungen bereit gestellte Dienste zu überlasten. Im Kontext von VPN bedeutet dies, dass die Pakete der Nutzer nicht oder nur eingeschränkt weitergeleitet werden. Die Klasse von Angriffen stellen eine ernst zunehmende Gefahr da, weil sie in der Regel einfach zu realisieren sind und gegen eine Reihe von Unterarten keine direkten Schutzmechanismen existieren. Nach [Mil92] wird primär in zwei Angriffstypen unterschieden:

- **Zerstörung von Ressourcen:** Beispielsweise durch physische Attacken, das Ausnutzen von Software-Schwachstellen oder Rechtemissbrauch können Angreifer unter Umständen VPN-Knoten oder Teile des Transportnetzes unbrauchbar machen. Gegen solche Angriffe helfen bekannte Sicherungsmaßnahmen, wie physischer Schutz oder Redundanz, die Härtung von Software-Komponenten und der Einsatz von vertrauenswürdigem Personal. Eine Einflussnahme eines Autokonfigurationsansatzes ist nur insofern möglich, dass durch ihn zum einen kein Single Point of Failure entstehen darf, und es dem Angreifer zum anderen so schwer wie möglich fallen soll, relevante Punkte des VPN zu identifizieren.
- **Erschöpfung von Ressourcen:** Einem Angreifer ist es potentiell möglich, Systemressourcen entweder durch eine einmalige Reservierung oder durch ständige Anfragen zu belegen. Diese Ressourcen stehen für legitime Systeme nicht mehr zur Verfügung, und insbesondere bei einer Überlastung durch ständige Anfragen ist eine Lösung der Situation oft schwierig. Angreifer können im Prinzip die folgenden Ressourcen überlasten:
  - **Rechenzeit:** Durch das häufige Auslösen rechenintensiver Operationen, zum Beispiel beim Einsatz asymmetrischer Kryptographie, können legitime Anfragen unter Umständen nicht korrekt bearbeitet werden. Sind für die Operationen Smartcards vorgesehen, wie es in Hochsicherheitsumgebungen der Fall ist, verschärft sich das Problem weiter, da diese über extrem begrenzte Ressourcen verfügen. Detaillierte Messungen finden sich dazu in Abschnitt A.1. Einen möglichen Ausweg stellen Crypto Puzzles [JB99, ANL01] dar, bei denen anfragende Systeme zunächst selbst Rechenzeit investieren müssen, um eine Anfrage generieren zu können.
  - **Speicher:** Ähnlich verhält es sich bei der Speicherauslastung: Reserviert ein System für jede einzelne Anfrage zunächst selbst relativ wenig Speicher, können Angreifer durch das Stellen sehr vieler Anfragen diesen schnell ausschöpfen. Auch gegen diese Art des Angriffs existiert eine Gegenmaßnahme, die zumindest in einigen Szenarien Abhilfe verschafft. Durch die Verlagerung sämtlicher Zustandsinformationen in die übertragenen, kryptographisch gesicherten Netzwerkpakete, kann eine Speicherreservierung bis zur Authentisierung des Gegenübers vermieden werden. Dieser so genannte Cookie-Mechanismus eignet sich insbesondere, um Angriffe mit gefälschten Adressen zu unterbinden [Kar94, Edd07].

– **Netzwerkbandbreite:** Wenn Ressourcenzerstörung, Speicher- oder Prozessorauslastung nicht möglich sind, beziehungsweise das Opfer über einen schmalbandigen Netzwerkanschluss verfügt, besteht ein einfacher Angriff in der Generierung einer großen Anzahl von Datenpaketen und der damit verbundenen Netzwerküberlastung. Sollten Angreifer relevante IP-Adressen kennen, ist ohne Einwirken durch den ISP kein Schutz gegen solche Angriffe möglich, da der Schaden bereits durch das Erreichen des VPN-Knotens ausgelöst werden kann. In Bezug auf VPN ist dieser Angriff besonders effektiv, da lediglich die VPN-Knoten selbst anhand kryptographischer Verfahren entscheiden können, welche Pakete legitim sind. Daher ist der Standardansatz der Ingress-Filterung nicht oder nur mit noch höherem Administrationsaufwand möglich, wenn einzelne Sicherheitsbeziehungen beim ISP freigeschaltet werden. Die Erschöpfung von Netzwerkbandbreite verbleibt aufgrund des schwierigen Schutzes, des niedrigen technischen Niveaus des Angriffes und der möglichen Schäden durch anhaltenden Ausfall des VPN, das ernstzunehmendste Problem in Bezug auf die Verfügbarkeit.

Da VPN-Autokonfigurationssysteme nach Möglichkeit ohne Einbindung des ISP funktionieren sollten, sind die einzigen universellen Maßnahmen gegen Sabotageangriffe somit eine Dezentralisierung und eine Verschleierung der VPN-Infrastruktur. Mechanismen wie Cookies und Client Puzzles [ANL01] müssen bereits auf Ebene des zugrunde liegenden kryptographischen Protokolls wie IPsec oder TLS implementiert werden.

Graceful Degradation

Das Kompromittieren einzelner Systeme des VPN soll möglichst lokale Auswirkungen haben, und nicht betroffene Knoten und Client-Geräte sollen — soweit möglich — ohne Beeinträchtigung der Sicherheit weiterarbeiten [PSDY00].

DoS–Recovery

Sind Teile eines VPN von einem Sabotageangriff betroffen, so sollen die verbleibenden VPN-Knoten automatisch ihre Topologie an die neuen Gegebenheiten anpassen und Daten über nicht betroffene Bereiche weiterleiten [PSDY00]. Zudem soll es möglich sein, betroffene VPN-Knoten über andere IP-Adressbereiche schnellstmöglich wieder in das Netz zu integrieren.

3.2 Existierende Systeme zur Autokonfiguration von VPN

In den folgenden Abschnitten wird für einzelne Autokonfigurationsmechanismen betrachtet, inwiefern sie die aufgestellten Anforderungen berücksichtigen. Dazu werden

sie in drei Kategorien unterteilt: Zunächst werden zentralisierte Ansätze vorgestellt, bevor im zweiten Unterabschnitt einige dezentralisierte Systeme analysiert werden. Diese besitzen keinen Single Point of Failure, wenngleich einige dedizierte Netzwerkkomponenten existieren. Im letzten Unterabschnitt werden die unterschiedlichen vollständig verteilt arbeitenden Ansätze vorgestellt.

3.2.1 Zentralisierte VPN–Konfigurationssysteme

Die in den folgenden Abschnitten vorgestellten drei Ansätze besitzen alle einen zentralen VPN-Konzentrator, Schlüssel- oder Verwaltungsserver. Diese können zwar zum Teil physisch redundant ausgelegt werden, sind aber durch den jeweiligen Ansatz in Anzahl von Geräten und Standorten fest begrenzt.

Cisco Easy VPN

Hinter dem Markennamen Easy VPN, teilweise auch in der Schreibweise EzVPN, vermarktet Cisco eine Konfigurationslösung zur Einrichtung von Remote Access beziehungsweise einfachen Hub-and-Spoke-Szenarien auf Basis von IPsec [Bha08]. Einzelne VPN-Endpunkte oder VPN-Gateways verbinden sich in Easy VPN zu einem zentralen VPN-Konzentrator, welcher Konfigurationsaufgaben wie die Vergabe von IP-Adressen und die Weiterleitung aller Datenpakete übernimmt. Für viele Administratoren ist die Möglichkeit, VPN-Nutzer über IKE XAUTH [BP01] zu authentisieren ein funktionaler Vorteil, weil so ein Abgleich von Nutzernamen und Passwort über Remote Authentication Dial In User Service (RADIUS) ermöglicht wird.

Aus funktionaler Sicht kann Easy VPN private IP-Adressbereiche in den privaten Teilnetzen konfigurieren, bietet eine VPN-Gateway-Unterstützung und funktioniert in Unicast-Netzen. Easy-VPN-Klienten können sich hinter NAT-Gateways befinden, wobei eine direkte Kommunikation zwischen von NAT betroffenen Geräten nicht vorgesehen ist, da jegliche Paketweiterleitung ohnehin über den Konzentrator stattfindet, der zwangsläufig über eine öffentliche IP-Adresse verfügen muss. Trotz starker Vereinfachungen bei der Administration kann gerade die Anbindung von VPN-Gateways noch relativ komplex werden, da sehr viele Feineinstellungen nötig sind. Ferner können verschachtelte Szenarien, sowie Multicast innerhalb der privaten Netze nicht realisiert werden.

Zusätzlich zu den allgemeinen Problemen zentraler Ansätze, müssen in Easy VPN nach einem zeitweiligen Ausfall des VPN-Konzentrators erst alle Sicherheitsbeziehungen wieder aufgebaut werden, wodurch die Robustheit und Verfügbarkeit weiter eingeschränkt werden.

Aber auch die Sicherheitsziele werden nicht nur durch den gewählten zentralen Ansatz negativ beeinflusst. Easy VPN bietet sowohl in Bezug auf Vertraulichkeit als auch in Bezug auf Datenauthentizität keine Ende-zu-Ende-Sicherung, sodass der VPN-Konzentrator alle Daten im Klartext erhält, selbst wenn sie nur zwischen zwei Klienten

umgeleitet werden. Ein Kompromittieren des Konzentrators führt daher VPN-weit zu einem Verlust aller Sicherheitseigenschaften.

Ein weiteres verbreitetes Problem von Easy VPN Installationen ist die Möglichkeit von Man-in-the-Middle-Angriffen auf das XAUTH-Protokoll [Cis06a]. Das Sicherheitsproblem ist zwar seit über fünf Jahren bekannt, allerdings existieren immer noch viele verwundbare Installationen und ein aktiv gepflegtes Angriffswerkzeug [Roe10]. Gründe hierfür sind wohl eine fehlende Aufklärung durch Cisco; so zeigen mehrere Anleitungen auf Ciscos Webseiten eine unsichere Easy VPN Einrichtung [Cis04], sowie die wesentlich komplexere Konfiguration sicherer Easy VPN Systeme.

Cisco Group Encrypted Transport (GET) VPN

Unter dem Namen GET VPN [Cis07, Bha08] vermarktet Cisco einen weiteren zentralisierten Autokonfigurationsansatz, welcher eigentlich ein Konglomerat aus einer Reihe von Techniken und Standards, wie beispielsweise Group Domain of Interpretation (GDOI) [BHHW03], darstellt. Das primäre Ziel von GET ist die Konfiguration extrem großer, vollvermaschter VPN (bis zu 10.000 Endpunkten), bei denen alle VPN-Gateways über ein von einem ISP gemietetes Netz verbunden sind.

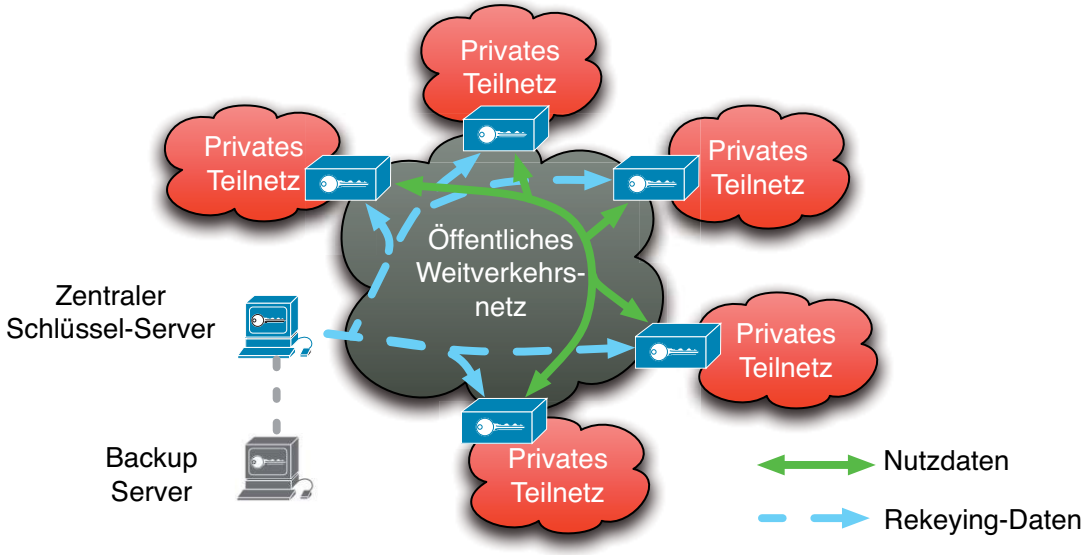


Abbildung 3.1: Beispielszenario für ein GET VPN

Abbildung 3.1 illustriert ein GET VPN, welches wie folgt konfiguriert wird:

1. Dem VPN beitretende Gateways kontaktieren zunächst einen festkonfigurierten Schlüssel-Server und führen mit ihm eine Authentisierung und Autorisierung auf Basis von IKE aus. Sollte der Schlüssel-Server nicht verfügbar sein, kann einer von maximal sieben Backup-Servern verwendet werden, sofern diese ebenfalls vorkonfiguriert wurden.

2. Der Schlüssel-Server, beziehungsweise der kontaktierte Backup-Server, sendet nach erfolgreicher Aushandlung mittels GDOI zwei symmetrische Gruppenschlüssel zum beitretenden VPN-Gateway. Einer der Schlüssel wird verwendet, um eine Sicherheitsbeziehung einzurichten; der zweite wird zur Entschlüsselung von Rekeying-Paketen des Schlüssel-Servers benötigt.

3. Da alle VPN-Gateways den gleichen symmetrischen Schlüssel erhalten und somit die gleiche Sicherheitsbeziehung einrichten, können sie anschließend verschlüsselte und integritätsgeschützte Daten transparent miteinander austauschen.

4. Sowohl für den Schutz von Nutzdaten als auch zum Schutz der Rekeying-Pakete verteilt der Schlüssel-Server in vordefinierten Intervallen neues Schlüsselmaterial. Aus Gründen der Skalierbarkeit wird dafür in der Regel die Multicast-Unterstützung des Transportnetzwerks genutzt. Ist kein Multicast verfügbar, werden die VPN-Teilnehmer in regelmäßigen Abständen einzeln kontaktiert.

Eine Besonderheit von GET ist die von Cisco beworbene *tunnellose Verschlüsselung*, welche einen weiteren IPsec-Modus darstellt (siehe Abbildung 3.2). Trotz des Namens wird dabei ein Verfahren eingesetzt, welches dem Tunnelmodus ähnelt und auch den gleichen Übertragungsaufwand besitzt. Allerdings wird im Unterschied zum Tunnelmodus der äußere IP-Header nicht auf Basis einer IPsec-Sicherheitspolitik gebildet, sondern durch Kopie eines Großteils des privaten inneren IP-Header, wie dem Differential Service Code Point (DSCP), der Quell- und der Zieladresse, in den unverschlüsselten äußeren Teil. Der innere Header dient somit lediglich noch der authentisierten Übertragung.

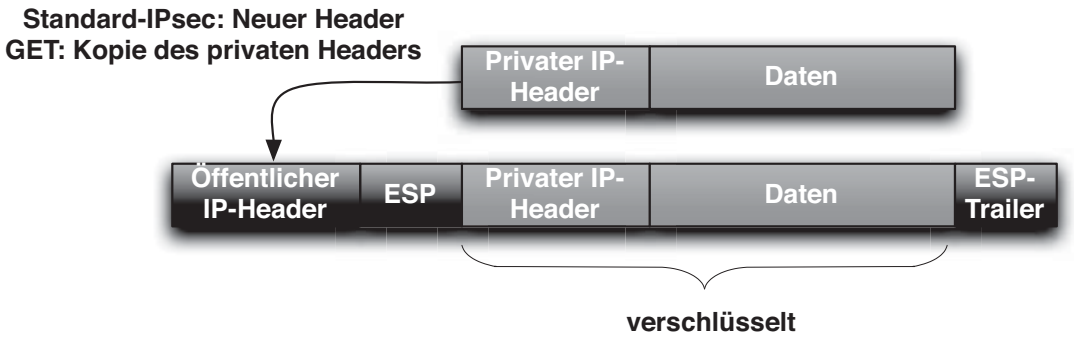


Abbildung 3.2: *Tunnellose Verschlüsselung* in GET durch Kopie des inneren IP-Header

Der Ansatz führt zu einem hochgradig skalierbaren System, welches sich transparent in heutige ISP-Netze integriert. Durch die Nutzung einer einzigen Sicherheitsbeziehung und der Kopie des inneren Header können Multicast und Quality of Service (QoS) realisiert werden, sofern das Transportnetz diese unterstützt. Allerdings werden dadurch ebenfalls NAT-Traversal und eine Unterstützung verschachtelter Strukturen verhindert. Die Nutzung privater IP-Adressbereiche ist nur möglich, sofern diese vom ISP korrekt weitergeleitet werden.

Besonders nachteilig sind beim Einsatz von GET jedoch die vergleichsweisen schlechten Sicherheitseigenschaften [RS09]. Auch wenn GET Datenvertraulichkeit gegenüber

Externen im gewissen Maße erreicht, können interne Angreifer aufgrund der Gruppenschlüssel beliebige Daten entschlüsseln, selbst wenn sie weder Empfänger noch Sender sind. Auch wenn GDOI eine PFS-Option spezifiziert, wird diese laut Cisco's Dokumentation nicht implementiert. Ferner ist die Topologie des VPN leicht zu ermitteln, da jedes Gateway einzig mit dem Schlüssel-Server über IKE kommuniziert. Aufgrund der tunnellosen Verschlüsselung können darüber hinaus Datenströme bis auf den jeweiligen Klienten aufgelöst werden. Eine weitere Nebenwirkung ist für Client-Geräte die Möglichkeit, an beliebige externe Hosts unverschlüsselte Daten zu schicken, indem sie diese in den IP-Header einbetten. Dieser verdeckte Kanal ist zwar im Gegensatz zu einer direkten Kommunikation schmalbandig, im Vergleich zu einer IPsec-Tunnelmodus-Beziehung jedoch extrem breit. Bei letzterer können auch nicht beliebige Systeme adressiert werden; der Angreifer müsste hier zusätzlich noch die Kommunikation im schwarzen Netz mitlesen können.

Um Datenintegritäts- und Authentizitätsziele zu erreichen, muss zum einen sichergestellt werden, dass kein VPN-Gateway oder Schlüssel-Server kompromittiert wird. Zum anderen müssen Administratoren explizit den zeitbasierten Replay-Schutz aktivieren. Aber auch dieser bietet im Gegensatz zum herkömmlichen Mechanismus, der auf Sliding Windows basiert, nur einen begrenzten Schutz, denn innerhalb des konfigurierten Zeitfensters können Pakete beliebig oft wieder eingespielt werden. Signifikanz erhält das Problem beispielsweise, da im Kontext von Wireless Local Area Networks (WLANs) bereits gezeigt wurde, dass das Wiedereinspielen von Paketen zu effizienteren kryptographischen Angriffen führen kann [TWP08].

In Bezug auf den kontrollierter Zugang findet nur eine statische Erstprüfung statt, da das dynamische Rekeying ohne weitere Prüfung mithilfe eines symmetrischen Verfahrens durchgeführt wird. Ein einmal beigetretenes VPN-Gateway kann somit nicht durch protokolltechnische Maßnahmen aus dem Verbund entfernt werden, wodurch eine Zusicherung von Forward Access Control nicht möglich ist. Außerdem werden Schlüssel beim Beitritt eines neuen VPN-Gateways nicht gewechselt, sodass Backward Access Control ebenfalls nicht realisiert wird.

Die Verfügbarkeit eines mit GET konfigurierten VPN wird primär durch die zentralen Schlüssel-Server eingeschränkt, von denen ein Angreifer weiß, dass es maximal acht gibt. Wird ein DoS-Angriff kurz vor dem Ende eines Rekey-Intervalls ausgeführt, ist kein weiterer Betrieb möglich.

Des Weiteren realisiert GET die Graceful Degradation Eigenschaft nicht. Durch Kompromittieren eines beliebigen VPN-Gateways oder Schlüssel-Servers können netzwerkweit weder Vertraulichkeit, noch Datenauthentizität, Instanzenauthentisierung oder kontrollierter Zugang gewährleistet werden. Die Sicherheit des VPN wird durch das symmetrische Rekeying über die Zeit auch nicht wieder hergestellt.

In Bezug auf DoS-Recovery ist es beispielsweise nicht oder nur mit sehr großem Aufwand möglich, einen Schlüssel-Server auf einen anderen IP-Adressbereich zu migrieren, da eine Rekonfiguration aller VPN-Gateways erfolgen muss.

Hamachi^2

Die Firma LogMeIn vermarktet unter dem Namen Hamachi^2 [Log09] einen proprietären VPN-Client, welcher sowohl vollvermaschte als auch Hub-and-Spoke-Szenarien aufbauen kann. Die Konfiguration von VPN-Gateways ist allerdings nur im Hub-and-Spoke-Modus mit einem einzigen Gateway als zentralem Koordinator möglich. Hamachi^2 ist extrem einfach über eine Webseite der Firma LogMeIn zu konfigurieren, zu welcher sich auch die VPN-Clients über ein proprietäres Protokoll verbinden. Die Absicherung zwischen den Clients erfolgt ebenfalls über ein proprietäres Protokoll, welches aber nach den gleichen Prinzipien wie IKE entworfen sein soll. Der zentrale Server tritt dabei auch als NAT-Mediator auf, um direkte Sicherheitsbeziehungen zwischen von NAT betroffenen Geräten herzustellen. Damit in Beziehung steht eine interessante Eigenheit von Hamachi^2: die Nutzung des 5.0.0.0/8 IP-Adressraumes, welcher zurzeit von der Internet Assigned Numbers Authority (IANA) nicht vergeben ist. Jeder Hamachi^2-Client erhält eine virtuelle Adresse aus diesem Raum, um Kollisionen mit privaten IP-Adressenbereichen zu vermeiden, wenn der Client sich hinter einem NAT-Gateways befindet.

Die Firma LogMeIn versucht die Sicherheitsmerkmale von Hamachi^2 herauszustellen, indem sie beispielsweise ihr Protokoll von IKE ableiten und Lastverteilung über Akamai betreiben. Nichtsdestotrotz: Vertraulichkeit, kontrollierter Zugang, Authentisierung und Verfügbarkeit der von Hamachi^2 konfigurierten VPN stehen unter der Kontrolle von LogMeIn. Mechanismen für eine dynamische Zugriffskontrolle werden nicht in der Sicherheitsdokumentation behandelt, sodass davon ausgegangen werden muss, dass keine solchen Mechanismen implementiert sind.

Fazit

Auch wenn das Prinzip der Zentralisierung eine Realisierung vergleichsweise einfacher und effizienter Konfigurationsansätze erlaubt, so besitzen diese alle einen potentiellen Single Point of Failure beziehungsweise Flaschenhals. Skalierbarkeits–, Robustheits- und möglicherweise Verfügbarkeitsprobleme sind die Folge, wenn nicht von Seiten der Infrastruktur meist kostenintensive Vorkehrungen zum Schutz der zentralen Instanz getroffen werden. Die IP-Adressen zentraler Koordinatoren können darüber hinaus relativ einfach durch Verkehrsflussanalysen ausfindig gemacht werden. Dies führt dazu, dass die Realisierung von Infrastructure Hiding nicht möglich ist. In Bezug auf Verfügbarkeit ist eine Migration auf einen anderen IP-Adressbereich während eines DoS-Angriffes in der Regel nicht möglich, da IP-Adressen oder Domain Name System (DNS) Namen fest in allen VPN-Knoten konfiguriert sind.

3.2.2 Dezentrale VPN–Konfigurationssysteme

Während die bereits vorgestellten zentralen Autokonfigurationssysteme vergleichsweise einfach zu verwalten sind und eine Vielzahl funktionaler Anforderungen erfüllen, limitiert der Koordinator Robustheits-, Verfügbarkeits- und Skalierbarkeitseigenschaften.

Einen möglichen Ausweg stellen dezentralisierte Ansätze dar, bei denen die Konfiguration — zumindest im Prinzip — von mehreren Systemen durchgeführt werden kann. Im Folgenden werden fünf solcher dezentraler Verfahren vorgestellt.

Konfiguration über DNS und DNSSEC

Mehrfach wurde die Verwendung des globalen DNS-Verzeichnisdienstes für die Verteilung von VPN Schlüsselmaterial vorgeschlagen [Ric05, Jos06], wobei einige Vorschläge auch die sicherere Variante DNSSEC [LG02, MGMOK06] diskutieren. Die Funktionsweise ist bei allen erwähnten Ansätzen gleich: Sobald eine Sicherheitsbeziehung aufgebaut werden soll, wird eine DNS-Anfrage gestellt, und mithilfe der zurückgegebenen Daten wird anschließend eine Authentisierung durchgeführt. In [RR05] wird auch eine optimistische IPsec-Variante spezifiziert, in der nur dann eine IKE-Aushandlung erfolgt, falls ein so genannter *IPSECKEY*-Eintrag für den betreffenden Kommunikationspartner im DNS vorhanden ist.

Der größte Vorteil des Ansatzes ist die einfache Konfiguration der VPN-Geräte. Jedoch entstehen bereits aus funktionaler Sicht einige schwerwiegende Defizite. So ist es weder möglich, VPN-Gateways oder verschachtelte Szenarien zu konfigurieren, noch private IP-Adressbereiche innerhalb des VPN zu verwenden. Der Einsatz in Umgebungen mit NAT wird ebenfalls nicht diskutiert.

In Bezug auf die Sicherheit müssen mögliche Probleme der verantwortlichen DNS-Server berücksichtigt werden. Neben Authentisierungs- und Integritätsaspekten schließt dies auch Verfügbarkeits- und Robustheitsfragen mit ein, da die korrekte Beantwortung der DNS-Anfragen kritisch für das VPN wird. Besonders bei Einsatz von Standard-DNS kann sich die Möglichkeit des Fälschens von Antworten [Kam08] als problematisch erweisen. Auch wenn die Entwicklung von DNSSEC das erwähnte Problem beseitigt, verbleiben offene Punkte. Prominente Beispiele sind die Herausforderung eine Organisation zu finden, welche die Root-Zone signiert, sowie die Sicherung des Signaturpfades, welche von der Integrität aller Zwischenattestationen abhängt [BD04].

Wippien

Wippien [Pet10] ist ein weiteres System zur dezentralen Konfiguration von VPN-Topologien. Im Gegensatz zum vorherigen Ansatz erfolgt die Aushandlung von Sicherheitsbeziehungen allerdings nicht über DNS-Server sondern über die Jabber-Infrastruktur, welche primär einen dezentral organisierten Instant-Messaging-Dienst bereitstellt. Um sich mit einem Jabber-Kontakt zu verbinden, führt Wippien einen sehr vereinfachten Schlüsselaustausch über die jeweiligen Jabber-Server aus. Mithilfe des abgeleiteten Schlüssels werden dann die direkt zwischen den Endsystemen aufgebauten Sicherheitsbeziehungen gesichert. Dazu kommt ein proprietäres Protokoll zum Einsatz, das eine Verschlüsselung auf Basis des Advanced Encryption Standard (AES) durchführt.

Trotz der Ähnlichkeiten zum vorgestellten DNS-Ansatz erlaubt Wippien die Nutzung privater IP-Adressbereiche und durch die Verwendung des Jabber-Dienstes auch den Aufbau von Sicherheitsbeziehungen über NAT-Gateways hinweg. Wippien stellt allerdings keine Unterstützung von VPN-Gateways zur Verfügung, und verschachtelte Topologien können ebenfalls nicht konfiguriert werden. Erweiterte Netzwerkdienste wie Multicast sind ebenfalls nicht angedacht. Wippien stellt des Weiteren zwar ein dezentrales System dar; die Skalierbarkeit wird allerdings durch die vollständig vermaschte Struktur zwischen Jabber-Kontakten limitiert. Die Robustheit des Ansatzes hängt außerdem von den Jabber-Servern ab, deren Ausfall für ihre jeweiligen Klienten kritisch ist.

Unter dem Gesichtspunkt der Sicherheit verbleiben schwerwiegende Bedenken. Innerhalb weniger Minuten wurden im Rahmen dieser Recherchen mehrere schwerwiegende Sicherheitsprobleme entdeckt [Ros10], die auch nach mehreren Monaten durch den Entwickler nicht behoben wurden. Dazu zählt neben verschiedenen Methoden, Man-in-the-Middle-Angriffe durchzuführen, auch die Möglichkeit Schlüsselmaterial vorherzusagen. Algorithmen zum Schutz der Datenintegrität und –authentizität sind, sofern vorhanden, weder dokumentiert noch Teil der Open-Source-Implementierung von Wippien.

Social VPN

Um einfach Overlay-Netze zwischen Privatpersonen zu etablieren, setzt Social VPN [FBJW08] eine Facebook-Applikation ein, die automatisch Zertifikate zwischen „befreundeten“ Profilen des sozialen Netzwerks austauscht. Diese werden später in einem an IPsec angelehnten Schlüsselaustausch zur Ableitung kryptographischer Schlüssel genutzt. Durch den Einsatz eines strukturierten *Brunet*-Overlays [GABF06] zum Finden von VPN-Teilnehmern und zur anschließenden Weiterleitung von IP-Paketen, funktioniert Social VPN im Anschluss an den zentralen Schlüsselaustausch vollständig verteilt.

Interessant ist der Mechanismus, mit dem in Social VPN IP-Adresskollisionen vermieden werden: Jede aufgebaute Sicherheitsbeziehung erhält eine virtuelle IP-Adresse, welche aber nur lokal eindeutig ist. Diese wird dann mittels NAT auf die reale IP-Adresse des Kommunikationspartners abgebildet. Um dem Nutzer trotzdem leicht zu merkende Adressen zur Verfügung zu stellen, wird zusätzlich eine lokale DNS-Schnittstelle implementiert, welche Facebook-Namen auf die korrespondierenden lokalen IP-Adressen abbildet.

In Bezug auf funktionale Anforderungen kann Social VPN keine VPN-Gateways konfigurieren und ist nicht in der Lage, mit verschachtelten VPN-Topologien umzugehen. Einfache Weiterleitungen zur Umgehung von NAT-Gateways bilden dabei die einzige Ausnahme. Abgesehen davon werden sowohl funktionale als auch nichtfunktionale Ziele weitgehend erreicht. Social VPN ist einfach zu bedienen, vergibt private IP-Adressen, funktioniert auch beim Einsatz von NAT-Gateways, skaliert über die Anzahl der Teilnehmer und kann durch Nutzung des Brunet-Overlays relativ robust auf Fehler reagieren, wobei Detailprobleme wie Netzwerkpartitionierungen auch hier

nicht adressiert werden. Social VPN bietet des Weiteren einen einfachen Mechanismus zur Behandlung von Multicast-Paketen, welche simpel an die vollständige direkte Nachbarschaft gesendet werden.

Allerdings treten bei Betrachtung der Sicherheit von Social VPN größere Probleme auf. Ein Austausch von Identitäten und Zertifikaten über Facebook oder ähnliche soziale Netze impliziert durch das Risiko des Identitätsdiebstahls [BSBK09] erhebliche Schwachstellen. Es wird auch nicht gezeigt, wie mit Kontakten umgegangen wird, die ihre Namen ändern und so eine DNS-Kollision erzeugen. Ferner wird auch die Forderung nach Graceful Degradation erschwert, da Profile aus den Netzen nicht ohne Weiteres und in angemessener Zeit entfernt werden können. Auch die Autoren von Social VPN scheinen sich dieser Probleme bewusst geworden zu sein, da aktuelle Versionen des Systems — wie Wippien — auf der Jabber-Infrastruktur und manuell ausgetauschten kryptographischen Fingerabdrücken aufbauen. Leider wird die Sicherheit des zugrundeliegenden Peer-to-Peer-Netzwerks durch die Autoren nicht weiter adressiert, sodass auch wenn Ende-zu-Ende-Sicherheit gewährleistet wird, Angreifer das ungesicherte Brunet attackieren können. So können durch gezielte Routing- und DoS-Angriffe die Verfügbarkeit beliebig gewählter Teilnehmer negativ beeinflusst werden.

N2N

Network to network (N2N) [DA08] ist ein weiteres proprietäres System, um innerhalb eines VPN autonom Sicherheitsbeziehungen zu erstellen. Auch wenn die Autoren ihren Ansatz als Peer-to-Peer-Netzwerk bezeichnen, unterscheidet er sich fundamental von vollständig verteilten Systemen wie P2PVPN oder TED, da jedes VPN über ein oder mehrere dedizierte *Supernodes* verfügen muss.

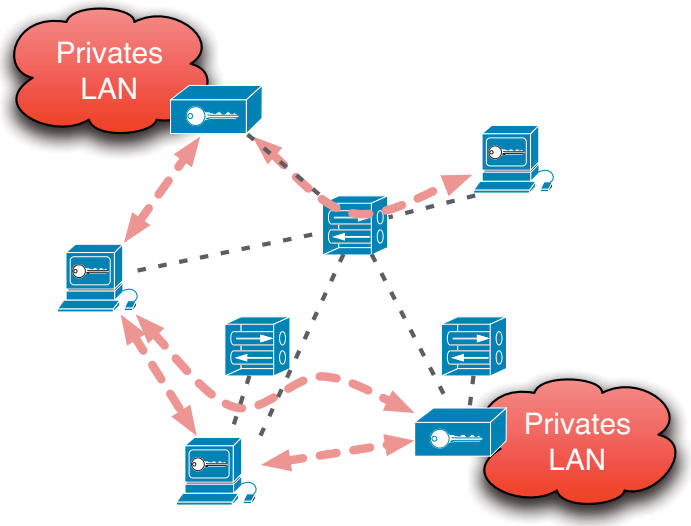


Abbildung 3.3: Beispiel einer N2N Topologie mit drei Supernodes und fünf Edge Nodes (zwei davon mit privaten Teilnetzen)

Wie in der Beispieltopologie in Abbildung 3.3 dargestellt, stellen die Supernodes im Prinzip Server dar, die Daten zwischen den VPN-Teilnehmern weiterleiten. Diese werden in N2N als *Edge Nodes* bezeichnet. Um die Weiterleitung effizienter zu gestalten, können die Edge Nodes auch untereinander direkte Sicherheitsbeziehungen aufbauen, sodass Supernodes im Idealfall anschließend Daten nur weiterleiten, falls keine direkte Kommunikation im Transportnetz möglich ist (beispielsweise aufgrund von NAT). Ferner geschieht die Verteilung von Broadcast-Paketen immer über Supernodes. Das Routing gestaltet sich daher relativ einfach: Pakete werden immer direkt über eine Sicherheitsbeziehung zu einem verantwortlichen Edge Node ausgeliefert. Falls keine solche Sicherheitsbeziehung existiert, erfolgt eine Weiterleitung zum Supernode, welcher dann das Paket zum Ziel leitet. Um aus Robustheits- oder Skalierbarkeitsgründen mehrere Supernodes zu unterstützen, bedienen sich die N2N-Entwickler eines Tricks: Auf den Edge Nodes werden dazu mehrere VPN-Prozesse gestartet, welche sich dann jeweils zu einem anderen Supernode verbinden. Da N2N auf der roten Seite auf Datensicherungsschicht arbeitet, können die Prozesse anschließend über virtuelle Ethernet-Bridges gekoppelt werden. Diese Methode erlaubt auch die Kopplung mehrerer N2N-VPN.

Gemessen an den funktionalen und nichtfunktionalen Kriterien, ist N2N mit — Ausnahme der Kopplung — einfach zu konfigurieren, ermöglicht ganze Teilnetze einzubinden, erlaubt die Nutzung privater IP-Adressräume sowie die Integration von über NAT-Gateways verbundenen Edge Nodes. Dabei wird auf keine besonderen Mechanismen des Transportnetzes wie Multicast zurückgegriffen. Aufgrund der vollvermaschten Topologie zwischen den Edge Nodes eines Supernode und der Nutzung von Broadcast-Nachrichten skaliert N2N relativ schlecht. Werden mehrere Supernodes eingesetzt, kann die Skalierbarkeit erhöht werden, allerdings wird sie weiterhin durch die Nutzung des Spanning Tree Protocol (STP) als Routing-Protokoll beschränkt. Dies hat ebenfalls negativen Einfluss auf die Effizienz, da nicht alle optimalen Wege gefunden und verwendet werden.

Bereits die Webseite von N2N [DA10] beschreibt mehrere Sicherheitsprobleme der aktuellen Version: Die Pakete werden zwar mit Twofish im Cipher-Block-Chaining (CBC)-Modus verschlüsselt, bieten aber keinen Message Authentication Code (MAC) zur Sicherung von Datenintegrität und –authentizität. Zudem enthält N2N keinen Mechanismus zum Schutz vor wieder eingespielten Paketen, und die verwendeten Schlüssel können nicht in regelmäßigen Abständen gewechselt werden. Durch die Nutzung symmetrischer Gruppenpasswörter zur Verschlüsselung ist es weder möglich, einzelne Instanzen zu authentisieren, noch eine dynamische Zugriffskontrolle zu realisieren.

Neben diesen dokumentierten Schwachstellen existieren weitere:

- Der Verschlüsselungsalgorithmus verwendet keinen Initialisierungsvektor (IV) und erleichtert so Verkehrsfluss- und Kryptoanalysen.

- Da die ausgetauschten Daten nur durch einen symmetrischen Gruppenschlüssel geschützt werden, kann Graceful Degradation nicht umgesetzt werden.

- Die Sicherheit der Overlay-Kommunikation wird von N2N nicht adressiert, sodass bereits das Fälschen einer Ethernet-Adresse in einem der Teilnetze verheerende Folgen für die Sicherheit des VPN hat.
- Da Supernodes keine Zugangskontrolle für die Edge Nodes durchführen, können Daten von beliebigen Systemen über sie weitergeleitet werden.
- Aus Skalierbarkeitsgründen kann ein VPN nur wenige Supernodes verwenden. Darüber hinaus erfolgt die Organisation der Weiterleitung zwischen Supernodes über STP, sodass nur Baumstrukturen ohne Redundanz entstehen. Daher wird die Resistenz von N2N-VPN gegenüber DoS-Angriffen geschwächt.

Viele der Sicherheitsprobleme in N2N resultieren somit aus der Nutzung eines proprietären kryptographischen Protokolls, welches keinem Standardisierungsprozess wie IPsec oder TLS unterlag.

Cisco Dynamic Multipoint VPN (DMVPN)

DMVPN [BKW05, Cis06b, Flu07b] bezeichnet wie GET eine Sammlung von IPsec-basierten Protokollen und Praktiken, die von Cisco unter einem Namen vermarktet werden. Die von DMVPN erstellten Topologien erscheinen zunächst dem eben vorgestellten N2N-Ansatz ähnlich. Auch hier existieren eine Reihe statischer Knoten, welche in diesem Fall als *Hubs* bezeichnet werden. Diese sind allerdings im Gegensatz zu N2N untereinander mit IPsec-Tunneln verbunden und verwenden zur Weiterleitung von Paketen untereinander ein Interior Routing-Protokoll, wie Open Shortest Path First (OSPF) oder Enhanced Interior Gateway Routing Protocol (EIGRP). Dynamische *Spoke*-Knoten verbinden sich zu den Hubs, welche die Weiterleitung zu anderen Hubs oder Spokes übernehmen. Für die Optimierung dieser Topologien verwendet DMVPN nach Bedarf dynamische IPsec-Beziehungen zwischen Spokes, welche den Weg über den Hub abkürzen.

Die in DMVPN eingesetzten Autokonfigurationsmechanismen reduzieren den Aufwand bei der Einrichtung dynamischer Beziehungen. Dennoch wird gerade bei der Konfiguration mehrerer Hubs für die Gewährleistung von Fehlertoleranz und Skalierbarkeit noch sehr viel manuelle Interaktion benötigt. Sowohl Hubs als auch Spokes können als VPN-Gateways fungieren, deren Teilnetze auch über private IP-Adressbereiche verfügen können. Die Konfiguration verschachtelter Hub-Strukturen ist aus Skalierbarkeitsgründen vorgesehen, wobei diese hierarchisch organisiert sein müssen. Die verschachtelte Anordnung von Spokes ist nicht vorgesehen. DMVPN benutzt keine erweiterten Netzwerkdienste und kann auch Beziehungen zwischen Spokes, welche sich hinter einem NAT-Gateway befinden, sowie öffentlich erreichbare Hubs und Spokes konfigurieren. Eine Einrichtung dynamischer Beziehungen zwischen Spokes, welche beide jeweils über NAT angebunden sind, ist nicht möglich. Die Weiterleitung von Multicast-Verkehr ist ebenfalls möglich, allerdings erfolgt diese immer über die Hub-Topologie, in der durch die statische Struktur Protocol Independent Multicast (PIM) eingesetzt werden kann. Insgesamt werden somit alle funktionalen Kriterien wenigstens zum Teil durch DMVPN adressiert.

Das Erfüllen nichtfunktionaler Ziele bedarf einer differenzierteren Diskussion. So ist es in DMVPN beispielsweise möglich, einem Spoke mehrere Hubs zuzuordnen und auf diese Weise eine höhere Robustheit zu erreichen. Dazu müssen jedoch eine gleiche Anzahl virtueller Netzwerkschnittstellen inklusive eindeutiger IP-Adressen konfiguriert werden. Wenn also ein Ausfall von maximal drei Hubs toleriert werden soll, wächst durch Ciscos Dateiformat die Größe der Konfiguration in jedem Spoke auf 400 % an, da für jeden Hub umfangreiche Detaileinstellungen nötig sind. Die Skalierbarkeit der entstehenden VPN hängt von einer Reihe von Faktoren ab: der Anzahl der eingesetzten Hubs, der Menge des direkt über die Hubs weitergeleiteten Verkehrs, der Mobilität der Spokes und der Anzahl der Sicherheitsbeziehungen, die zwischen Spokes aufgebaut werden müssen. Eine große Anzahl von Hub Gateways kann, je nach konfigurierter Topologie, die Verkehrsbelastung im Schnitt senken. Gleichzeitig wird aber die Zeit verlängert, bis mobile Spokes korrekt im VPN annonciert sind. In Szenarien, in denen Spokes vor allem miteinander Nutzdaten austauschen, erzeugt DMVPN automatisch eine vollvermaschte Topologie, sodass sich die Zeit für den Aufbau von Sicherheitsbeziehungen stark verzögern kann. Gleichwohl und besonders im Vergleich zu anderen Ansätzen skaliert DMVPN in den meisten Szenarien gut, ist relativ effizient, und die erforderliche Zeit für Rekonfigurationsvorgänge hängt primär von der Konvergenz des Routing-Protokolls ab.

In Bezug auf die Datensicherheit werden alle von DMVPN übertragenen Daten durch IPsec-Beziehungen gesichert, sodass Vertraulichkeit, inklusive PFS und einem Schutz gegen verdeckte Kanäle, sowie Datenintegrität und –authentizität gewährleistet werden. Zwei Teilziele in Bezug auf Vertraulichkeit werden allerdings nicht erreicht: Zum einen ist keine Ende-zu-Ende-Sicherung möglich, da Hubs Klartextdaten erhalten, und zum anderen ist Infrastructure Hiding problematisch, da die Hubs durch die Kommunikationsstruktur einfach zu entdecken sind. Einem Angreifer ist es somit möglich, weitere Attacken gezielter vorzubereiten und durch Beobachtung von Spoke-to-Spoke-Verbindungen Verkehrsflussanalysen durchzuführen. Die Zugriffskontrolle in DMVPN basiert auf zwei Verfahren. Dabei muss zunächst eine gültige IPsec-Beziehung hergestellt werden und anschließend eine Authentisierung mithilfe des Next Hop Resolution Protocol (NHRP) und auf Basis eines Gruppenpasswortes erfolgen. Leider ist es nicht möglich, dieses Gruppenpasswort einfach innerhalb des VPN zu wechseln, sodass weder ein periodischer Wechsel noch Forward Access Control gewährleistet werden können. Im Vergleich zu manuell konfigurierten VPN reduziert DMVPN den Konfigurationsumfang; allerdings sind die Einstellungen erheblich komplexer und immer noch so weitreichend, dass das Risiko von Konfigurationsfehlern zumindest bei ausfallsicheren Topologien nicht unerheblich sein dürfte.

Ein größeres Problem stellt die Konfiguration der inneren IP-Adressen dar. Diese werden ohne weitere Verifikation aus der Spoke-Konfiguration übernommen, sodass bereits ein einzelnes kompromittiertes Spoke-System sowohl die Zugriffskontrolle und als auch die Verfügbarkeit des gesamten VPN in Mitleidenschaft ziehen kann. Durch das Umgehen der Zugriffskontrolle werden dann unter Umständen auch Man-in-the-Middle-Angriffe möglich und somit weitere Sicherheitsziele verletzt. Neben diesen Auswirkungen auf die Graceful Degradation hängt die Verfügbarkeit auch von der DoS-Resistenz des VPN ab. DMVPN bietet in dieser Hinsicht allerdings keine weitere

Unterstützung; der Administrator muss seine Hub-Strukturen so redundant planen, dass ein gewünschter Grad an DoS-Resistenz erreicht wird. Ist ein DoS-Angriff gegen Hubs erst einmal erfolgreich, wird eine Rekonfiguration schwierig, da die Spoke-Systeme lediglich ihre statisch eingetragenen Hubs kontaktieren.

Fazit

Im Vergleich zu den zentralen Autokonfigurationsansätzen bieten die gerade vorgestellten, dezentralen Systeme im Prinzip bessere Skalierbarkeits- und Robustheitseigenschaften. Gezielte Sabotageangriffe sind aber beispielsweise wie bei zentralen Ansätzen problematisch, da die Zuordnung von Server beziehungsweise Hubs zu VPN-Knoten weiterhin statisch erfolgt. Der Funktionsumfang der dezentralen Systeme variiert stark; ein hoher Funktionsumfang geht aber in jedem Fall mit einem hohen Konfigurationsaufwand einher.

3.2.3 Vollständig verteilte VPN–Konfigurationssysteme

Während dezentrale Autokonfigurationskonzepte noch exponierte Komponenten besitzen, sind diese in den verteilt realisierten Systemen nicht mehr vorhanden. Im folgenden Abschnitt werden einige Konzepte vorgestellt, bei denen in allen VPN-Knoten der gleiche Algorithmus abläuft.

Opportunistic Encryption

Die einfachste Form eine verteilte Sicherung umzusetzen, stellt die Umsetzung mittels einer „Best Effort“ Strategie dar. Das bedeutet, es wird zunächst versucht eine SSL-Verbindung herzustellen oder eine IPsec-Beziehung aufzubauen, bevor das erste Datenpaket übermittelt wird. Schlägt der Aufbau fehl, zum Beispiel weil der Gegenüber keine Verschlüsselung unterstützt, wird das Datenpaket ungeschützt ausgeliefert. Better-Than-Nothing Security (BTNS) ist ein vor kurzem standardisierter Ansatz [WR08], welcher die Aushandlung unauthentisierter Beziehungen in IPsec erlaubt. Allerdings wird hierbei auf die Alternative einer völlig ungesicherten Übertragung verzichtet.

Wird dieser Ansatz in Endsystemen eingesetzt, kann auf einfache Weise ein skalierbares, effizientes und beim Einsatz von Cookies und Client Puzzles auch DoS-resistentes VPN konstruiert werden, da keine zentralen Dienste existieren und Graceful Degradation gewährleistet wird. Dessen ungeachtet werden funktionale Anforderungen relativ schlecht erfüllt: Weder die Einrichtung privater IP-Adressbereiche, noch verschachtelte Tunnel oder Beziehungen zwischen von NAT abgeschirmten Geräten sind möglich. Die Gewährleistung von Sicherheit ist aufgrund der fehlenden Authentifizierung lediglich unter Ausschluss aktiver Angreifer möglich.

Cryptographically Generated Addresses (CGA)

Wie bei der BTNS-Strategie wird auch durch den Einsatz von CGA [LMK07, Aur05, Aur03] kein vollständig vertrauenswürdiger Sicherheitsmechanismus umgesetzt. Allerdings wird im Gegensatz dazu ein gewisser Schutz gegen aktive Angreifer erreicht, indem sich jedes Endgerät zwar ein eigenes Zertifikat ausstellt, dieses aber mit der IP-Adresse korrelieren muss. Dafür sucht sich das Gerät eine freie IP-Adresse, deren letzten Stellen mit der kryptographischen Prüfsumme des Zertifikates übereinstimmen, beispielsweise beim Einsatz von IPv6 die letzten 60 Bit. Aktive Angreifer werden durch Client Puzzles daran gehindert, sehr viele Zertifikate zu generieren und so auf lange Sicht, Zertifikate für eine IP-Adresse zu erzeugen, welche der eines potentiellen Opfers entspricht. Aber auch mit den in [BOH09] vorgeschlagenen Optimierungen wird die gebotene Sicherheit gegen Man-in-the-Middle-Angriffe durch die Anzahl der Bits limitiert, welche für die Repräsentation der Prüfsumme verwendet werden.

Aus funktionaler Sicht ergeben sich nur wenige erfüllte Anforderungen, da CGA nur in IPv6-Netzen eine einfache Konfiguration gewährleistet. In IPv4-Netzen sind die Adressen zu kurz, um einen sinnvollen Einsatz zu rechtfertigen. Allerdings werden mithilfe der einfachen Struktur die nichtfunktionalen Ziele Skalierbarkeit, Effizienz und Robustheit hervorragend erfüllt. Schnelle Adressänderungen, wie sie für Agilität und DoS-Recovery benötigt werden, sind nicht auf diese Art zu bewerkstelligen, da neue Zertifikate erstellt und die neuen IP-Adressen bekannt gemacht werden müssen. Trotz der leichten Verbesserung der Sicherheit gegenüber BTNS ist das erreichte Niveau weit unter dem manuell konfigurierter VPN.

Cisco Tunnel Endpoint Discovery (TED)

Das TED-Protokoll [Bha08, BKW05, Flu07a] ist Teil eines weiteren IPsec-Autokonfigurationssystems von Cisco, und trotz zaghafter Standardisierungsbemühungen [Flu00] ist die einzig bekannte Implementierung Teil der proprietären Router-Firmware des Unternehmens.

Die Einrichtung von IPsec-Beziehungen verläuft bei einem Einsatz von TED in den folgenden vier Schritten (siehe auch Abbildung 3.4):

1. Datenpakete ohne gültige Sicherheitsbeziehung werden zunächst vom VPN-Gateway verworfen.
2. Im Unterschied zu einem manuell konfigurierten IPsec-VPN wird ein IKE-Paket erzeugt und in Richtung der Zieladresse des verworfenen Datenpaketes versendet.
3. Das IKE-Paket wird vom korrespondierenden IPsec-Gateway abgefangen, und es wird ein vollständiger Schlüsselaustausch durchgeführt.
4. Nach dem erfolgreichen Aufbau der IPsec-Beziehung werden die durch den Klienten ausgelösten, erneuten Übertragungen des Paketes, wie im manuell konfigurierten, VPN von der IPsec-Behandlung der Gateways erfasst.

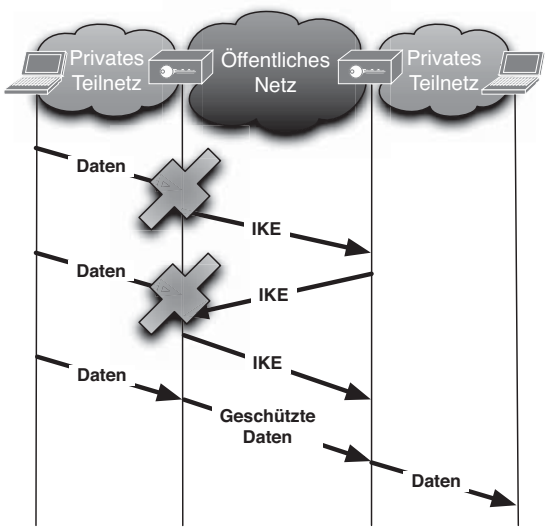


Abbildung 3.4: Beispiel des Aufbaus einer IPsec-Beziehung in TED

Der größte funktionale Nachteil dieses einfachen Verfahrens ist der Bedarf an öffentlichen IP-Adressen innerhalb privater Teilnetze, da das Routing von IKE-Nachrichten durch die Router des Transportnetzes erfolgen muss. Verschachtelte Sicherheitsbeziehungen, Multicast und von NAT betroffene Systeme können mit TED nicht konfiguriert werden. Weiterhin ist die Konfiguration noch relativ komplex, und TED kann nicht auf Routing-Probleme im Transportnetz reagieren.

Im Kontrast dazu sind die Sicherheitsziele relativ gut erfüllt. Gegenüber manuell konfigurierten VPN hat TED nur den potentiellen Nachteil, dass Klienten mit ihren Datenpaketen IKE-Pakete an beliebige Ziele generieren können. Administratoren können zwar den IP-Adressbereich einschränken; sie verlieren aber die Flexibilität des Verfahrens, wenn sie alle potentiellen IP-Adressen vorher eintragen. Daher ist davon auszugehen, dass Klienten durch das Generieren von IKE-Paketen in bestimmten Intervallen zumindest einen schmalbandigen Kanal an beliebige Hosts außerhalb des VPN erzeugen können. Ferner ist die Wiederherstellung des VPN unter DoS-Angriffen problematisch, da alle Klienten eines betroffenen VPN-Gateways neue öffentliche IP-Adressen erhalten müssen.

Security Policy Protocol (SPP)

Das SPP [SS99, BLLM01, SC02] ist ein nicht mehr weiter verfolgter Standardisierungsversuch der IETF, welcher große Ähnlichkeiten zu TED aufweist. Klienten nutzen hier einen *traceroute*-ähnlichen Mechanismus, um Geräte zu erkennen, die IPsec-Sicherheitspolitiken durch Zwangsverschlüsselung umsetzen. SPP bietet wesentlich mehr Funktionalität als TED in Bezug auf die Aushandlung von Sicherheitspolitiken bei unterschiedlichen administrativen Domänen. Die grundlegenden funktionalen, nichtfunktionalen und Sicherheitseigenschaften sind jedoch zu TED identisch.

Proactive Multicast IPSEC Discovery Protocol (PMIDP)

Einen gänzlich anderen Ansatz verfolgt das PMIDP [Tra06], bei dem IPsec-Gateways ihre inneren privaten IP-Adressbereiche durch das Versenden von verschlüsselten Multicast-Paketen innerhalb des nicht vertrauenswürdigen Netzes annoncieren. Das Transportnetz gewährleistet, dass diese Pakete an alle anderen relevanten VPN-Gateways ausgeliefert werden, die anschließend je nach Bedarf IPsec-Beziehungen aufbauen können.

Das größte funktionale Defizit ist die notwendige Unterstützung von Multicast im Transportnetz, sodass PMIDP beispielsweise nicht im Internet verwendet werden kann. Außerdem werden weder NAT-Traversal noch die Einrichtung verschachtelter Strukturen unterstützt. Allerdings stellt PMIDP einen der wenigen Ansätze dar, die auch einen Multicast-Dienst innerhalb des VPN bereitstellen. In Bezug auf Robustheit und Effizienz muss abgewogen werden, wie häufig die Netze annonciert werden: häufiges Senden sorgt für eine schnelle Wiederherstellung des VPN nach Netzwerkausfällen. Gleichzeitig wird die Effizienz des Ansatzes verringert, da nicht nur in Bezug auf die Anzahl der VPN-Gateways linear viele Pakete versendet werden müssen, sondern insgesamt auch quadratisch viele Pakete empfangen werden. Gerade wenn das System auch in Umgebungen mit schmalbandigen, unzuverlässigen Transportnetzverbindungen funktionieren soll, wird so die Skalierbarkeit stark eingeschränkt. Die Robustheit des Ansatzes könnte darüber hinaus durch Routing-Mechanismen innerhalb der Overlays erhöht werden.

Einige Fragen verbleiben jedoch in Bezug auf die Sicherheit:

- PMIDP setzt voraus, dass alle VPN-Gateways in der Lage sind, Multicast-Nachrichten an eine vordefinierte Gruppe zu versenden. Es ist demzufolge Aufgabe des Transportnetzes, VPN-Gateways von anderen Hosts zu unterscheiden, und diese daran zu hindern, einen DoS-Angriff durch das Fluten von Multicast-Paketen an die gleiche Zieladresse auszuführen. Mögliche Gegenmaßnahmen, wie eine Ratenkontrolle, werden allerdings von PMIDP nicht adressiert.

- Schutzmechanismen gegen wiedereingespielten Nachrichten sind nicht dokumentiert, sodass davon auszugehen ist, dass potentielle Angreifer Subnetze mit veralteten äußeren IP-Adressen annoncieren können.

- Die inneren, privaten IP-Adressen werden in PMIDP zwar verschlüsselt, um Infrastructure Hiding zu erreichen. Mechanismen zur Erneuerung der benötigten Gruppenschlüssel werden jedoch nicht diskutiert.

- Die annoncierten IP-Adressen werden nicht durch eine Zertifizierungsinstanz attestiert, sodass kompromittierte VPN-Gateways beliebige Adressbereiche für sich beanspruchen und so Verkehr anziehen können. Somit kann keine Graceful Degradation gewährleistet werden.

Die Beschränkung auf Transportnetze mit Multicast-Fähigkeiten und die verbleibenden Sicherheitsprobleme begrenzen die Einsatzgebiete von PMIDP erheblich.

WASTE

Auch wenn WASTE wahrscheinlich eher aufgrund firmenpolitischer Gründe [Hu03] und nicht wegen technischer Neuerungen auf sich aufmerksam machte, so stellt es doch das erste VPN-Overlay dar, welches mit eigenen Routing-Fähigkeiten auf private Nutzer abzielt. Nutzer verbinden sich in WASTE zu Kontakten unter Nutzung asymmetrischer Kryptographie. Das entstehende partiell vermaschte Netzwerk wird anschließend verwendet, um Dateien zu übermitteln oder über privaten Chats zu kommunizieren [EHL05]. Im Vergleich zu anderen VPN-Konfigurationssystemen erlaubt WASTE lediglich spezifische im Protokoll verankerte Services und nicht die Weiterleitung beliebiger Pakete. Die Annoncierung der Dienste erfolgt über das Versenden von Broadcast-Nachrichten, welche über alle Links des VPN weitergeleitet werden. Antworten werden per Unicast versendet, wobei der Pfad über einen Backward-Learning-Algorithmus bestimmt wird.

Die Unterstützung von Gateways oder des Transports beliebiger Daten ist aufgrund der Zielgruppe von WASTE nicht vorgesehen; dafür ist es einfach zu verwenden und benötigt keine besonderen Netzwerkdienste. Zudem wird NAT-Traversal nicht unterstützt. In Bezug auf nichtfunktionale Ziele werden durch den einfachen Routing-Algorithmus die Skalierbarkeit und Effizienz stark eingeschränkt. Die Robustheit von WASTE-Netzen hängt darüber hinaus stark von der Topologie ab, die sich implizit durch die Nutzerbeziehungen ergibt.

Ebenfalls stark eingeschränkt sind die Sicherheitseigenschaften von WASTE. Zwar sollen Datenvertraulichkeit, –authentizität und –integrität durch kryptographische Maßnahmen gegen externe Angreifer gesichert werden, allerdings werden interne Angreifer nicht berücksichtigt. Der verwendete Propagating Cipher-Block-Chaining (PCBC)-Verschlüsselungsmodus erlaubt ferner das Vertauschen von Nachrichtenblöcken ohne kryptographischen Lawineneffekt [Mit03]. Innerhalb des VPN werden keinerlei Sicherheitsmaßnahmen verwendet, sodass Graceful Degradation sowie Ende-zu-Ende Sicherheit nicht gewährleistet werden und jeder Teilnehmer auf die Integrität aller anderen Teilnehmer vertrauen muss. DoS-Recovery ist möglich; dazu müssen sich betroffene Systeme nur mit einer anderen IP-Adresse wieder zum VPN verbinden.

P2PVPN

Einen weiteren interessanten Ansatz aus der Praxis stellt P2PVPN [Gin10] dar. Es nutzt ein proprietäres Protokoll, um partiell vermaschte Overlay-Netzwerke aufzubauen und Nutzdaten darüber weiterzuleiten. Die Pakete werden dazu von der Applikation entgegen genommen, und über TCP-Verbindungen an den nächsten Hop weitergeleitet, der in jedem P2PVPN-Knoten über einen Link-State-Routing-Algorithmus berechnet wird. Während die Paketweiterleitung demnach keine wesentlichen Neuerungen enthält, unterscheidet sich der Bootstrapping-Prozess signifikant von anderen Ansätzen. P2PVPN verbindet sich dazu mit einem BitTorrent-Server und fragt nach anderen Knoten mit dem gleichen Schlüssel. Auf diese Weise können auch VPN konfiguriert

werden, bei denen alle Klienten dynamische IP-Adressen besitzen. Auch wenn von P2PVPN nicht unterstützt, könnten so selbst VPN aufgebaut werden, bei denen alle VPN-Teilnehmer durch NAT vom Internet abgeschirmt sind.

Aus funktionaler Sicht bietet P2PVPN eine ganze Reihe von Fähigkeiten. Es ist einfach zu konfigurieren, ermöglicht zumindest die manuelle Einrichtung verschachtelter Sicherheitsbeziehungen, erlaubt die Verwendung privater IP-Adressbereiche und ist über das Internet zu verwenden. Nachteilig wirken sich nur die fehlende Gateway-Funktionalität und die begrenzten NAT-Traversal-Möglichkeiten aus. Da das System unstrukturierte Overlays erstellt in denen jeder Teilnehmer so viele Sicherheitsbeziehungen wie möglich proaktiv aufbaut, ist P2PVPN aus Skalierbarkeits- und Effizienzgründen lediglich für kleine VPN geeignet.

Die Sicherheitseigenschaften von P2PVPN sind in der untersuchten Version 0.7 relativ problematisch. Auch wenn das proprietäre kryptographische Protokoll durch die Verwendung von AES im CBC-Modus einen prinzipiellen Schutz aufweist, existiert kein Rekeying-Mechanismus. Auch die anderen Vertraulichkeitsziele können nicht erreicht werden: Es existiert keine Ende-zu-Ende-Sicherung, keine PFS, kein Schutz gegen verdeckte Kanäle und kein Mechanismus zur Umsetzung von Infrastructure Hiding. P2PVPN verfügt über keinen echten Schutz von Datenintegrität und –authentizität. Stattdessen wird davon ausgegangen, dass Angreifer keine Pakete mit gültigen UDP- oder TCP-Prüfsummen erzeugen können. Bei der Umsetzung des kontrollierten Zugangs unterscheidet P2PVPN zwischen zwei Nutzergruppen: Zum einen existieren VPN-Administratoren, die einen privaten Schlüssel kennen, welcher zur Erstellung des Netzwerks verwendet wurde. Zum anderen gibt es temporäre Nutzer, welche über ein Zertifikat verfügen, das mit dem privaten Schlüssel signiert wurde und als temporäre Zugriffsberechtigung verwendet werden kann. Da der zum Zertifikat gehörende private Schlüssel allerdings von einem Administrator generiert wird, kann dieser sich später für den Nutzer ausgeben. Das Entfernen von Administratoren aus dem VPN zur Realisierung von dynamischer Zugriffskontrolle ist nicht vorgesehen, und temporär eingeladene Nutzer können nur durch das Ablaufen ihres Zertifikates an einer weiteren Teilnahme gehindert werden. Graceful Degradation ist nicht vorgesehen, sodass die Sicherheit des VPN von der Integrität aller Teilnehmer abhängt.

tinc

Mit *tinc* [Sli10] existiert ein weiteres verteilt organisiertes System zur Konfiguration von VPN. Ähnlich wie P2PVPN nutzt es ein proprietäres Protokoll, um eine möglichst vollvermaschte Overlay-Struktur auszubilden. Allerdings müssen dazu im Gegensatz zu P2PVPN durch den Nutzer eine Reihe von Sicherheitsbeziehungen so konfiguriert werden, dass ein zusammenhängendes VPN entsteht. Erst dann werden durch tinc weitere Beziehungen hinzugefügt. Im Vergleich zu P2PVPN werden dazu auch eine relativ große Anzahl von Konfigurationsdaten in allen beteiligten Systemen benötigt, wie beispielsweise alle verwendeten öffentlichen Schlüssel, interne IP-Adressbereiche und nach Möglichkeit auch die öffentlichen IP-Adressen. Allerdings ist tinc auch in

der Lage, Verbindungen zwischen zwei von NAT betroffenen VPN-Knoten auszuhandeln.

Auch wenn tinc robust auf Netzwerkpartitionierungen reagieren kann, kann die Robustheit des resultierenden VPN je nach Struktur der manuell erstellten Sicherheitsbeziehungen stark negativ beeinflusst werden, etwa wenn Knoten- oder Kantenzusammenhang zu gering sind. Die vollständige Vermaschung mittels automatisch erstellter Beziehungen limitiert ferner Skalierbarkeit und Effizienz.

Unter Sicherheitsgesichtspunkten bleiben Ende-zu-Ende-Sicherung und PFS nicht berücksichtigt, und da beim periodischen Schlüsselwechsel keine erneute Authentifizierung stattfindet, kann keine dynamische Zugriffskontrolle stattfinden. Ähnlich wie bei der Robustheit wird auch die Verfügbarkeit von tinc-Netzen stark durch den manuell konfigurierten Teil der Netzwerkstruktur geprägt. Zur Wiederherstellung des Netzes unter potentiellen DoS-Angriffen müsste diese auch manuell umkonfiguriert werden. Aber nicht nur durch diesen Aspekt der Konfiguration gestaltet sich ein sicherer Betrieb von tinc-Netzen als komplexe Aufgabe. Vielmehr müssen zusätzliche Einstellungen vorgenommen werden, wie eine Erhöhung der MAC-Bitlänge, um ein mit IPsec oder TLS auch nur ansatzweise vergleichbares Sicherheitsniveau zu erreichen.

CloudVPN

CloudVPN [Kra09a] verwendet ähnlich wie tinc ein unstrukturiertes Overlay-Netzwerk, um verschlüsselte Nutzdaten zu transportieren. Hierbei wird allerdings nicht versucht, eine vollvermaschte dynamische Topologie zu erzeugen; es wird lediglich die Weiterleitung über eine statisch konfigurierte SSL-VPN-Struktur eingerichtet. Innerhalb des VPN annoncieren sich Knoten über Broadcast-Nachrichten, und Weiterleitungspfade werden entlang der Verbindungen zum Ursprungsknoten geschaltet. Dabei kommt als Routing-Metrik die Verzögerung der Broadcast-Nachrichten zum Einsatz. Zwei besondere Eigenschaften von CloudVPN sind die Möglichkeit, Multipath-Routing einzusetzen und Pakete der Schicht 2 weiterzuleiten.

Auch wenn CloudVPN einen Teil der VPN-Konfiguration übernimmt, muss die statische Topologie dennoch manuell eingerichtet werden. Das System besitzt zudem Defizite bei der Transparenz des Gateway-Modus, da jeder durch ein CloudVPN-Gateway angebundene Teilnehmer zunächst ein spezielles *gate* Programm ausführen muss. Dieses fängt Nutzdaten ab und leitet sie zur weiteren Verarbeitung über TCP zum eigentlichen Gateway. Verbindungen über NAT-Gateways können durch CloudVPN hergestellt werden, allerdings ist eine direkte Kommunikation zwischen von NAT betroffenen Kommunikationsendpunkten nicht möglich.

Aus nichtfunktionaler Sicht werden die zu erreichenden Robustheitseigenschaften ähnlich wie bei tinc durch die manuell eingerichtete Topologie dominiert und hängen unter anderem von Knoten- und Kantenzusammenhalt ab. Im Rahmen dieser Grenzen können Pakete umgeleitet werden, und VPN-Teile unabhängig voneinander weiteroperieren. Durch die regelmäßigen Annoncierungen können Skalierbarkeits-

und Effizienzeigenschaften potentiell negativ beeinflusst werden, beispielsweise wenn Pfade häufig ausfallen oder stark vermaschte Topologien konfiguriert werden. Auch CloudVPN bietet keine Ende-zu-Ende-Verschlüsselung, authentisiert keine IP-Adressen und führt keine periodischen Reauthentisierungen durch, sodass Graceful Degradation nicht umgesetzt werden kann. Reaktionen auf DoS-Angriffe sind nur durch manuelle Anpassung der VPN-Topologie möglich.

Fazit

Im Gegensatz zu den zentralen und dezentralen Ansätzen zeichnet sich eine verteilte VPN-Konfiguration aufgrund des Verzichtes auf exponierte Instanzen durch ihre potentiell besseren Eigenschaften in Bezug auf Skalierbarkeit, Robustheit und Verfügbarkeit aus. Darüber hinaus sind im Prinzip wichtige Komponenten schlechter zu beobachten, sodass Infrastructure Hiding ermöglicht wird. Nachteilig wirken sich die oft schlechteren funktionalen Eigenschaften, die Abhängigkeiten von speziellen Netzwerkdiensten und die schlechter zu gewährleistende Sicherheit aus.

3.3 Overlay–Netze zur Steigerung von Robustheit und Verfügbarkeit

Wie in den letzten Abschnitten dargestellt, bieten IP-basierte VPN primär Mechanismen für die Umsetzung der „klassischen“ Sicherheitsziele wie Vertraulichkeit, Integrität und Authentizität. Parallel zu dieser Entwicklung wird über die letzten zehn Jahre die Nutzung von Overlay-Netzen zum Schutz der Verfügbarkeit erforscht. Die Ergebnisse dieser Bestrebungen werden im Folgenden kurz vorgestellt.

3.3.1 Robuste Overlay–Netze

Resilient Overlay Networks (RON) [ABKM01] und Detour [SAA+99] stellen zwei Ansätze eines ähnlichen Konzeptes dar, bei denen vollvermaschte Overlay-Netze zwischen vordefinierten Endpunkten aufgebaut werden. Sind einzelne Knoten über direkte Transportnetzverbindungen aufgrund herkömmlicher Ausfälle oder Sabotageangriffe nicht erreichbar, erfolgt eine automatische Umleitung des Nutzdatenverkehres über geeignete andere Knoten. Ziel beider Ansätze ist darüber hinaus die Steigerung von Durchsatz und die Minimierung von Latenzen, was allerdings stark auf Kosten der Skalierbarkeit geschieht, sodass lediglich Netze mit Größen von unter 50 Knoten sinnvoll erscheinen [SZP+09].

Im Unterschied zu den vorgestellten VPN-Autokonfigutationssystemen konzentrieren sich beide Ansätze auf die Minimierung der Verzögerung und Robustheitsaspekte. Mögliche Sicherheitsprobleme, aber auch Selbstkonfigurationsaspekte werden nicht behandelt.

3.3.2 Sabotageschutz von Netzwerkdiensten durch Overlay–Netze

Neben Overlay-Netzen, die sich selbst gegen Ausfälle des Transportnetzes schützen, existieren eine ganze Reihe von Vorschlägen wie solche Netze als Zwischenvermittler beliebige Dienste gegen DoS-Angriffe absichern können.

Die erste Formulierung der Idee erfolgte in den Secure Overlay Services (SOS) [KMR02], welche Anfragen von Klienten über ein ratenlimitierendes Overlay an einen Server versenden. Die eigentliche IP-Adresse des Servers bleibt auf diese Weise verborgen und DoS-Angriffe werden durch die große Gesamtbandbreite des Overlay-Netzes unmöglich. Mayday [And03], das On-Demand Security Overlay Network (ODON) [KKVS09] und *rewire* [BNW05] greifen die Ideen und Annahmen von SOS auf, und verfeinern den Ansatz mit Fokus auf Effizienzsteigerungen bei Weiterleitung und Ausfallerkennung. FONet [KS06] erweitert den Ansatz durch die Kopplung verschiedener administrativer Domänen zu einem *Federated Overlay*. Die Architektur von OverDoSe [SSAP06] ähnelt ebenfalls SOS, verwendet allerdings Client Puzzles zur Ratenlimitierung. Die Autoren des Migrating OVErlay (MOVE) [SKN+05] schlagen vor, Nutzdaten ohne Umweg durch das Overlay direkt zu übermitteln. Erst bei erfolgreichen Angriffen wird der angebotene Dienst automatisch auf einen nicht betroffenen Overlay-Knoten migriert.

In GONE [FC06] werden die Weiterleitungsmechanismen durch spezielle Transportnetz-Router erbracht, welche selbst am Overlay-Netz teilnehmen und so einen transparenten Schutz bieten sollen.

Die Autoren von [WLC03] analysieren ein System, in dem jeder teilnehmende Klient auch gleichzeitig Teil eines schützenden Overlays im Sinne von SOS ist. Da nun konsequenterweise auch mit internen Angreifern gerechnet werden muss, wird im Artikel die Resistenz von Chord- und CAN-Strukturen gegen zufällige Identifikation der Dienste analytisch bestimmt.

Bis auf den zuletzt vorgestellten Ansatz, setzen die Konzepte somit auf dedizierte Overlay-Systeme zur Abwehr von DoS-Angriffen, deren Konfiguration, Betrieb und Sicherheit schwierig zu realisieren sind. Die Analyse in [WLC03] geht von relativ unrealistischen Rahmenbedingungen aus: Knoten sind hier statisch auf Basis ihrer Identifikatoren verbunden. Durch Dynamik verursachte Effekte sowie garantierte Grenzen werden nicht diskutiert.

3.4 Strukturierte Overlay–Netze für den Datentransport

Anwendungen für die Datenweiterleitung über Overlay-Netze existieren aber nicht nur aus den im letzten Abschnitt vorgestellten Verfügbarkeitsgründen. Stattdessen können innerhalb des Overlays auch weitere Netzwerkdienste implementiert oder generische Daten unter Nutzung eines flachen Adressraumes ausgeliefert werden. In den folgenden Unterabschnitten werden solche Ansätze strukturierter Overlay-Netze zur Umsetzung von generischem Routing und Paketweiterleitung vorgestellt.

3.4.1 Transport in Netzen mit garantierter, direkter Kommunikationsmöglichkeit

Die Internet Indirection Infrastructure (*i*3) [SAS+02, ZLS+05] ist ein Konzept für die generische, skalierbare Implementierung von Multi- und Anycast, sowie für die Behandlung der Mobilität von Endsystemen in Internet-ähnlichen Umgebungen. Dazu wird ein strukturiertes Overlay-Netzwerk eingesetzt, in welchem sich alle beteiligten Endsysteme registrieren und beispielsweise nach anderen Endsystemen oder Multicast-Quellen suchen können. Werden interessante Ziele gefunden, kann *i*3 im Overlay so genannte *Trigger* konfigurieren und über sie das Weiterleiten von Datenpaketen an einen Empfänger veranlassen. In [LAPS03] wird der Einsatz von *i*3 zur Eingrenzung von DoS-Angriffen beschrieben, indem beispielsweise der Paketdurchsatz innerhalb des Overlays limitiert wird.

Ähnlich wie bei VPN-Autokonfigurationssystemen war ein Designziel von *i*3 die vollständige Selbstkonfiguration des Overlay-Netzes. Im Unterschied zu diesen werden Knoten jedoch nicht über IP-Adressen sondern über unstrukturierte, zufällig generierte Identifikatoren angesprochen. Ferner haben die Autoren zwar Sicherheitsanforderungen formuliert, diese entsprechen aber nicht den Kriterien, wie sie in Teilabschnitt 3.1.3 formuliert und in VPN benötigt werden.

3.4.2 Indirekte Knotenanbindung durch Supernodes

Um auch Knoten, die über keine direkte Verbindung zum Internet verfügen, in eine Overlay-Struktur einzubinden wird im Ubiquitous P2P (UP2P) Ansatz [BBF+08] vorgeschlagen, in jedem Teilnetz ein einzelnes strukturiertes Overlay aufzubauen. Diese Teilnetze können von unterschiedlichen Administrationsdomänen betrieben werden und anschließend über so genannte Supernodes an einen globalen Verzeichnisdienst auf DHT-Basis angebunden. Durch diese Anbindung entsteht ein *Federated Overlay*.

Auch die Autoren von UP2P sprechen Sicherheit als ein wichtiges Thema an, versäumen aber die Formulierung konkreter Ziele und Mechanismen. Insbesondere ist der Ansatz der Supernodes in Bezug auf die Verfügbarkeit für ihre jeweiligen Teilnetze als kritisch zu erachten.

3.4.3 Ansätze für Mobile Ad–hoc–Netze (MANET)

Mit dem Unmanaged Internet Protocol (UIP) [For03, For08], dem Virtual Ring Routing (VRR) [CCN+06] und dem Scalable Source Routing [Fuh05, FDKC06] wurden, zumindest teilweise unabhängig voneinander, Protokolle zum Routing in Personal Area Networks beziehungsweise in mobilen Ad-hoc Netzen entwickelt. Alle drei Ansätze basieren auf der Einbettung von DHT in die Topologie des jeweiligen Transportnetzes. Dabei wird zwischen zwar im Overlay benachbarten, aber im Transportnetz nicht direkt erreichbaren Knoten, eine Paketweiterleitung auf Basis von Source Routing oder

virtueller Pfade realisiert. Die Kernidee ist es, das Routing dieser Pfade auch durch Overlay-Anfragen zu bestimmen, und so iterativ das Overlay in das Transportnetz einzugliedern.

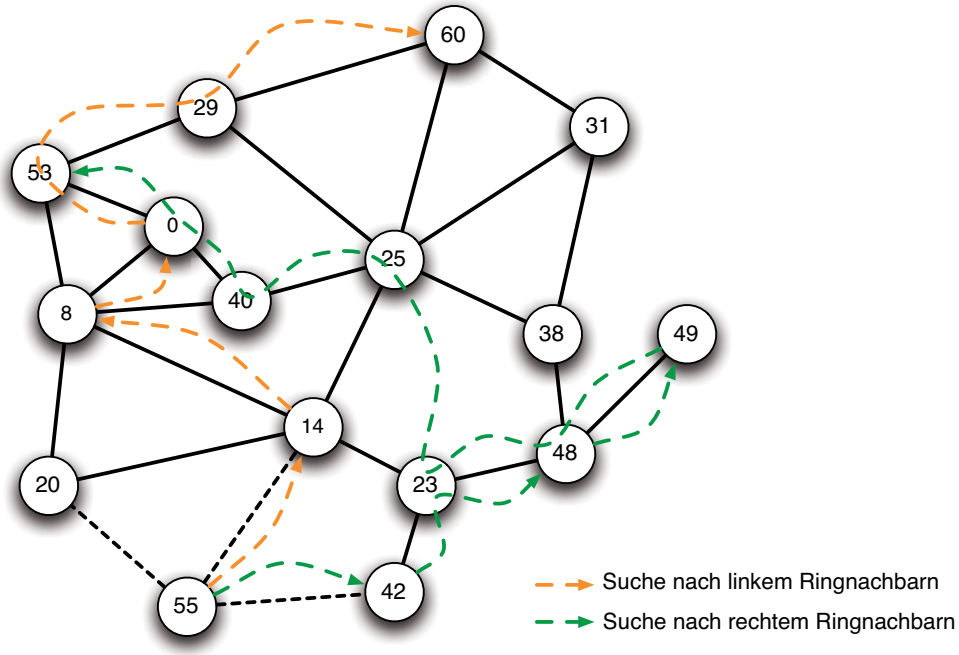


Abbildung 3.5: Beispiel für das Einfügen eines neuen Knotens in eingebettete Peer-to-Peer-Netze

Das in Abbildung 3.5 illustrierte Beispiel zeigt das Einfügen von Knoten 55 in eine geordnete Ringstruktur analog zu den vorgestellten Ideen. Der Knoten besitzt drei Verbindungen zu den Knoten 14, 20 und 42, von denen er aufgrund der Overlay-Distanz Knoten 14 und Knoten 42 zur Suche nach seinem linken und rechten Ringnachbarn verwendet. Knoten 14 sendet erstere an seinen Ringnachbarn 8 und dieser an die 0. Dieser Knoten 0 hat keine direkte Verbindung zu seinem linken Ringnachbarn 60, muss aber einen Pfad aus einer vorherigen Suche kennen. In diesem Beispiel werden die Knoten 53 und 29 zur Weiterleitung verwendet. Daraus ergibt sich ein Pfad zwischen 55 und 60 über die Knoten 14, 8, 0, 53 und 29.

Für den rechten Ringnachbarn erfolgt die Suche analog, allerdings ist im Beispiel eine Besonderheit gezeigt: Die Knoten 23 und 48 werden zweimal durchlaufen. An dieser Stelle wird in [For08] ein Abschneiden des Pfades implementiert, sodass der ermittelte Pfad zwischen Knoten 55 und 53 lediglich über die Knoten 42, 23, 25, 40 und 0 verläuft. Nach dem Einfügen können mithilfe weiterer Suchanfragen Pfade zu den Zielen von Datenpaketen entdeckt werden, entlang dieser später die eigentliche Weiterleitung erfolgt.

Wie bei *i*3 stand bei der Entwicklung dieser Protokolle die Autokonfiguration im Mittelpunkt der Entwicklung und auch hier werden unstrukturierte Identifikatoren verwendet. Darüber hinaus können zwar vermaschte Netze konfiguriert werden, jedoch werden oft nicht die optimalen Routen gefunden. Dies schränkt die Effizienz und Skalierbarkeit in den untersuchten Szenarien teilweise stark ein [MST+09] und ist wohl

der Grund für den bisher geringen Einfluss auf reale Systeme. Die Gewährleistung von Sicherheitseigenschaften wurde nicht näher untersucht.

3.5 Fazit

Insgesamt ergibt sich somit ein extrem heterogenes Bild. Während VPN-Autokonfigurationssysteme Sicherheitsanforderungen im Allgemeinen relativ gut umsetzen, ist die Verfügbarkeit selten ein Entwurfsziel. Forschungsprojekte, bei denen robuste beziehungsweise verfügbare Overlay-Netze konstruiert werden, lassen andere nichtfunktionale Anforderungen außen vor.

Zusammenfassend sind die Eigenschaften der einzelnen VPN-Konfigurationsansätze noch einmal in Abbildung 3.6 dargestellt. Ansätze zur Realisierung robuster Overlays, wie RON, wurden nicht in die Tabelle aufgenommen, da die Überschneidungen in Bezug auf funktionale Anforderungen ohnehin zu gering sind. Im Einzelnen wurde die Selbstkonfiguration mit drei Stufen bewertet (+, Ø, -). Die Anzahl unterstützter Gateways pro VPN (n, 1, 0), die Möglichkeit der Nutzung privater IP-Adressen (+, -), die möglichen Verschachtelungstiefen (n, 1, 0). Die Multicast-Anforderungen (m, u) ergaben sich direkt ohne weitere Einschätzung. Die Möglichkeiten von NAT-Traversal sind mit drei Stufen bewertet: keine Unterstützung über NAT angebundener Knoten „-“, Einbindung und Weiterleitung von NAT-Verkehr „Ø“ und Vermittlung direkter Beziehungen zwischen von NAT betroffenen Knoten „+“.

Die Bewertung der nichtfunktionalen Forderungen nach Robustheit, Skalierbarkeit und Effizienz erfolgt ebenfalls in drei Schritten (+, Ø, -). Auf eine Beurteilung von Agilität wurde verzichtet, da dies zur objektiven Bewertung ein gemeinsames Nutzerszenario voraussetzen würde.

Der untere Teil der Tabelle betrifft die einzelnen Sicherheitsziele. Dabei wurde auch die Ende-zu-Ende-Sicherheit in drei Stufen bewertet: keine Sicherung „-“, keine Sicherung gegen kompromittierte Gateways „Ø“ und garantierte Verschlüsselung bis zum Zielsystem „+“. In den folgenden Zeilen werden die PFS-Unterstützung und die Sicherung gegen verdeckte Kanäle beim Einsatz von VPN-Gateways dokumentiert. Beim Infrastructure Hiding wird die Schwierigkeit, exponierte Punkte wie zentrale Hubs oder Bootstrapping-Server zu finden, bewertet. Ferner bewertet Instanzenauthentisierung, ob einzelne Knoten kryptographisch unterschieden werden können und aufgrund der verwendeten inneren Adressen auf die Identität der beteiligten Knoten ohne Zweifel zurückgeschlossen werden kann. Um Datenintegrität und –authentisierung vollständig zu gewährleisten, muss ein ausreichender kryptographischer Schutz ebenfalls Ende-zu-Ende gewährleistet sein. Statische und dynamische Zugriffskontrolle werden ebenfalls analog zu den Anforderungen bewertet, das heißt, ist es möglich, Zugriffskontrolle auf Basis von Gruppenschlüsseln („Ø“) oder individuellen Zertifikaten („+“) zu realisieren? Wird der Besitz der privaten Schlüssel regelmäßig überprüft und ein Gruppenschlüssel regelmäßig gewechselt? Die minimale sicherheitsrelevante Konfiguration deckt sich in allen aufgeführten Fällen mit der Forderung nach einer einfachen Konfiguration

und wurde daher nicht erneut aufgeführt. Die letzten drei Zeilen beziehen sich auf die zentralen Verfügbarkeitseigenschaften, wie der Fähigkeit, Sabotageangriffen zu widerstehen, betroffene Netzteile leicht zu anderen IP-Adressbereichen zu migrieren und auch unter Anwesenheit interner Angreifer weiterhin die Sicherheitsziele bestmöglich zu garantieren.

Aus der Zusammenstellung lässt sich klar erkennen, dass keiner der vielen heterogenen Ansätze die Anforderungen auch nur ansatzweise erfasst. Beunruhigend ist insbesondere die hohe Anzahl an proprietären kryptographischen Protokollen, welche in diesem Fall — sofern sie überhaupt dokumentiert sind beziehungsweise im Quelltext einsehbar waren — alle Schwachstellen aufwiesen. Aber auch Systeme, die TLS oder IPsec einsetzen, sind oft unsicher, da beispielsweise in keinem Fall private innere Adressen kryptographisch attestiert werden. Die zeitgleichen Entwicklungen zum Schutz gegen DoS-Angriffe haben bisher keinerlei Eingang in VPN-Konfigurationssysteme gefunden. Es besteht daher ein großer Raum für Verbesserungen, wie sie in den folgenden Kapiteln diskutiert werden.

			zentral			dezentral					verteilt								
Allgemeine Eigenschaften	Topologie																		
	Ansatz		Easy VPN	Group Encrypted Transport (GET)	Hamachi[2]	Key distribution via DNSSEC	Wippien	Social VPN	N2N	DMVPN	Opportunistic Encryption	Cryptographically Generated Addresses	Tunnel Endpoint Discovery	Security Policy Protocol	Proactive Multicast IPSEC Discovery Protocol	WASTE	P2PVPN	tinc	CloudVPN
	Protokollschicht der VPN-Daten		3	3	4	3	4	4	4	3	3	3	3	3	3	4	4	4	4
	Protokollschicht der Nutzdaten		3	3	3	3	3	3	2	3	3	3	3	3	3	7	3	2/3	2/3
Funktionale Eigenschaften	Selbstkonfiguration		Ø	Ø	+	+	+	+	Ø	-	+	+	Ø	Ø	Ø	+	+	-	Ø
	Gateway-Funktion		n	n	1	0	0	0	n	n	0	0	n	n	n	0	0	n	0
	Private Adressen		+	-	+	-	+		+	+	-	-	-	-	+	+	+	+	+
	Verschachtelung		-	-	-	-	-	1	1	1/n	-	-	-	-	-	-	n	n	n
	Uni-/Multicast		u	u	u/m	u	u	u	u	u	u	u	u	u	m	u	u	u	u
	NAT Traversal		Ø	-	+	-	+	+	Ø	Ø	-	-	-	-	-	-	Ø	+	Ø
Nicht-funktionale Eigenschaften	Robustheit		-	-	-	-	-	+	+	Ø	Ø	Ø	Ø	Ø	Ø	Ø	+	Ø	Ø
	Skalierbarkeit		-	+	Ø	+	Ø	+	-	Ø	+	+	+	+	Ø	-	-	-	-
	Effizienz		+	+	Ø	+	Ø	+	+	+	+	+	+	+	-	-	-	-	-
Sicherheit	Vertraulichkeit	Ende-zu-Ende Sicherung	-	-	+	+	+	+	+	-	-	Ø	+	+	+	-	-	-	-
		PFS	+	-	+	+	-	+	-	+	+	+	+	+	+	-	-	-	+
		Schutz gegen verdeckte Kanäle	+	-	+	NA	NA	NA	+	+	NA	NA	Ø	Ø	+	NA	NA	+	+
		Infrastructure Hiding	-	-	-	-	-	+	-	Ø	NA	NA	Ø	Ø	-	+	Ø	+	+
	Instanzen-Authentisierung		+	-	?	+	-	+	-	Ø	-	Ø	+	+	Ø	Ø	-	+	Ø
	Datenintegrität/ -authentisierung		Ø	-	?	+	?	+	-	Ø	+	+	+	+	+	-	-	Ø	Ø
	Statische Zugriffskontrolle		+	+	+	-	+	+	Ø	+	-	-	+	+	+	Ø	Ø	+	+
	Dynamische Zugriffskontrolle		+	-	-	-	-	+	-	Ø	-	-	+	+	-	+	-	-	-
	Verfügbarkeit	DoS-Resistenz	-	-	Ø	-	-	Ø	-	Ø	+	+	+	+	-	Ø	Ø	Ø	Ø
		Graceful Degradation	-	-	-	+	+	+	-	-	+	+	+	+	-	-	-	Ø	-
		DoS-Recovery	-	-	-	-	-	-	-	-	+	Ø	-	-	-	+	+	-	Ø

Abbildung 3.6: Zusammenfassender Vergleich der vorgestellten VPN-Autokonfigurationsansätze

Unsere Sicherheiten dürfen nichts Starres werden, sonst brechen sie.

(Robert Walser)

Kapitel 4

Secure OverLay for IPsec Discovery (SOLID)

Aufgrund der Defizite der im letzten Kapitel vorgestellten Ansätze zur Autokonfiguration von VPN, wurde im Rahmen von [Ros07] und dieser Arbeit ein neues System, mit dem Namen Secure OverLay for IPsec Discovery (SOLID), entwickelt und unter anderem in [RSS10, RSSM09, RSM10b] publiziert. Dieses Kapitel leitet prinzipielle Entwurfsentscheidungen aus dem Lösungsraum ab. Dazu wird im folgenden Abschnitt auf grundlegende Annahmen und Funktionen der VPN-Autokonfiguration eingegangen. In den weiteren Abschnitten folgen detaillierte Erläuterungen zu den einzelnen Funktionen wie Bootstrapping, Discovery, Routing und Topologiekontrolle sowie der Multicast-Unterstützung.

4.1 Grundlegende Prinzipien und Annahmen der VPN–Autokonfiguration

In den folgenden vier Unterabschnitten wird zunächst auf die unterschiedlichen Kommunikationsebenen in einem VPN eingegangen. Anschließend erfolgt eine Aufzählung und Erläuterung relevanter Annahmen, bevor Kernfunktionalitäten von VPN-Autokonfigurationsmechanismen und Angreifermodelle behandelt werden.

4.1.1 Netzstruktur

Die Struktur eines VPN kann auf drei Ebenen eines Netzwerkschichtenmodells beschrieben und beeinflusst werden (siehe auch Abbildung 4.1). Die unterste relevante Schicht für VPN ist das Transportnetz, an welches die VPN-Knoten angeschlossen beziehungsweise in welches sie eingebettet sind.

Die mittlere der drei Schichten beschreibt die Beziehungen der VPN-Knoten, welche direkt miteinander über das Transportnetz kommunizieren können. Die eigentlichen Netzwerk-Router spielen nur noch indirekt durch ihre Eigenschaften in Bezug auf Stabilität und Leistungsfähigkeit eine Rolle. Da in VPN typischerweise viele Geräte

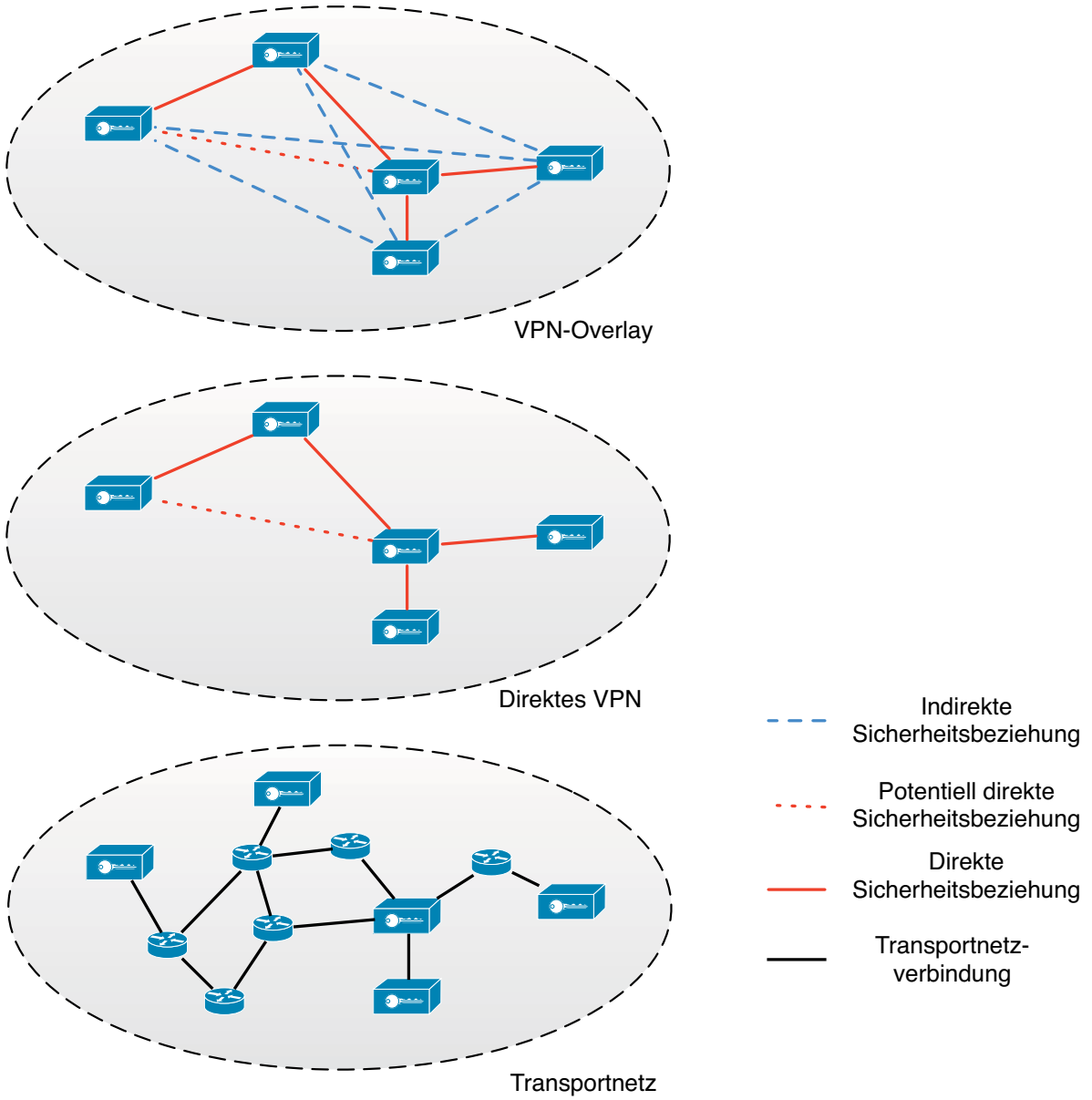


Abbildung 4.1: Schichten eines VPN: Transportnetz, direktes VPN und VPN-Overlay

direkt über das Transportnetz kommunizieren können, entstehen potentiell sehr große Cliquen, also vollständig vermaschte Knotengruppen. Wie ebenfalls in Abbildung 4.1 illustriert wird, kann daher unter Umständen auf den proaktiven Aufbau einiger Verbindungen in dieser Schicht verzichtet werden.

Knoten, welche nicht direkt benachbart sind, können nur miteinander kommunizieren, wenn ein Routing-Verfahren innerhalb des VPN zum Einsatz kommt. Die oberste Schicht stellt daher ein VPN-Overlay mit zusätzlichen indirekten Sicherheitsbeziehungen dar, welche durch das Transportnetz nicht direkt aufgebaut werden können, sondern über direkte Sicherheitsbeziehungen weitergeleitet werden. Spätestens auf dieser Ebene ist es möglich, dass jeder VPN-Knoten mit jedem anderen kommunizieren kann.

4.1.2 Annahmen

Alle folgenden Abschnitte basieren auf den folgenden Annahmen:

1. Die dem VPN zugrunde liegenden kryptographischen Algorithmen und Protokolle, wie beispielsweise AES und IPsec, sind in Bezug auf ihre Sicherheitseigenschaften wirksam. Das bedeutet insbesondere, dass sie in der Lage sind, die Vertraulichkeit, Integrität und Authentizität der übertragenen Daten sicherzustellen.

2. Ferner werden die Zertifikate für die VPN-Teilnehmer von einer vertrauenswürdigen Partei ausgestellt, die mit der Ausstellung gleichzeitig eine Zugehörigkeit zu dem jeweiligen VPN attestiert.

3. Alle Knoten eines VPN dürfen zumindest auf Overlay-Ebene ohne Einschränkung kommunizieren. Auf eine Verwendung von VPN-Policies zur Umsetzung von feinerer Zugriffskontrollmechanismen, wie in manuell konfigurierten VPN unter Umständen möglich, wird verzichtet. Diese sind in der Regel nicht formal spezifiziert und somit nicht automatisch zu konfigurieren und stellen außerdem eine Doppelung von Firewall-Funktionalität dar.

4. Unkompromittierte VPN-Knoten verhalten sich kooperativ und leiten beispielsweise Verkehr für andere Teilnehmer weiter. Da VPN normalerweise von einer einzelnen Organisation betrieben werden, stellt diese Annahme im Allgemeinen keine signifikante Einschränkung dar.

4.1.3 Kernfunktionen

VPN-Autokonfigurationsmechanismen müssen im Allgemeinen die folgenden fünf Aufgaben innerhalb dieser Netzstruktur erfüllen:

- **Bootstrapping:** VPN-Knoten müssen nach dem ersten Start oder nach Ausfällen die IP-Adressen mindestens eines anderen Knotens desselben VPN ermitteln. Dabei können entweder gleich alle relevanten VPN-Knoten ermittelt werden, oder aber die ermittelten Knoten erlauben eine gerichtete Suche nach diesen. Falls ein VPN tolerant gegenüber Netzwerkpartitionierungen werden soll, müssen Bootstrapping-Dienste darüber hinaus periodisch ausgeführt werden, um VPN-Knoten anderer Partitionen zu ermitteln.

- **Adresszuweisung:** Werden private IP-Adressbereiche innerhalb des VPN eingesetzt, müssen VPN-Gateways nach dem ersten Start einen Bereich für ihre inneren Teilnetze reservieren. Die Adressbereiche fungieren für die Dienste der höheren Schichten als eine Art Identität für die Klienten und werden darüber hinaus für Discovery, Routing und Zugriffskontrolle durch Anwendungen und Firewalls benötigt. Daher ist es wichtig, dass die Adresszuweisung überlappungsfrei und sicher geschieht.

- **Discovery:** Empfängt ein VPN-Gateway ein Datenpaket, für welches noch keine VPN-Verbindung existiert, so muss das entsprechende Ziel-Gateway anhand des Adressbereiches seines Subnetzes ermittelt werden.
- **Routing und Paketweiterleitung:** In direktem Zusammenhang zur Discovery steht das Problem, wie Datenpakete durch das VPN weitergeleitet werden sollen. Dabei misst sich die Effektivität von Routing-Mechanismen in VPN an der Anzahl der VPN-Gateways über die Pakete geleitet werden, da in jedem Schritt eine rechenzeitintensive Umverschlüsselung erfolgen muss.
- **Topologiekontrolle:** In dynamischen VPN müssen relevante Verbindungen identifiziert werden, die proaktiv aufgebaut werden, damit Routing und Discovery effizient erfolgen. Anpassungen an der VPN-Topologie müssen dabei kontinuierlich erfolgen, sodass das Netz nicht durch Knotendynamik zerfällt, beziehungsweise auf Veränderungen in der Transportnetzstruktur eingegangen werden kann.

4.1.4 Angreifermodell

Durch ihre Position in der Netzstruktur können auch die unterschiedlichen Angreifertypen abgeleitet werden. So wird im Folgenden primär nach Angreifern unterschieden, die sich außerhalb beziehungsweise innerhalb des VPN befinden.

Externe Angreifer

Das Kontrollieren von Bereichen des Transportnetzes, welche ebenfalls von einem VPN benutzt werden, ermöglicht einem potentiellen Angreifer

- das passive Durchführen von Verkehrsflussanalysen,
- die Identifikation weiterer VPN-Knoten außerhalb des beobachteten Bereiches (zum Beispiel um gezielte DoS-Angriffe durchzuführen) und
- das vollständige oder selektive Unterbrechen des Datenflusses in diesem Bereich.

Vertraulichkeit, Integrität und Authentizität der transportierten Daten sowie kontrollierter Zugang werden jedoch nach Annahme 1 durch die kryptographischen Protokolle weiter gewährleistet.

Interne Angreifer

Befindet sich ein Angreifer innerhalb eines VPN, so muss je nach Mächtigkeit in weitere Subtypen unterschieden werden:

- **Kompromittierter Klient-Computer:** Verfügt ein Angreifer über unbeschränkten Zugriff auf ein Gerät in einem privaten Teilnetz, so wird angenommen er könne:
  - Datenverkehr aus und zu diesem Netz mitlesen und manipulieren,

  - durch das Generieren von Paketen Einfluss auf reaktive Teile der Topologiekontrolle ausüben,
  - Datenverkehr mit beliebigen Ursprungsadressen aus diesem Netz generieren, um DoS-Angriffe innerhalb des VPN durchzuführen.

- **Kompromittierter VPN-Knoten:** Durch die Kontrolle eines VPN-Knotens besitzt ein Angreifer zusätzlich die Möglichkeit
  - der Einflussnahme auf das Signalisierungsprotokoll des Autokonfigurationsmechanismus,
  - des vollständigen oder selektiven Unterbrechens des über ihn weitergeleiteten Datenflusses,
  - der Durchführung von Verkehrsflussanalysen und
  - bei der Weiterleitung von Klartextdaten deren Inhalt einzusehen beziehungsweise zu manipulieren.
- **Kompromittierter VPN-Knoten mit mehreren Instanzen:** Erhält ein Angreifer direkten Zugriff auf den privaten Schlüssel eines VPN-Knotens, beispielsweise durch das Auslesen einer Smartcard, so ist er unter Umständen zusätzlich in der Lage, mehrere von ihm kontrollierte Instanzen mit der Identität des Knotens in das VPN einzugliedern. Auf diese Weise erhält er größeren Einfluss, um beispielsweise Verkehrsflussanalysen durchzuführen.
- **Mehrere kompromittierte VPN-Knoten:** Angreifer, welche mehr als einen Knoten kontrollieren, können mit den verschiedenen Identitäten erweiterte Angriffe durchführen, indem sie sich beispielsweise in Routing-Protokollen attestieren, eine kurze, direkte Verbindung zu besitzen, und so mehr Verkehr anziehen (Sinkhole-Angriff).

In den folgenden Abschnitten wird auf die Realisierung der einzelnen Dienste innerhalb von SOLID eingegangen.

4.2 Bootstrapping

Für das Einbinden neuer oder zeitweilig getrennter Knoten in ein VPN existieren keine Ansätze, die in allen Szenarien zufriedenstellend arbeiten. Daher wird zum einen die Abhängigkeit von diesem Dienst in SOLID minimiert und dient lediglich dem Auffinden eines ersten Knotens. Zum anderen kommt je nach Szenario eine Reihe von Strategien zur direkten Kontaktierung von VPN-Knoten zum Einsatz:

- **Alte relevante Knoten:** War ein VPN-Knoten nur zeitweilig vom VPN getrennt, genügt es in der Regel Verbindungen zu vormaligen Nachbarknoten aufzubauen. Auch zur Überwindung von Netzwerkpartitionierungen können die IP-Adressen vormaliger Nachbarn regelmäßig kontaktiert werden.

- **Internet-Gateway:** Befindet sich ein VPN-Knoten in einer geschachtelten Topologie, genügt es das beispielsweise über das Dynamic Host Configuration Protocol (DHCP) erhaltene Internet-Gateway auf kompatible VPN-Zugehörigkeit zu überprüfen.
- **Multi- oder Broadcast:** Innerhalb lokaler Netze kann darüber hinaus auch über Broad- oder Multicast-Nachrichten nach potentiellen Nachbarknoten gesucht werden.
- **Statische Knoten:** Um auch in globalen VPN die Knoten ohne weitere Infrastrukturen neu einzufügen, besteht keine andere Möglichkeit als zumindest einen anderen VPN-Knoten durch einen Administrator statisch vorzukonfigurieren. Diesem obliegt es dabei auch sicherzustellen, dass ein zusammenhängendes VPN entsteht.

Um im globalen Szenario nicht auf aktuelle statische Definitionen angewiesen zu sein, ist es sinnvoll, alternativ einen der folgenden, externen Verzeichnisdienste zur Ermittlung eines ersten Knotens hinzuzuziehen:

- **Lightweight Directory Access Protocol (LDAP):** VPN-Knoten können einen LDAP-Server nutzen, um eine Nutzer- oder Gerätedatenbank direkt über eine TLS gesicherte Verbindung abzufragen. Dieser Ansatz ist zwar oft leicht zu implementieren und in Bezug auf Vertraulichkeit, Authentizität, Integrität und kontrollierten Zugang sehr gut abzusichern, allerdings ist die Verfügbarkeit des Bootstrapping-Dienstes schwer zu gewährleisten, da der LDAP-Server einen Single Point of Failure darstellt.
- **DNS:** Alternativ kann das DNS-Protokoll verwendet werden, um IP-Adressen bekanntzugeben. Authentizität und Integrität können dabei zwar über DNSSEC garantiert werden, allerdings lässt sich Vertraulichkeit nur über nicht standardkonforme Erweiterungen einbringen. Darüber hinaus hängt auch in diesem Fall die Verfügbarkeit von den autoritativen DNS-Servern ab.
- **Nutzung anderer dezentraler Dienste:** Neben Verzeichnisdiensten, welche für genau diesen Zweck entwickelt wurden, können auch Systeme zweckentfremdet werden, welche eigentlich andere Dienste bereitstellen. Insbesondere die Nutzung des Internet Relay Chat (IRC) [KHW+09] und Jabber [WJBF10] wurden in diesem Zusammenhang vorgeschlagen. Beide Architekturen zeichnen sich durch eine dezentrale Organisation aus und ermöglichen im Prinzip auch einen kryptographischen Austausch zur Absicherung des Bootstrapping-Vorgangs. Allerdings ist auch hier der Ausfall des Rendezvous-Servers kritisch für das System.
- **Öffentliche DHT:** VPN könnten darüber hinaus über ein öffentliches Peer-to-Peer-Netzwerk aufgebaut werden. Dazu wird in [DJWC08, DW09] vorgeschlagen, systematisch Verbindungsanfragen weit verbreiteter Systeme an zufällige IP-Adressen und Ports zu versenden. Wird nach einigen Minuten oder Stunden ein erster Teilnehmer des Peer-to-Peer-Netzwerks gefunden, kann über diesen dem Peer-to-Peer-Netz beigetreten werden. Anschließend kann eine gezielte Suche nach anderen Teilnehmern des gleichen VPN erfolgen. Im Vergleich zu anderen Verzeichnisdiensten ist dieses Verfahren vollständig verteilt, und dadurch relativ

sicher gegen gezielte DoS-Angriffe. Allerdings kann ein Beitritt potentiell relativ lange dauern, und das aggressive Versenden von Nachrichten macht unter Umständen erst auf ein VPN aufmerksam, zumindest bei einem häufigen Initiieren des Bootstrapping-Vorganges oder großen VPN.

- **Cloud-Dienst:** Für kleine oder mittlere VPN kann das Bootstrapping möglicherweise besser durch einen Fremdanbieter realisiert werden, der *Cloud*-Dienste anbietet (beispielsweise Amazon Simple Storage Service[1]). Diese bieten für einen geringen Preis eine sichere Möglichkeit eigene Dienste mit hohen Verfügbarkeitsgarantien zu realisieren. Die dafür verwendeten Rechenzentren besitzen in der Regel sehr gute Verbindungen zum Internet und dürften somit wirkungsvolle DoS-Angriffe erheblich erschweren. Nachteilig ist unter Umständen nur die Beobachtbarkeit des VPN durch den Anbieter.

4.3 Adresszuweisung

Die Zuweisung von inneren IP-Adressen an einzelne VPN-Knoten muss drei Bedingungen genügen: Sie muss konfliktfrei sein, den kryptographischen Nachweis des Besitzes innerer IP-Adressen erlauben und im Fall von VPN-Gateways Teilnetzgrößen möglichst optimal abschätzen, sodass wenige IP-Adressen verschwendet werden, aber auch kein Mangel auftritt.

Da die Vergabe der IP-Adressen durch zentrale Dienste aufgrund potentieller Verfügbarkeitsprobleme ausgeschlossen werden muss, ist es prinzipiell nur möglich, einen verteilten Algorithmus zu konstruieren oder die Adressen vorzukonfigurieren. Allerdings ergeben sich beim Einsatz verteilter Algorithmen Sicherheitsprobleme, die ohne weitere Vorgaben durch ein Autokonfigurationssystem nicht gelöst werden können:

- **Overclaiming:** Einzelne VPN-Gateways können durch potentiell kompromittierte Klient-Geräte getäuscht werden, sodass sie für ein zu großes Netzwerk IP-Adressen reservieren und auf diese Weise den Adressraum erschöpfen.
- **Mehrfachvergabe von Adressen:** Da Duplicated Address Detection (DAD) innerhalb des VPN aufgrund möglicher Netzpartitionierungen nicht in Betracht gezogen werden kann, muss die Konfliktfreiheit der Adressbereiche über eine systematische Vergabe erfolgen (beispielsweise mittels Binary Split [MM00]). Aber auch bei der Verwendung kryptographischer Attestierungen kann ohne den Einsatz manipulationssicherer Hardware nicht ausgeschlossen werden, dass einzelne Blöcke durch kompromittierte VPN-Knoten mehrfach vergeben werden, da dieser in den Zustand vor der Vergabe zurückgesetzt werden kann.
- **Adressfreigabe:** Treten Knoten dem VPN, beispielsweise aufgrund eines geschlossenen Standortes, nicht länger bei, müssen zumindest auf längere Sicht die reservierten IP-Adressen wieder freigegeben werden. Ein Automatismus ist jedoch mit dem Risiko einer zu frühen Freigabe verbunden.

[1] `http://aws.amazon.com/s3/`

Aufgrund dieser potentiellen Probleme muss für jeden Einzelfall abgewogen werden, inwiefern eine dynamische Vergabe zu kalkulierbaren Risiken führt. Daher wird in SOLID die Adressvergabe durch eine Zertifizierungsinstanz vorgenommen, indem jedem Knoten mindestens ein eindeutiger IP-Adressbereich attestiert wird. Beim Einsatz von IPv6 ist es darüber hinaus denkbar, einen eindeutigen IP-Adressbereich mit fester Subnetzmaske durch Hashing des Zertifikates zu berechnen. Bei IPv4 ist dies aufgrund des kleinen privaten Adressbereiches nicht sinnvoll.

4.4 Discovery

Um eine Abbildung von inneren, roten IP-Adressen auf VPN-Zielknoten und somit eine äußere Adresse zu realisieren, erfolgt in herkömmlichen VPN entweder eine Zuordnung in der Konfiguration jedes Knotens, beispielsweise über die IPsec Security Policy, oder eine Weiterleitung über eine Standardbeziehung an ein zentrales Hub-Gateway. In automatisch konfigurierten VPN erfolgt die Zuordnung im Allgemeinen über einen Discovery-Mechanismus.

Aus Skalierbarkeits- und Robustheitsgründen bietet sich ein reaktiver Aufbau von Sicherheitsbeziehungen und eine Verteilung der Zuordnung mithilfe eines strukturierten Peer-to-Peer-Overlays an, sodass jeder Knoten jeweils nur einen kleinen Teil der Zuordnung erfassen muss. Lediglich in Szenarien, in denen eine direkte Kommunikation über das Transportnetz nicht zwischen allen Teilnehmern möglich ist, muss diese gegebenenfalls auf VPN-Overlay-Ebene realisiert werden.

In den folgenden drei Unterabschnitten werden verschiedene Aspekte der Abbildung einer VPN-Konfiguration in ein Peer-to-Peer-System diskutiert.

4.4.1 Verteilte Datenbank oder direkte Abbildung

Strukturierte Peer-to-Peer-Systeme sind in der Regel als verteilte Datenbank konzipiert. Dabei speichert jeder teilnehmende Knoten Daten für einen Schlüsselbereich, der sich aus seinem Identifikator ableitet. Andere Knoten können die Daten durch Routing von Anfragen zum jeweils verantwortlichen Knoten ermitteln.

Bei der Nutzung von Peer-to-Peer-Strukturen zur VPN-Discovery existiert prinzipiell die Möglichkeit dieses Konzept unverändert zu übernehmen, und einfach die äußeren IP-Adressen, beziehungsweise in indirekten Szenarien eine Route vom speichernden Knoten zum Zielknoten, als Daten zu erfassen. Alternativ können sich die Identifikatoren der Knoten direkt aus ihren inneren IP-Adressen ableiten, sodass jeder Knoten in der Peer-to-Peer-Struktur für sich selbst verantwortlich ist.

Je nach Einsatzszenario bieten beide Möglichkeiten der Realisierung einige Vorteile:

- **Agilität:** Bei Realisierung mittels einer verteilten Datenbank müssen mobile Knoten lediglich die Daten in einem verantwortlichen Knoten aktualisieren, um erreichbar zu bleiben. Im alternativen Ansatz fehlen diese Bezugspunkte, und sie müssen nach einem Adresswechsel wieder vollständig in das Overlay-Netz integriert werden.

- **Effizienz:** Besitzt ein Knoten mehrere innere unzusammenhängende IP-Adressen beziehungsweise Teilnetzbereiche, so muss er sich in der verteilten Datenbank einfach an mehreren Stellen registrieren. Bei der Realisierung als direkte Abbildung muss sich der entsprechende Knoten mehrfach in die Peer-to-Peer-Struktur einbinden, sodass entsprechend viele zusätzliche Sicherheitsbeziehungen aufgebaut werden müssen. Gleichzeitig erreichen Pakete bei der direkten Abbildung unmittelbar den Zielknoten. Somit wird die Verzögerung gesenkt.

- **Sicherheit:** Bei der Realisierung mittels einer verteilten Datenbank besitzt ein fest zugeordneter, verantwortlicher Knoten die Möglichkeit, jegliche eingehende Anfragen nach Belieben zu filtern. Daher hängt die Verfügbarkeit eines Knoten direkt von der Verfügbarkeit und Integrität des korrespondierenden Knotens ab. Bei der alternativen Realisierung ist ein Knoten nur vollständig zu kontrollieren, wenn alle Nachbarknoten mit direkten Sicherheitsbeziehungen kompromittiert werden.

Im Folgenden wird lediglich eine Realisierung der direkten Abbildung weiterverfolgt, da nicht davon auszugehen ist, dass einzelne VPN-Knoten eine hohe Anzahl unzusammenhängender Teilnetzbereiche besitzen. Ferner kann durch Wahl einer geeigneten Peer-to-Peer-Struktur im nächsten Abschnitt der Nachteil für mobile Knoten minimiert werden.

4.4.2 Struktur des Peer–to–Peer–Netzes

Für die Realisierung strukturierter Peer-to-Peer-Systeme haben sich in den vergangenen Jahren hauptsächlich die folgenden vier grundlegenden Ansätze bewährt (siehe auch Teilabschnitt 2.4.1), welche sich im Wesentlichen durch die Wahl der Nachbarknoten und die Routing-Strategie unterscheiden:

- **Mehrdimensionaler Torus:** In CAN besitzen Knoten eine Koordinate in einem k-dimensionalen Torus und Kanten zu den jeweiligen Nachbarn in jeder Dimension. Beim Routing wird die euklidische Distanz minimiert.

- **Ring-Basis:** Algorithmen wie Chord und Pastry konstruieren zunächst einen Ring, wobei die Knoten nach ihren Identifikatoren angeordnet sind. Weiterhin baut jeder Knoten zusätzlich Querverbindungen durch den Ring auf, sodass jeder Knoten jeden anderen in $\mathcal{O}(\log n)$ Schritten erreicht.

- **Generalisierter Hyperwürfel:** In Kademlia und ähnlichen Protokollen stellt jeder Knoten eine Ecke eines approximierten, k-dimensionalen Hyperwürfels dar. Das Routing erfolgt entlang der Kanten und erfordert maximal $\mathcal{O}(\log_k n)$ Schritte.

- **Butterfly-Graph:** Viceroy und darauf aufbauende Ansätze konstruieren so genannte Butterfly-Netze, in welchen sich jeder Knoten in einer von k Ebenen befindet und nach einer speziellen Konstruktionsvorschrift je zwei Links in die Ebene über und unter sich besitzt. Jeder Knoten erreicht jeden anderen in $\mathcal{O}(k) \hat{=} \mathcal{O}(\log n)$ Schritten.

Für den Einsatz in VPN sind Ansätze, die mehrdimensionale Tori oder Hyperwürfel verwenden, weniger geeignet, da sie beim Einfügen eines Knotens zunächst den Aufbau von $\mathcal{O}(k)$ Verbindungen erfordern. Beim Aufbau von weniger Verbindungen kann nicht garantiert werden, dass alle Knoten gefunden werden. Ähnlich verhält es sich bei der Reparatur der Struktur nach Ausfällen.

Besser geeignet sind Systeme, die zunächst eine geringe Anzahl von Verbindungen benötigen. Butterfly-Netze benötigen maximal vier Verbindungen, allerdings sind sowohl Topologiekontrolle als auch Routing-Algorithmen im Vergleich zu anderen Verfahren relativ komplex und schwerer gegen interne Angreifer abzusichern. Beispielsweise müssen Verbindungen auf Grundlage der Identifikatoren der Nachbarknoten angepasst werden, und bestimmte Knoten sind mehr als andere für das Weiterleiten von Verkehr einer bestimmten Gruppe verantwortlich.

Im Fall der Ringalgorithmen sind die Querverbindungen nicht notwendigerweise erforderlich, sodass bereits zwei Verbindungen ein erfolgreiches Routing sicherstellen. Gleichzeitig basieren Routing und Topologiekontrolle lediglich auf der Minimierung der Differenz zwischen Identifikatoren, welche kryptographisch gesichert werden können. Daher wird in SOLID eine Ringstruktur verwendet. Die dadurch zu erreichenden Latenzen bezüglich der Strukturanpassung nach dem Einfügen und Entfernen von Knoten sollten somit minimiert werden und sind in der quantitativen Auswertung näher zu bestimmen.

4.4.3 Querverbindungen auf Basis von Stichproben

Die Verwendung einer Ringstruktur bietet einen weiteren Vorteil: Sie ist nicht notwendigerweise auf eine Gleichverteilung der Identifikatoren angewiesen. Während DHT-Algorithmen eine Lastverteilung durch die Verwendung von Hash-Funktionen sowohl für die Identifikatoren als auch für die Suchschlüssel erreichen, ist dies für die Suche nach VPN-Gateways nicht möglich, da im Allgemeinen variable Subnetzmasken zu berücksichtigen sind. Es ist zwar theoretisch möglich, präfixerhaltende Hash-Funktionen zu konstruieren, dafür muss jedoch die genaue Verteilung des Schlüsselraums bekannt sein. Werden nicht nur VPN-Endgeräte verbunden, oder stehen nicht ausreichend viele IP-Adressen zur Verfügung, um in jedem Fall feste Subnetzmasken zu realisieren, muss auf ein Hashing der Identifikatoren verzichtet werden. Daher sind in SOLID VPN-Knoten genau dann Nachbarn in der Ringstruktur, wenn kein dritter Knoten mit einem zwischen den beiden anderen liegenden, inneren IP-Adressbereich existiert.

Um auf ein Hashing der Schlüsselwerte zu verzichten, verwenden die vorgestellten Skip-Graphen (siehe Teilabschnitt 2.4.2) daher einen zweiten Identifikator zur Bestim-

mung der Querverbindungen. Dieser wird aus einem gleichverteilten Zufallsraum generiert und sichert so asymptotisch eine maximal logarithmische Länge des Suchpfades zu. Die statische Zuweisung von Overlay-Partnern hat jedoch beim Einsatz in VPN einen entscheidenden Nachteil: Die Erreichbarkeit von Knoten kann durch gezielte Störungen sehr gut beeinflusst werden. Darüber hinaus können die Querverbindungen nicht an das Transportnetz und die durch Nutzdaten bereits aufgebauten Sicherheitsbeziehungen angepasst werden.

Daher wird in SOLID ein alternatives System realisiert: Jeder Knoten hält dazu eine Stichprobe S der inneren IP-Adressen anderer am VPN teilnehmenden Knoten vor. Auf Basis dieser Stichprobe werden Querverbindungen eingerichtet, sodass der zu durchsuchende Schlüsselraum in jedem Overlay-Schritt halbiert wird. Effizient wird dies vor allem, weil aufgrund der statischen Zuweisung von IP-Adressen im VPN, sich die Grundlage für diese Stichprobe über die Zeit nicht stark verändert. Die Qualität der Auswahl hängt somit nicht von der Aktualität sondern primär von der Größe der Stichprobe ab. Daher wird der Einfluss des Stichprobenumfangs $|S| = s$ auf die erwartete Länge der Suchpfade $E(l)$ nachfolgend analytisch untersucht. Es wird dabei ohne Beschränkung der Allgemeinheit vorausgesetzt, dass alle Knoten von 0 bis $n-1$ durchnummeriert sind. Im Unterschied zum realen VPN-Konfigurationssystem, werden im Modell darüber hinaus Querverbindungen unidirektional verwendet. Dies führt zu einer Vereinfachung des Modells, und erlaubt es grundsätzlich die gleichen Aussagen zu treffen.

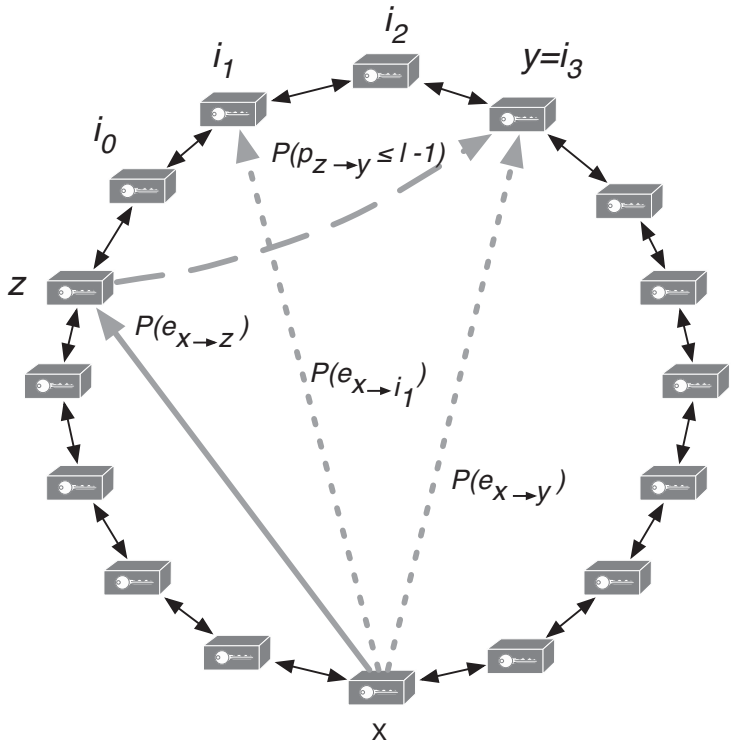


Abbildung 4.2: Beispielskizze der Wahrscheinlichkeiten für Pfadlängen: Ein Pfad der Maximallänge l existiert, falls ein direkter Pfad zwischen Knoten x und z besteht, kein Pfad zu $i_{0,\ldots,3}$ existiert und ein Pfad der Maximallänge $l-1$ von z zu y vorhanden ist.

Die Wahrscheinlichkeit, dass der Pfad zwischen zwei Knoten x und y eine bestimmte maximale Länge l nicht überschreitet $P(p_{x\to y} \leq l)$, ist die Wahrscheinlichkeit, dass x durch den Suchalgorithmus den Knoten y über maximal $l-1$ Zwischenknoten erreicht. Aufgrund des lokalen Wissens im Overlay, wird zur Suche ein Greedy-Verfahren angewendet, das heißt es wird in jedem Schritt ein nächster Knoten z ausgewählt, der $y-z$ minimiert. Im Basisfall bedeutet dies, dass ein Pfad der Länge 1 genau mit der Wahrscheinlichkeit existiert, mit der eine direkte Kante zwischen den Knoten vorhanden ist ($P(e_{x\to y})$). Ist der Pfad länger, so ergibt sich eine Situation, wie sie in Abbildung 4.2 skizziert ist. Die Gesamtwahrscheinlichkeit, dass ein Pfad von x zu y mit maximal l-Overlay-Schritten existiert, ergibt sich dann aus der Wahrscheinlichkeit, dass ein direkter Pfad von x zu einem Knoten z existiert, von welchem wiederum ein Pfad in weniger als l-Schritten zu y führt. Ferner darf keine Sicherheitsbeziehung von x zu einem Knoten i existieren, welcher eine geringere Distanz zu y als z aufweist.

Die $P(p_{x\to y} \leq l)$ besitzen somit die folgenden Berechnungsvorschriften:

$$
\begin{aligned}
P(p_{x\to y} \leq 1) &= P(e_{x\to y}) \\
P(p_{x\to y} \leq 2) &= P(e_{x\to y}) + \sum_{z=x+1}^{y-1} P(\hat{e}_{x\to z,y}) \cdot P(e_{z\to y}) \cdot \prod_{i=z+1}^{y} \left(1 - P(\hat{e}_{x\to i,y})\right) \\
&\vdots \\
P(p_{x\to y} \leq l) &= P(e_{x\to y}) + \sum_{z=x+1}^{y-1} P(\hat{e}_{x\to z,y}) \cdot P(p_{z\to y} \leq l-1) \cdot \prod_{i=z+1}^{y} \left(1 - P(\hat{e}_{x\to i,y})\right)
\end{aligned}
$$

Dabei beziehen sich die Indizes der Summationen und Produkte jeweils auf die arithmetische Ringstruktur, und $P(\hat{e}_{x\to z,y})$ ist die Wahrscheinlichkeit, dass eine Kante von x nach z existiert, unter der Voraussetzung, dass keine bessere Kante zur Verfügung stand. Formal ist $P(\hat{e}_{x\to z,y})$ wie folgt definiert:

$$
P(\hat{e}_{x\to z,y}) = P\left(e_{x\to z} \;\middle|\; \bigcap_{j=z+1}^{y} \overline{e_{x\to j}}\right)
$$

Zunächst zum Basisfall: die Wahrscheinlichkeit, dass ein Pfad von einem Knoten x zu y direkt existiert $P(e_{x\to y})$, ist ein sicheres Ereignis, falls x und y Ringnachbarn sind. Alternativ kommt genau dann eine Verbindung zustande, wenn y eine direkte Querverbindung von x ist. Dafür muss y in der Stichprobe von x sein, und darüber hinaus in der geordneten Stichprobe an genau der Stelle für die Querverbindung für das q-Quantil der Stichprobe liegen. Die Menge gültiger Quantile ist hierbei als $Q = (q_0, \ldots, q_{\lceil \log_2 s \rceil - 1}) = (1/2, 1/4, \ldots, 1/2^{\lceil \log_2 s \rceil})$ definiert. Wird nun das jeweils gewählte Element der Stichprobe einfach durch Berechnung von $\lceil qs \rceil$ bestimmen, kann es bei kleinen Werten vorkommen, dass mehrere Querverbindungen durch die Rundung auf dasselbe Element der Stichprobe zeigen, wodurch das Ergebnis verzerrt würde. Daher bestimmt sich die Wahl der Querverbindungen $R = \{r_0, \ldots\}$ rekursiv:

$$
r_0 = \lceil q_0 \cdot s \rceil = \lceil 1/2 \cdot s \rceil
$$

$$r_i = \begin{cases} \lceil q_i \cdot s \rceil & \text{falls } r_{i-1} \neq \lceil q_i \cdot s \rceil \\ r_{i-1} - 1 & \text{falls } r_{i-1} = \lceil q_i \cdot s \rceil \wedge r_{i-1} > 1 \end{cases}$$

Da somit ein Knoten nicht für mehrere Querverbindungen in Frage kommt, schließen die Einzelereignisse einander aus, und die Wahrscheinlichkeiten können addiert werden. Bei der Stichprobe wird angenommen, dass außer dem Element selbst und den Ringnachbarn alle verbleibenden $n - 3$ Elemente gleich wahrscheinlich vorkommen. Es ergibt sich die Gesamtformel für $P(e_{x \to y})$:

$$P(e_{x \to y}) = \begin{cases} 0 & \text{falls } x = y \\ 1 & \text{falls } |x - z| = 1 \\ \sum\limits_{r_k \in R} P(y\ r_k\text{-Querverbindung für } x) & \text{sonst} \end{cases}$$

Die Wahrscheinlichkeit, dass y als Querverbindung an der Stelle r_k der Stichprobe ausgewählt wird, ist ein einfaches Auswahlproblem ohne Wiederholungen und führt auf eine hypergeometrische Verteilung. Dafür müssen aus der Grundmenge ohne x und seinen Nachbarn genau $r_k - 1$ Elemente kleiner als y und $s - r_k$ Elemente größer y gezogen werden. Wird $(y - x)$ auf die Ringarithmetik bezogen, kann die Wahrscheinlichkeit wie folgt berechnet werden:

$$P(y\ r_k\text{-Querverbindung für } x) = \frac{\binom{(y-x)-2}{r_k - 1}\binom{n-2-(y-x)}{s - r_k}}{\binom{n-3}{s}}$$

Um die Wahrscheinlichkeit $P(\hat{e}_{x \to z,y})$ zu berechnen, also dass eine Verbindung unter der Voraussetzung existiert, dass eine Reihe anderer Verbindungen nicht existieren, muss die Gesamtformel nur wenig angepasst werden. Da die Ringverbindungen unabhängig von der Stichprobe sind, wirkt sich die Voraussetzung allein auf die Auswahl der Querverbindungen aus:

$$P(\hat{e}_{x \to z,y}) = \begin{cases} 0 & \text{falls } x = z \\ 1 & \text{falls } |x - z| = 1 \\ \sum\limits_{r_k \in R} P\left(z\ r_k\text{-Querverbindung für } x \;\middle|\; \bigcap\limits_{j=z+1}^{y} \overline{e_{x \to j}}\right) & \text{sonst} \end{cases}$$

Die Berechnung der entstandenen bedingten Wahrscheinlichkeit ist allerdings erheblich komplexer als der unbedingte Fall. Wie in Abbildung 4.3 illustriert ist, existieren die Elemente $z + 1$ bis y in der Grundmenge, und i von ihnen kommen auch in der Stichprobe vor. Da sie in der Berechnung bereits ausgeschlossen wurden, dürfen sie in der Stichprobe nicht an Positionen für Querverbindungen auftreten. Allerdings steht ihre exakte Anzahl nicht fest, es müssen aber weniger als $r_{k-1} - r_k$ sein, da sonst eines der Elemente zu der r_{k-1}-Querverbindung geworden wäre. Ist $k = 0$, müssen es weniger als $s - r_k$ Elemente sein, da die Größe der Stichprobe nicht mehr Elemente zulässt.

Per Definition sei daher im Folgenden $r_{-1} = s$. Da die Ereignisse für verschiedene i einander ausschließen, können die günstigen Fälle analog zu Abbildung 4.3 für alle in Frage kommenden i wie folgt aufsummiert werden:

$$n_{günstig} = \sum_{i=0}^{r_{k-1}-r_k-1} \binom{(z-x)-2}{r_k-1} \binom{1}{1} \binom{y-z}{i} \binom{n-2-(y-x)}{s-r_k-i}$$

Die Gesamtmenge der Ereignisse, in denen die $(y-z)$ Elemente nicht an Stellen für Querverbindungen erscheinen, ist ebenfalls abhängig von der Anzahl der tatsächlich gezogenen Elemente i. Darüber hinaus kann die Folge der i Elemente an jeder Stelle der geordneten Stichprobe beginnen, sie darf allerdings weder über die Stichprobengröße hinausgehen, noch eine Stelle beinhalten, welche zur Wahl der Querverbindung verwendet wird. Die Werte der Elemente vor und nach der Stichprobe dürfen jeweils frei gewählt werden, sodass n_{gesamt} durch folgenden Term berechnet werden kann:

$$n_{gesamt} = \sum_{i=0}^{y-z} \sum_{r_l \in R} \sum_{q=r_l}^{r_{l-1}-i} \binom{(z-x)-1}{q} \binom{y-z}{i} \binom{n-2-(y-x)}{s-q-i}$$

Für die bedingte Wahrscheinlichkeit, dass z die r_k-Querverbindung für den Knoten x ist, ergibt sich somit die Gesamtformel:

$$P\left(z\ r_k\text{-Querverbindung für } x \;\middle|\; \bigcap_{j=z+1}^{y} \overline{e_{x \to j}}\right) = \frac{n_{günstig}}{n_{gesamt}} =$$

$$\frac{\sum_{i=0}^{r_{k-1}-r_k-1} \binom{(z-x)-2}{r_k-1} \binom{1}{1} \binom{y-z}{i} \binom{n-2-(y-x)}{s-r_k-i}}{\sum_{i=0}^{y-z} \sum_{r_l \in R} \sum_{q=r_l}^{r_{l-1}-i} \binom{(z-x)-1}{q} \binom{y-z}{i} \binom{n-2-(y-x)}{s-q-i}}$$

Mithilfe des Modells können nun die durchschnittlichen Erwartungswerte in Abhängigkeit von Overlay- und Stichprobengröße berechnet werden. Dabei ist zu erwarten,

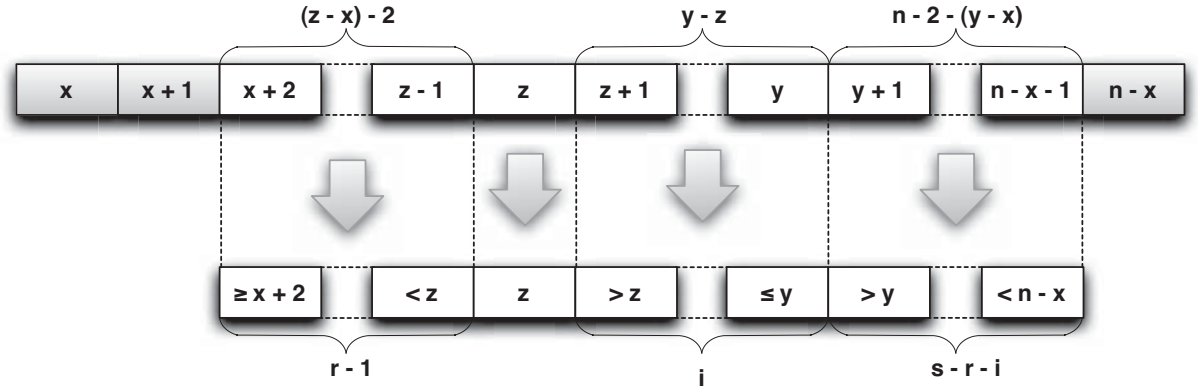


Abbildung 4.3: Abbildung der Grundmenge an Knoten auf einzelne Bereiche der Stichprobe

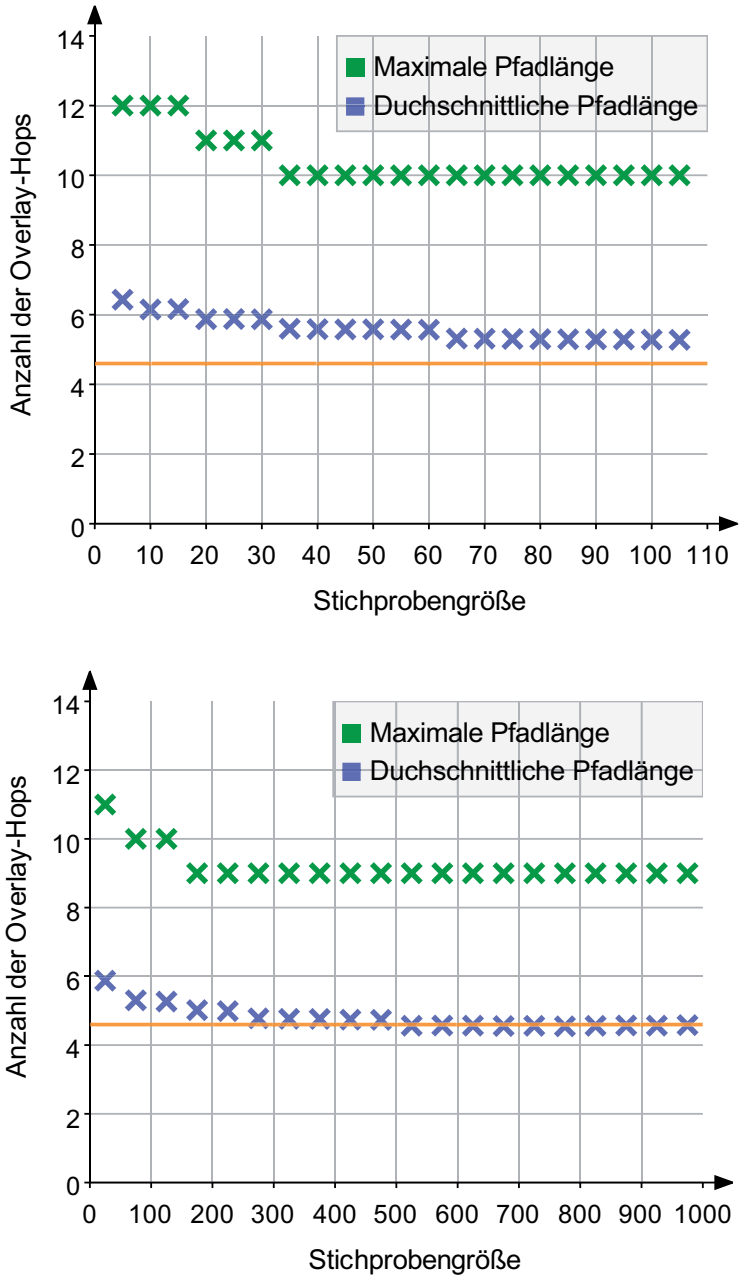


Abbildung 4.4: Erwartete durchschnittliche und maximale Pfadlängen (99%-Quantil) in Abhängigkeit von der Stichprobengröße für $n = 1000$ (links bis $s = 105$ rechts bis $s = 975$)

dass die durchschnittlichen und maximalen Pfadlängen mit Umfang der Stichprobe abnehmen, da zum einen die Abbildung genauer wird und zum anderen die Anzahl der Querverbindungen steigt.

Abbildung 4.4 zeigt zum einen diesen Mittelwert der Erwartungswerte für $n = 1000$ und zum anderen auch das 99%-Quantil der Wahrscheinlichkeiten, also die Länge, unter welcher 99 % der Pfade bleiben.

Wie erwartet sinkt die durchschnittliche Pfadlänge von zunächst 6,4 auf 4,6 Overlay-Hops ab, und das 99%-Quantil sinkt gleichzeitig von 12 auf 9 Hops. Dass Werte nahe

dem eingezeichneten Optimum im durchschnittlichen Fall erst bei einer Stichprobengröße von circa 500 Knoten erreicht werden, liegt allerdings nicht nur an der Qualität der Stichprobe, sondern ist viel mehr stark von der Anzahl an Querverbindungen abhängig. Diese steigen durch die Definition von R lediglich logarithmisch in Bezug auf s und haben einen erheblichen Einfluss.

Wie Abbildung 4.5 zeigt, kann daher eine weitere Optimierung vorgenommen werden. Bei dieser Berechnung wurden gleichverteilt aus der Stichprobe gewählte, zusätzliche Querverbindungen eingefügt, wenn durch die Stichprobe weniger als die optimalen acht Ziele vorgegeben wurden. Mithilfe der zusätzlichen Verbindungen werden zunächst gegenüber dem unbereinigten Modell im durchschnittlichen Fall zwischen 0,3 und einem Hop eingespart und auch im Maximalfall sinkt die benötigte Weglänge bei sehr kleinen Stichprobengrößen (10 und 15) schneller ab. Dies zeigt, dass unter der Annahme der Gleichverteilung der Stichproben bereits relativ schnell gute Overlay-Topologien zu erzeugen sind. Ferner zeigt das bereinigte Modell, dass der Aufbau weiterer Beziehungen, wie zum Beispiel zum Transport von Nutzdaten, je nach Ziel mit hoher Wahrscheinlichkeit größere Verbesserungen als eine Vergrößerung der Stichprobe bewirken.

Im Gegensatz zum Skip-Graph-Verfahren erhalten die Knoten durch die Erfassung der Stichprobe eine insgesamt genauere Sicht auf die aktuelle Ringstruktur. Auf diese Weise wird es beispielsweise möglich, Knoten für Querverbindungen zu wählen, die zwar in Bezug auf die Overlay-Struktur leicht suboptimal erscheinen, deren Selektion aber aufgrund anderer Metriken sinnvoll ist; so können etwa bezüglich des Transportnetzes nahe Knoten bevorzugt werden. Inwiefern dies den Mehraufwand gegenüber Skip-Graphen rechtfertigt, muss allerdings im realen System quantitativ ermittelt werden.

Weiterhin ist es notwendig zu untersuchen, ob die nun nicht notwendigerweise symmetrische Wahl von Querverbindungen eine Auswirkung auf die Balancierung des Knotengrads hat; denn bei der stichprobenbasierten Auswahl könnten, ebenfalls im Unterschied zum Skip-Graph-Ansatz, bestimmte Knoten überdurchschnittlich häufig für Querverbindungen ausgewählt werden. Eine geringere Skalierbarkeit und Robustheit wäre die Folge.

4.5 Routing und Paketweiterleitung

Wird bei der Discovery ein Weg zu einem VPN-Knoten gefunden, so kann durch den Austausch von IP-Adressen unmittelbar eine Sicherheitsbeziehung zwischen den Knoten aufgebaut werden, falls eine direkte Kommunikation über das Transportnetz möglich ist. Ein Routing für die durch das VPN ausgetauschten Daten wird erst benötigt, falls die Knoten sich nicht direkt erreichen können. In folgenden Abschnitten werden die Routing-Metrik, die verschiedenen Routing- und Wei-

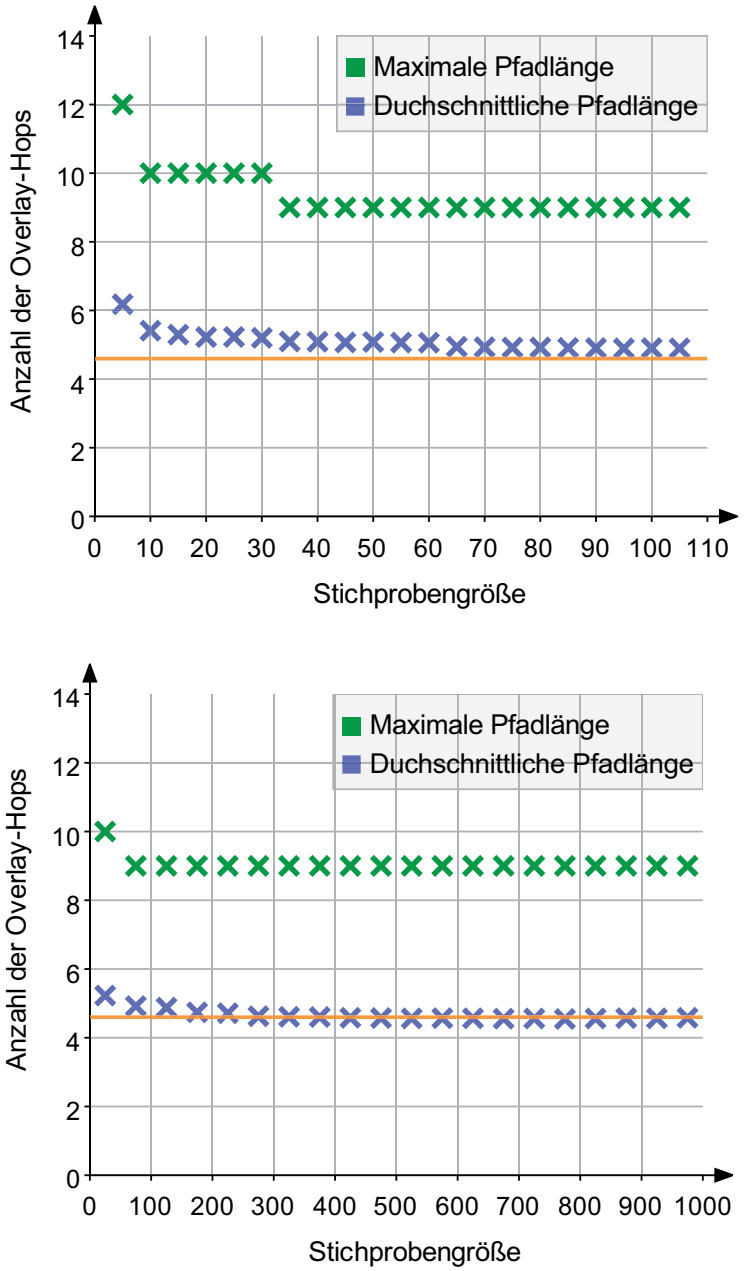


Abbildung 4.5: Erwartete durchschnittliche und maximale Pfadlängen (99%-Quantil) bei zusätzlichen zufälligen Querverbindungen ($n = 1000$, links bis $s = 105$ rechts bis $s = 975$)

terleitungsmechanismen, sowie deren direkte Implikationen auf die VPN-Sicherheit diskutiert.

4.5.1 Routing–Metrik

Wichtigste Metrik bei der Wegewahl in VPN ist die Anzahl der Overlay-Hops, da jeder einzelne VPN-Zwischenknoten zumindest eine Überprüfung der Datenauthentizität und –integrität, sowie gegebenenfalls eine Umverschlüsselung durchführen muss.

Diese kryptographischen Maßnahmen sind normalerweise ressourcenintensiv, sodass sie in der Regel die Verzögerung und oft auch die Bandbreite von VPN-Verbindungen dominieren.

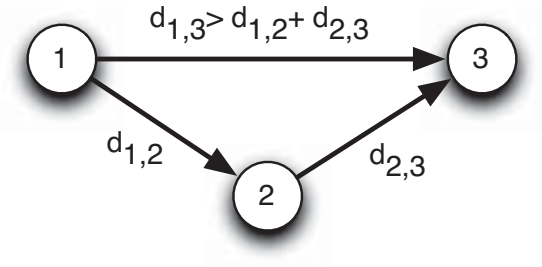


Abbildung 4.6: Transportnetztopologie, bei welcher die Dreiecksungleichung in Bezug auf die Verzögerungszeiten verletzt ist

Trotz direkter Konnektivität ist es theoretisch denkbar, dass durch das Weiterleiten von Paketen über andere Overlay-Knoten Latenzoptimierungen erzielt werden können. Wie in Abbildung 4.6 illustriert, basieren solche Situationen auf der Verletzung der Dreiecksungleichung. Die Anzahl der Verbindungen, bei denen eine so starke Verletzung der Ungleichung erfolgt, dass die Verzögerung durch die Umverschlüsselung kompensiert wird, ist im Allgemeinen aber gering (siehe auch Abschnitt A.2). Somit kann eine direkte Kommunikation zwischen VPN-Knoten immer bevorzugt werden, und es ist keine weitere Optimierung direkter Verbindungen durch das Routing nötig.

4.5.2 Routing–Mechanismen

Im Vergleich zum Routing in Transportnetzen ist das Finden eines Weges für das VPN-Overlay kein unmittelbares Problem: Die VPN-Knoten können einander immer entlang der Routen für die Discovery erreichen. Diese können allerdings bereits in VPN mit direkter Erreichbarkeit $\mathcal{O}(\log n)$-Overlay-Schritte lang werden. Bei der Berücksichtigung indirekter Wege während der Suche kann sich dieser Wert sogar auf $\mathcal{O}(n)$ VPN-Knoten erhöhen. Daher sollte ein Routing-Mechanismus eine reaktive Optimierung der Datenweiterleitung vornehmen.

Verteilung des Wissens

Link-State-, Distanzvektor- und Pfadvektorverfahren werden in Transportnetzen eingesetzt, um aktuelle Topologieinformationen im gesamten Netzwerk zu verbreiten. Für die flexibleren VPN-Topologien mit reaktiv aufgebauten Verbindungen scheinen diese Protokolle nur bedingt geeignet, da das Fluten selbst kleiner Topologieänderungen erhebliche Skalierbarkeits- und Effizienzprobleme bewirken kann. Potentielle externe Angreifer können darüber hinaus durch periodisches kurzes Unterbrechen

von Sicherheitsbeziehungen das Versenden von Rekonfigurationsmeldungen im gesamten VPN bewirken. Ferner wird bei der Anwesenheit internen Angreifern sofort die gesamte oder zumindest ein nicht unerheblicher Teil der VPN-Topologie bekannt.

Strukturierte Einbettung

Alternativ kann das Routing in der VPN-Ebene ebenfalls über das strukturierte Overlay erfolgen, bei welchem durch den Aufbau bestimmter Verbindungen eine schnelle reaktive Knotensuche ermöglicht wird. Der Vorteil dieses Ansatzes ist, dass einzelne Topologieänderungen nicht im gesamten VPN annonciert werden müssen.

Können alle VPN-Knoten einander direkt über das Transportnetz erreichen, kann dessen Routing direkt für das VPN verwendet werden und alle Pakete zunächst über den Discovery-Pfad und anschließend über direkte Sicherheitsbeziehungen ausgetauscht werden. Auch in hierarchischen Netzwerkumgebungen, wie bei UP2P (vergleiche Teilabschnitt 3.4.2) vorgesehen, gestaltet sich das Routing einfach: Entweder können Knoten im lokalen Overlay gefunden und direkt mit ihnen über das Transportnetz kommuniziert werden, oder die Verbindung erfolgt über den Supernode des Netzes, welcher sich direkt zu dem korrespondierenden Supernode verbinden kann.

Problematisch an beiden Ansätzen ist die fehlende Behandlung partieller Transportnetzausfälle und mangelnde Unterstützung komplexerer Netztopologien. Aus diesem Grund wird in SOLID kein vereinfachtes Routing wie bei i^3 oder UP2P verwendet, sondern es werden auch Wege für den iterativen Aufbau des VPN-Overlays selbst mithilfe des Discovery-Verfahrens bestimmt. Wie es in den folgenden Abschnitten näher beschrieben wird, werden diese Pfade dann auch je nach topologischen Gegebenheiten abgekürzt.

4.5.3 Abkürzungsalgorithmen

Werden die Discovery-Pfade für längere Zeit zur Weiterleitung von Nutzdaten verwendet oder von der Topologiekontrolle aufrechterhalten, erfolgt eine Abkürzung durch eine Reihe verschiedener, sich ergänzender Topologieoptimierungen.

Der Umlenkungsalgorithmus

Die Basis für die Optimierungen bildet die Umlenkung von Datenströmen, wie in Abbildung 4.7 illustriert. Dabei stellt ein VPN-Knoten fest, wenn Daten über die gleiche Netzwerkschnittstelle versandt werden, über die sie auch empfangen wurden. In solchen Fällen wird der Ursprungsknoten über die Adresse des nächsten Knotens auf dem Pfad informiert. Der Ursprungsknoten kann so, nach einer Konsistenzprüfung, seine

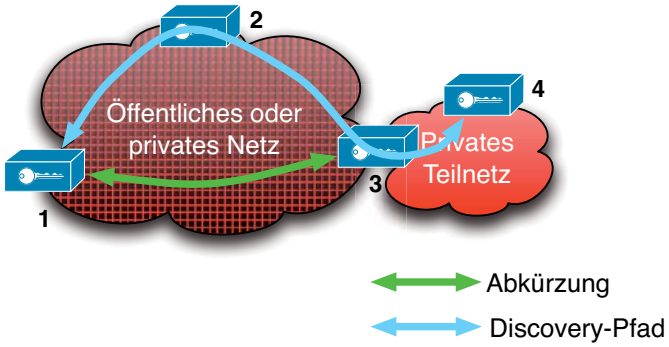


Abbildung 4.7: Umlenkung von Sicherheitsbeziehungen: Knoten 2 teilt 1 mit, dass Knoten 4 besser über Gateway 3 erreicht wird.

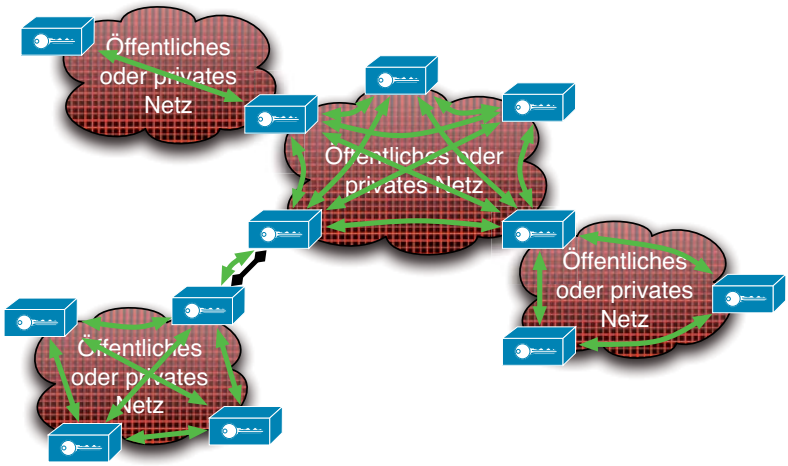


Abbildung 4.8: Transportnetztopologie, die optimale Abkürzungen auf Basis von Umleitungen erlaubt

Pfade aktualisieren beziehungsweise zunächst einen Verbindungsaufbau initiieren. Auf diese Weise wird der Pfad nach und nach optimiert.

Abbildung 4.8 zeigt eine Transportnetztopologie, die sich besonders für eine solche Optimierung eignet. Die einzelnen Teilnetze sind hierbei zyklenfrei verbunden, und innerhalb der Teilnetze bilden die einzelnen VPN-Knoten potentiell Cliquen aus, welche aufgrund der Transportnetzstruktur jeweils einen gemeinsamen Knoten haben. Formal handelt es sich um einen Graphen $G = (V, E)$, bei dem aus jedem Kreis $(v_0, v_1, \ldots, v_n, v_0)$ folgt, dass die Kanten $e_{x,y}$ für alle paarweise verschiedenen Knoten v_x und v_y existieren. Der Umlenkungsalgorithmus kann in diesem Szenario anhand des Vergleiches von ein- und ausgehender Netzwerkschnittstelle feststellen, ob Vorgänger und Nachfolger im Pfad derselben Clique angehören, und somit ein direkter Pfad zwischen beiden existieren könnte. Wie im Folgenden gezeigt, wird bereits durch den einfachen Umlenkungsalgorithmus in solchen, häufig auftretenden VPN-Topologien eine optimale Pfadwahl erreicht.

Satz:
In VPN, bei denen für jeden potentiellen Kreis gilt $\exists(v_0, v_1, \ldots, v_n, v_0) \Rightarrow (\exists\, e_{x,y} \mid \forall x, y \in \{0, \ldots, n\} \wedge x \neq y)$, führt der Umlenkungsalgorithmus immer zu Pfaden mit minimaler Anzahl an Overlay-Hops.

Beweis durch Kontraposition. Angenommen der Umlenkungsalgorithmus findet in gegebenen Graphen keine optimalen Topologien

$\Rightarrow$ $\exists$ nicht-optimaler Pfad $\mathcal{P} = (v_0, \ldots, v_{n-1})$ von Knoten v_0 zu Knoten v_{n-1}, durch den Umlenkungsalgorithmus optimiert
$\exists$ optimaler Pfad $\mathcal{O} = (o_0, \ldots o_{m-1})$ mit $o_0 = v_0 \wedge o_{m-1} = v_{n-1}$
$\exists\, x$, sodass $\forall\, i \leq x\ o_i = v_i \wedge o_{x+1} \neq v_{x+1}$
$\exists\, y, z$, sodass $y > x \wedge z > x \wedge o_{y-1} \neq v_{z-1} \wedge o_y = v_z$

Ohne Beschränkung der Allgemeinheit sollen y und z minimal sein. Das bedeutet, wenn $\mathcal{P}$ und $\mathcal{O}$ mehrere gemeinsame Teilstücke haben, soll das erste ungleiche gewählt werden.

Fall 1 o_x, o_y sind Teil einer Clique C
Fall 1.1 $\forall\, j \in \{x+1, \ldots, z-1\} :\ v_j \in C$

$\Rightarrow$ Umlenkungsalgorithmus hätte in Knoten v_j erkannt, dass v_{j-1} und v_{j+1} direkt kommunizieren können $\lightning$

Fall 1.2 $\exists\, j \in \{x+1, \ldots, z-1\} :\ v_j \notin C$ und $v_{j-1}, v_{j+1} \in C$

$\Rightarrow\ v_{j-1} = v_{j+1}$ nach Voraussetzung der Graphenstruktur
$\Rightarrow$ Umlenkungsalgorithmus hätte doppelte Benutzung der Kante erkannt, daher hätte v_j Knoten v_{j-1} angewiesen die Daten direkt weiterzuleiten $\lightning$

Fall 1.3 $\exists\, j \in \{x+1, \ldots, z-1\} :\ v_j, v_{j+1} \notin C$ und $v_{j-1} \in C$

$\Rightarrow\ \exists\, v_k$ mit $v_k = v_j \wedge k \in \{x+1, \ldots, z-1\} \setminus j$, da $v_z = o_y$ nach Fallvoraussetzung wieder Teil von C ist, und der Pfad v_{j-1} nach v_z nur erneut über v_{j-1} nach C führen kann ohne die Voraussetzung zu verletzen
$\Rightarrow$ Die entstandene Schleife muss aber nach vorausgesetzter Graphenstruktur in einem Einzelknoten oder einer Clique „umkehren“, was einem Widerspruch analog zu 1.1 oder 1.2 entspricht $\lightning$

Fall 2 $o_x \in C_1,\ o_y \in C_2,\ C_1 \neq C_2$
$\forall\, j \in \{x+1, \ldots, z-1\} : v_j \notin \{o_{x+1}, \ldots, o_{y-1}\}$ (nach Voraussetzung)
$\Rightarrow$ zwischen C_1 und C_2 existieren zwei disjunkte Pfade $(v_x, \ldots v_{z-1}, v_z)$ und $(o_x, \ldots o_{y-1}, o_y)$
$\Rightarrow$ nach Graphenstruktur folgt aus dem entstehenden Kreis $(v_x, \ldots, v_{z-1}, v_z, o_{y-1}, \ldots, o_x) \rightsquigarrow C_1 = C_2 \lightning$

□

Kürzen von Routing–Pfaden

Im Beweis hat sich bereits ein weiterer sinnvoller Abkürzungsalgorithmus angedeutet: Analog zum unabhängig entwickelten Verfahren in [For08] können Discovery-Anfragen detektiert werden, die in Bezug auf das Transportnetz über Schleifen weitergeleitet

werden, und eine Kürzung des später genutzten Pfades erfolgen. Die Fälle 1.2 und 1.3 des Umlenkungsalgorithmus werden dann direkt ausgeschlossen.

Direkter Beziehungsaufbau in gleichen Netzen

Eine unter Umständen langwierige Optimierung der Pfade kann ferner abgekürzt werden, indem die äußeren IP-Adressen der Kommunikationsendpunkte auf eine mögliche direkte Konnektivität untersucht werden. Verfügen beide VPN-Knoten über öffentliche IP-Adressen oder sind Teil desselben Teilnetzes, kann die Möglichkeit einer direkten Kommunikation ohne aufwendige Zwischenschritte überprüft werden.

Routing–Optimierung bei gestörter Kommunikation

Können einige VPN-Knoten nicht direkt miteinander kommunizieren, beispielsweise aufgrund von Transportnetzausfällen oder nicht traversierbaren NAT-Gateways, führt eine reine Optimierung auf Anzahl der Overlay-Hops nicht zwangsläufig zu guten Ergebnissen. Es müssen stattdessen möglichst gute VPN-Knoten gefunden werden, welche die Weiterleitung für die betroffenen Systeme übernehmen. Eine mögliche Metrik für die Bewertung der Güte ist dabei primär die erwartete Verzögerung, aber auch die freie Bandbreite und Anzahl der Sicherheitsbeziehungen des potentiell weiterleitenden Knotens, um Überlastsituationen zu vermeiden. Im Beispiel von Abbildung 4.9 bedeutet dies, dass die gestörte Kommunikation zwischen VPN-Knoten 1 und 3 unter Umständen besser über Knoten 5 als durch Knoten 2 bewerkstelligt wird.

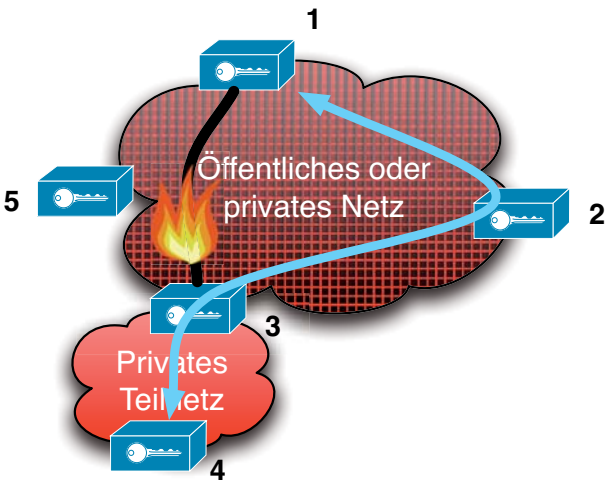


Abbildung 4.9: Beispiel einer Optimierung basierend auf Verzögerungszeiten und freier Bandbreite

Unter der Voraussetzung, dass Knoten 5 und Knoten 2 eine gemeinsame Sicherheitsbeziehung besitzen, ist eine Optimierung relativ einfach zu bewerkstelligen. Die Initiative geht dabei von Knoten 2 aus, der aus seiner eigenen Nachbarschaft beispielsweise

durch die Verwendung eines Netzwerkkoordinatensystems einen geeigneteren Kandidaten für die Weiterleitung identifiziert. Stimmt Knoten 5 einer Weiterleitung zu, baut er gegebenenfalls Sicherheitsbeziehungen zu Knoten 1 und 3 auf, legt einen Routing-Eintrag an und bestätigt die Bereitschaft an Knoten 2. Knoten 2 initiiert daraufhin nach Prüfung der Antwort eine Umleitung des Verkehrs in Knoten 1.

Die Güte der so erzielten Optimierung hängt von zwei Faktoren ab: Zum einen muss die Topologiekontrolle der weiterleitenden Knoten Verbindungen zu möglicherweise besser geeigneten anderen Knoten aufbauen, um so zumindest nach mehreren Optimierungsschritten den optimalen Weiterleitungspunkt zu identifizieren. Zum anderen können durch den Algorithmus nur Pfade optimal bestimmt werden, bei denen genau ein Zwischenknoten existiert. Sind mehr Schritte notwendig, wird unter Umständen ein lokales Minimum gefunden, welches sich jedoch ohne globale Sicht der Knoten nicht mit Sicherheit auflösen lässt. Daher wird zur Gewährleistung der Skalierbarkeitseigenschaften in solchen unwahrscheinlichen Fällen auf Maßnahmen verzichtet, die über die lokale Optimierung hinausgehen.

4.5.4 Mechanismen zur Paketweiterleitung

Einen weiteren Einfluss auf das Routing hat der Weiterleitungsmechanismus. Da von einem Longest-Prefix-Match-Algorithmus aufgrund der nicht topologiegerechten Vergabe von Adressen keine hohe Effizienz zu erwarten ist, sind im Fall von VPN theoretisch zwei verschiedene Arten der Weiterleitung denkbar: Zum einen können Pfade durch das VPN geschaltet werden, und zum anderen Mechanismen mit Ähnlichkeiten zu Source-Routing verwendet werden.

Verschachtelte VPN–Tunnel

Source-Routing kann in VPN durch den Aufbau verschachtelter Tunnel realisiert werden. Wie in Abbildung 4.10 illustriert, baut dabei die Datenquelle Tunnel so ineinander auf, dass der erste Zwischenknoten die Pakete einmal entschlüsselt und anschließend zum nächsten Zwischenknoten im korrespondierenden Tunnel weiterleitet. Alle weiteren Zwischenknoten entfernen die jeweils äußeren Tunnel und den Tunnel vom Quellknoten. Am Zielknoten wird dann der letzte Tunnel entfernt.

Offensichtlicher Vorteil des Verfahrens ist die Sicherung gegenüber potentiell kompromittierten Zwischenknoten. So ist in der Beispielskizze für Zwischenknoten 1 das Ziel weder zu erkennen, noch ist es möglich Zwischenknoten 2 zu umgehen. Würde zusätzlich beim Entschlüsseln überprüft werden, von welchem anderen Zwischenknoten ein Paket kommen darf, kann durch einzelne kompromittierte Knoten kein anderer Weiterleitungspfad gewählt werden. In VPN mit mehreren kompromittierten Knoten können Pakete in diesem Fall lediglich unterschiedlich zwischen diesen umgeleitet werden.

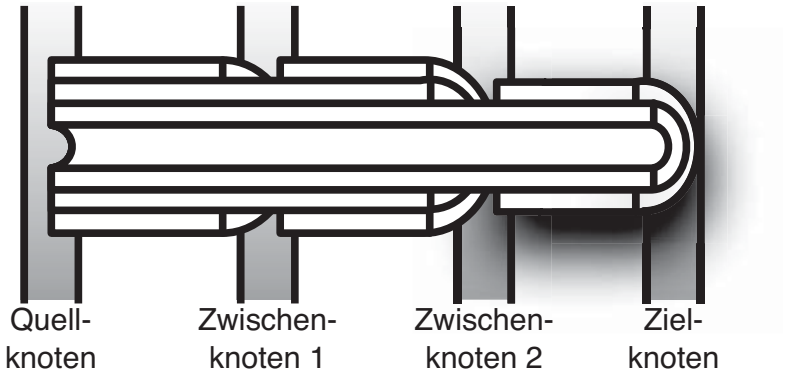


Abbildung 4.10: Beispiel für die Realisierung von Source-Routing mithilfe verschachtelter VPN-Tunnel

Das starre System stellt allerdings einen Nachteil in dynamischen Netzen dar, in diesem Fall ist es beispielsweise einem Zwischenknoten nicht möglich, den folgenden ausgefallenen Zwischenknoten durch einen anderen zu ersetzen. Beim Einsatz des Umlenkungsalgorithmus zur Abkürzung von Pfaden muss in jedem Fall der Quellknoten informiert werden. Darüber hinaus kann der Übermittlungsaufwand stark wachsen, da beispielsweise beim Einsatz von IPsec-IPv4-Tunneln mit AES-CBC-Verschlüsselung und HMAC-SHA-1-96 Authentisierung für jedes Zwischensystem ein Header von bis zu 75 Bytes hinzugefügt wird (siehe [Man07]). Durch die Verschachtelungsstrategie kann sich für die Endsysteme auch die Maximum Transmission Unit (MTU) durch Pfadwechsel sprunghaft ändern, sodass sie entweder nicht ausgeschöpft wird, oder die Pakete stärker fragmentiert werden. Ferner sinkt zwar die Beobachtbarkeit durch interne Angreifer, aufgrund der sich verändernden Paketlängen können externe Angreifer aber allein durch passive Beobachtung Weiterleitungspfade einfacher rekonstruieren. Dazu müssen sie lediglich eingehende Pakete einer bestimmten Länge mit ausgehenden Paketen korrelieren, die genau um einen Protokollkopf kleiner sind. Verhindert werden kann diese potentielle Korrelation nur durch ein Padding gleicher Länge. Hierbei steigt der Übertragungsaufwand aber noch weiter.

Einfache Verschachtelung und Aufbau virtueller Pfade

Eine Alternative stellt die Weiterleitung über virtuelle Pfade dar, bei denen Zwischensysteme auch Routing-Einträge für die über sie lediglich indirekt erreichten Zielknoten vorhalten. In diesem Fall müssen zwischen direkt kommunizierenden VPN-Knoten Tunnel eingerichtet werden und, wie in Abbildung 4.11 dargestellt, wird zusätzlich eine Ende-zu-Ende-Sicherung durchgeführt.

Im Vergleich zum Verfahren mit verschachtelten Tunneln ist es internen Knoten hier möglich, bessere Verkehrsflussanalysen durchzuführen, da aus jedem Paket die Zieladresse hervorgeht. Gleichzeitig wird eine Verkehrsflussanalyse für externe Angreifer erschwert, da der Verkehr nicht durch die Paketgrößen vom indirekt übermittelteten Verkehr anderer Systeme zu unterscheiden ist. Im Gegensatz zum vorherigen Verfahren

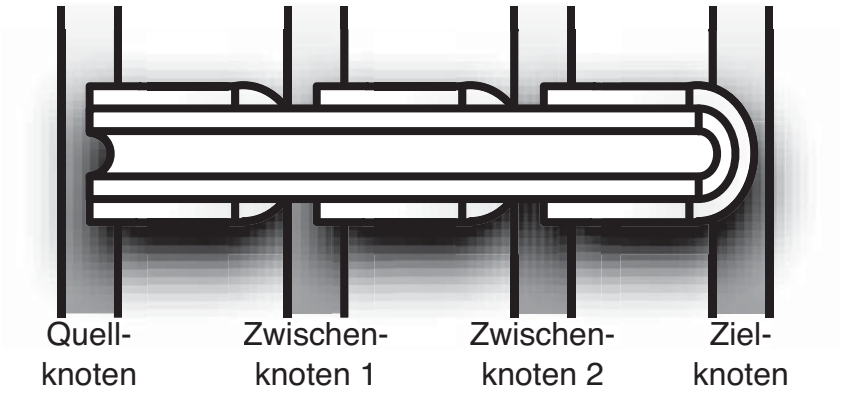


Abbildung 4.11: Beispiel einer Übertragung mithilfe virtueller Pfade

ist es hier Zwischenknoten allerdings möglich, Daten über andere Pfade zum Ziel zu leiten, und so potentiell Sicherheitspolitiken zu umgehen. Durch die obligatorische Ende-zu-Ende-Sicherung führt dies aber zu keiner Verletzung der üblichen Sicherheitsziele, wie Vertraulichkeit, Authentizität und Integrität der Daten.

Da das pfadbasierte System darüber hinaus aufgrund einer kleineren Anzahl von Sicherheitsbeziehungen und dem Versand von weniger Verwaltungsinformationen starke Effizienzvorteile aufweist, überwiegen dessen Vorteile deutlich.

4.5.5 Fazit

Zusammenfassend wird somit auch das Routing in SOLID lediglich auf der Basis lokaler Informationen durchgeführt. Dabei wird initial der durch den Discovery-Mechanismus ermittelte Pfad verwendet, welcher anschließend durch verschiedene Optimierungsmaßnahmen sukzessive verkürzt wird. Wie effizient diese Verfahren funktionieren, hängt dabei primär von zwei Faktoren ab: Zum einen wie gut es gelingt die Zyklen in der direkten VPN-Struktur aufzulösen, und zum anderen inwiefern optimale Weiterleitungspunkte bei partiellen Kommunikationsstörungen gefunden werden. Beide Kriterien bedürfen daher ebenfalls einer quantitativen Evaluation.

Die eigentliche Weiterleitung gestaltet sich hingegen vergleichsweise einfach. Aufgrund der Vorteile des pfadbasierten Systems werden Pakete in SOLID maximal zweimal verschlüsselt: einmal Hop-by-Hop und einmal Ende-zu-Ende. Anschließend erfolgt eine Weiterleitung über virtueller Pfade zum Zielknoten.

4.6 Topologiekontrolle

Die Aufgabe der Topologiekontrolle ist der proaktive Aufbau relevanter Sicherheitsbeziehungen, sowie deren Aufrechterhaltung beziehungsweise Adaption an neue Gegebenheiten in Transportnetz und VPN. Beim Einsatz eines strukturierten Overlay-Netzes zur Realisierung des VPN müssen analog zu den Ansätzen aus Teilabschnitt 3.4.3 in erster Linie die für die Overlay-Struktur wichtigen Beziehungen

aufgebaut werden. Im Fall von SOLID also die Verbindungen zur Realisierung der Ringstruktur.

Der Aufbau der Ringverbindungen verläuft dabei nach folgendem einfachen Algorithmus: Sobald nach einem Start eines Knotens durch den Bootstrapping-Prozess relevante andere Knoten identifiziert werden, erfolgt ein Aufbau Sicherheitsbeziehungen zu diesen. Anschließend wird über diese ersten Sicherheitsbeziehungen eine Suche im Overlay initiiert, um gezielt nach Ringnachbarn zu suchen. Sobald Antwortpakete eingetroffen sind, erfolgt der Aufbau direkter oder indirekter Sicherheitsbeziehungen zu den identifizierten Knoten, und der neueingefügte Knoten kann innerhalb des VPN gefunden werden.

Anschließend wird der Aufbau weiterer Sicherheitsbeziehungen induziert, welcher zum einen je nach Bedarf durch Nutzdaten ausgelöst wird. Zum anderen erfolgt der proaktive Aufbau der Querverbindungen, der Beziehungen des Umlenkungsalgorithmus und der auf Basis des Vivaldi-Koordinatensystems initiierten Kontakte.

Bei der Konstruktion des Overlays wird jedoch auch der Aufbau weiterer Beziehungen aufgrund netztopologischer Gegebenheiten nötig, auf die im Folgenden eingegangen wird.

4.6.1 Divergenz zwischen Transportnetz– und VPN–Routing

Werden herkömmliche Routing-Tabellen zur Paketweiterleitung verwendet, kann es aufgrund unterschiedlicher Reihenfolgen beim Aufbau von Sicherheitsbeziehungen zu Instabilitäten kommen. Ein möglicher Fall ist in Abbildung 4.12 dargestellt.

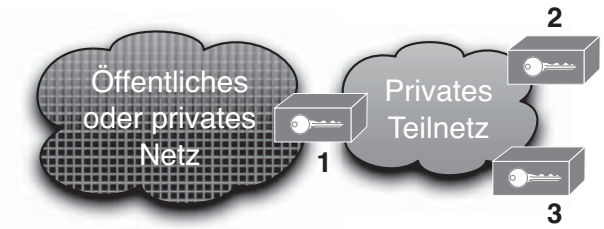


Abbildung 4.12: Sonderfall für die Topologiekontrolle: Aufbau indirekter Beziehungen zwischen Gateway 1 und 3 ist nicht über Knoten 2 möglich

Kommt es zu einem Ausfall des für das private Teilnetz verantwortlichen Gateways 1, werden Knoten 2 und 3 vom Rest des Netzes getrennt und besitzen lediglich eine direkte Sicherheitsbeziehung zueinander. Wird nun Gateway 1 ersetzt und der Bootstrapping-Mechanismus ausgelöst, baut einer der beiden nachgelagerten Knoten eine direkte Beziehung zu Gateway 1 auf. Löst nun der Umlenkungs- oder Ringkonstruktionsalgorithmus aus, baut auch der zweite nachgelagerte Knoten eine indirekte Verbindung zu Gateway 1 auf. Problematisch daran ist, dass Gateway 1 für das private Teilnetz verantwortlich ist, und somit einen Routing-Eintrag für das rote Netz über ihn angelegt werden muss. Das Gateway ist aber lediglich über einen Knoten

des gleichen Netzes zu erreichen, sodass nach erfolgtem Aufbau einer indirekten Sicherheitsbeziehung aufgrund der Routing-Schleife keine weitere Kommunikation stattfinden kann.

Auf Anlegen des Routing-Eintrags kann aber auch nicht einfach verzichtet werden, da vor einem Aufbau einer Sicherheitsbeziehung nicht klar ist, ob es sich beim inneren Netz von Gateway 1 und dem äußeren Netz des nachgelagerten Knoten um dasselbe Netz handelt. Eine Aufgabe der Topologiekontrolle ist es daher, Situationen in denen die Aushandlung indirekter Sicherheitsbeziehungen zu Routing-Divergenzen führt, zu erkennen und durch den proaktiven Aufbau direkter Beziehungen zu unterbinden.

4.6.2 Abhören von Kontrollinformationen zur Beschleunigung der Routing–Konvergenz

Der gleichzeitige Aufbau sehr vieler Sicherheitsbeziehungen, wie etwa nach dem Neustart eines zentralen Transportnetz-Routers oder nach einem Stromausfall, führt unter Umständen zu inkonsistenten Overlay-Strukturen, da die verschiedenen virtuellen Pfade zur Weiterleitung zu einem VPN-Knoten unter Umständen mittels sich überschneidender Overlay-Pfade aufgebaut werden.

Abbildung 4.13 zeigt eine Topologie, in der das Einfügen von Knoten 1 zur Bildung einer Routing-Schleife führt. Versendet der Knoten gleichzeitig Nachrichten zum Finden seines rechten und linken Ringnachbarn, und legt jede der Nachrichten pro Zeiteinheit einen Overlay-Schritt zurück, so überholen sich die Nachrichten in Knoten 2 und 4. Dadurch entsteht eine Routing-Schleife, da Knoten 2 Knoten 1 über den Pfad 2→3→4→5→1 zu erreichen versucht, jedoch Knoten 4 inzwischen zu Knoten 1 über den Pfad 4→7→2→6→1 weiterzuleiten versucht. Die Schleife könnte erst nach Zeitüberschreitung durch eine erneute Übertragung der Suchanfragen beseitigt werden.

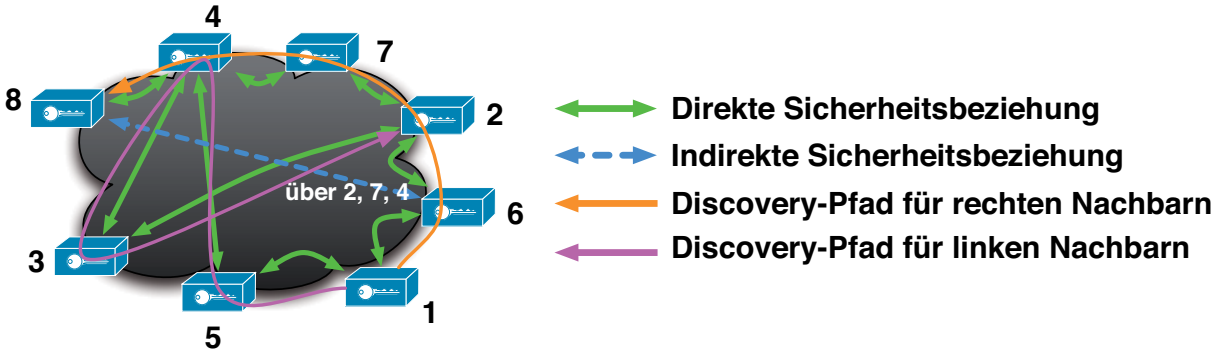


Abbildung 4.13: Beispieltopologie für die Bildung von Routing-Schleifen

Mögliche Lösungen für die Inkonsistenzen umfassen:

- **Verzögerte Suchanfragen:** Durch einen kontrollierten langsameren Aufbau von Verbindungen können Inkonsistenzen, die durch den gleichen Host ausgelöst werden, vermieden werden. Allerdings führt dies zum einen zu einer stark verminderten Effizienz, da keine Anfragen parallel bearbeitet werden können, und im schlimmsten Fall auf eine Zeitüberschreitung gewartet werden muss. Zum anderen müssen unter Umständen auch eingehende Anfragen gedrosselt werden, weil durch die Beantwortung von Suchanfragen Routing-Schleifen entstehen können.

- **Flapping-Route-Protection:** Als weitere Lösungsmöglichkeit kann das Konzept der Flapping-Route-Protection aus herkömmlichen Netzen übernommen werden, wobei innerhalb eines festgelegten Zeitintervalls im Sekundenbereich keine Routen aktualisiert werden. Alternativ ist es möglich innerhalb des Zeitintervalls nur Aktualisierungen mit echt kleinerer Anzahl an Overlay-Schritten zu akzeptieren. Allerdings schwankt der optimale Zeitraum für den Schutz aufgrund der kryptographischen Verzögerungen im VPN gerade bei starken Netzfluktuationen stark, sodass große Werte eine Konvergenz bei Ausfällen verzögern und kleine Werte nicht ausreichen.

- **Auswertung weitergeleiteter Nachrichten:** Die für die Overlay-Konvergenz wichtigsten Verbindungen sind in SOLID die Ringverbindungen. Da es sich bei diesen um symmetrische Beziehungen handelt, muss nicht auf die Anfrage des neu eingefügten Knotens gewartet werden. Stattdessen kann jeder VPN-Knoten bei jeder von ihm weitergeleiteten Nachricht prüfen, ob ihr Ursprungsknoten für die eigene Topologiekontrolle interessant ist. In diesem der Fall, wird eine Verbindung proaktiv über den Knoten aufgebaut, von dem die Nachricht erhalten wurde. Neben der Vermeidung von vielen Fällen bei denen Inkonsistenzen entstehen, werden auch gleiche Pfade mehrfach verwendet und die Reaktionsgeschwindigkeit des Netzes insgesamt erhöht.

Neben einem vergleichsweise kurzen Schutz vor dem Überschreiben von Routen, werden in SOLID daher der Topologiekontrolle Informationen über weitergeleitete Nachrichten bereitgestellt.

4.6.3 Overlay–Wiederherstellung nach Netzpartitionierungen

Da strukturierte Overlay-Netze per Definition nur bestimmte Verbindungen aufbauen, und die verwendeten Konstruktionsalgorithmen auf das sequentielle Einfügen von Knoten abgestimmt sind, besteht die theoretische Möglichkeit, dass es nach Ausfällen in einen stabilen, aber inkonsistenten Zustand gerät. Ähnliche Situationen sind nicht nur durch den Ausfall einzelner VPN-Gateways oder Transportnetzfehler zu erwarten, sondern auch beim gleichzeitigen Start einer großen Anzahl von VPN-Knoten, wie beispielsweise nach einem Stromausfall. Auch hier entstehen in verschachtelten Topologien durch unterschiedliche Startzeiten und Netzwerkverzögerungen zunächst autonome Teilstrukturen, die sich anschließend verbinden müssen.

Ein Beispiel einer besonders anfälligen Netzwerktopologie ist in Abbildung 4.14 gezeigt, bei der ein zentraler VPN-Knoten die Vermittlung zwischen zwei Teilnetzen übernimmt, er sich aber auf Basis seines Identifikators lediglich zu einem Teilnetz verbinden muss. Fällt der Knoten aus, bilden sich, wie in Abbildung 4.14 illustriert, zwei getrennte Overlay-Strukturen aus, die unabhängig voneinander operieren können. Wird der zentrale Knoten 5 wieder reaktiviert, verbinden sich die Teilstrukturen aber möglicherweise nicht wieder zu einem Ring. Stattdessen wird sich der Knoten wieder in den Ring des rechten Netzwerks integrieren, weil die Knoten ihm aufgrund der kürzeren Verzögerungszeit schneller antworten. Lernt der Knoten andere Knoten aus dem öffentlichen Netz durch Bootstrapping kennen, ist die Chance insbesondere bei großen VPN relativ hoch, dass er nur Knoten findet, für die er nicht relevant ist. In diesem Fall bleiben zwei Ringe bestehen, die lediglich über einige wenige Querverbindungen miteinander verbunden sind, da Knoten 2 und 3 beziehungsweise 7 und 9 einander nicht mittels Overlay-Suchen finden.

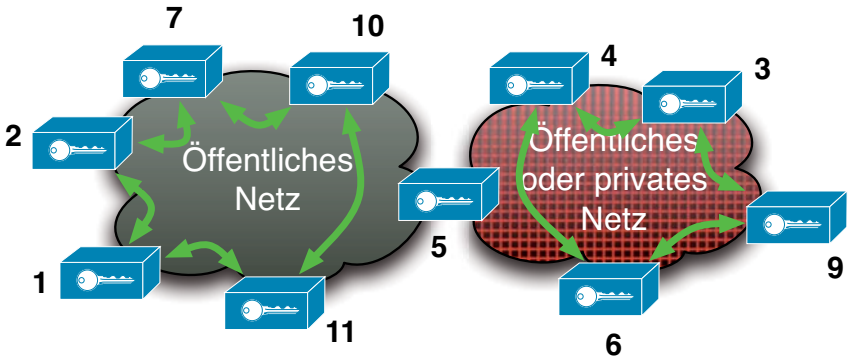


Abbildung 4.14: Beispiel einer Netztopologie bei der ein Ausfall eines zentralen Knotens eine permanente Overlay-Partitionierung bewirken kann

Da eine topologiegerechte Vergabe von Identifikatoren, bei der Übergangsknoten immer in allen Teilen des Transportnetzes Nachbarn haben, aufgrund potentieller Knotenmobilität und mangelnder Flexibilität auszuschließen ist, werden in den folgenden Abschnitten Maßnahmen zur konsistenten Zusammenführung der Teilstrukturen diskutiert.

Proaktiver Aufbau zufälliger VPN–Verbindungen

Um die partitionierten Overlay-Netze überhaupt zu verbinden, müssen die sonst eigentlich nur beim Einfügen von Knoten benötigten Bootstrapping-Mechanismen in regelmäßigen Abständen ausgeführt werden. In vielen kurzzeitigen Ausfallszenarien ist insbesondere das Kontaktieren der IP-Adressen vormals gültiger VPN-Knoten sinnvoll, deren Verbindungen aufgrund von Zeitüberschreitungen abgebaut wurden. Zu den erhaltenen äußeren IP-Adressen müssen zufällig Sicherheitsbeziehungen aufgebaut werden, damit sichergestellt wird, dass Teilnetze überhaupt zusammenwachsen können.

Fallen VPN- und Transportnetzgrenzen aufeinander (wie im Beispielnetz der Abbildung 4.14), können zum einen sehr einfach Partitionierungen entstehen. Zum anderen können die Teilnetze in diesem speziellen Fall aber relativ schnell wieder verbunden werden, indem jeder weiterleitende Knoten für jede seiner Netzwerkschnittstellen versucht, eine geringe Mindestanzahl von Sicherheitsbeziehungen aufzubauen. Nach dem Neustart des zentralen Knotens ergibt sich somit de facto sofort ein zusammenhängendes VPN, jedoch nicht notwendigerweise eine konsistente Overlay-Struktur.

Pfadannoncierung repräsentativer Knoten

In VRR [CCN+06] wird zur Behandlung von Overlay-Partitionierungen im Kontext von MANET vorgeschlagen, dass die kürzesten Pfade zu einem *repräsentativen* Knoten jedes Teilnetzes an alle Teilnehmer annonciert werden. Die Autoren von VRR haben dazu den Knoten mit dem kleinsten Knotenidentifikator gewählt. Kommt es zu einer Vereinigung von Overlay-Netzen lernen diese Knoten einander kennen und dienen als eine Art „Kristallisationskeime“ für die neu auszubildende Struktur.

Aus Sicherheitssicht ist die Wahl von Koordinatoren jedoch potentiell problematisch, da sie zusätzliche exponierte Punkte darstellen. Kann ein Angreifer den Aufbau von Verbindungen zwischen den gewählten Knoten beispielsweise durch Grayhole-Angriffe verhindern, ist die Konvergenz nicht länger gesichert.

Suche nach Ringnachbarn pro Netzwerkschnittstelle

Eine die Verfügbarkeit erhöhende Variante für das Verbinden von VPN-Teilen, die durch den temporären Ausfall eines zentralen Gateways getrennt waren, stellt die regelmäßige, getrennte Suche nach Ringnachbarn über jede einzelne Netzwerkschnittstelle dar. Sucht Knoten 5 aus dem Beispiel in Abbildung 4.14 in beiden Netzen nach Nachbarn, so findet er sie zwar nur im rechten Teilnetz, die Knoten 2 und 7 erkennen aber anhand der Suchanfragen, dass auf die Overlay-Struktur bezogen Knoten 5 zwischen sie gehört. Daher werden die Verbindungen zwischen 2 und 5 sowie 7 und 5 aufgebaut und im weiteren Verlauf die optimalen Nachbarn im nachgelagerten Netz gefunden.

Auch hier stellen zwar die Knoten 2 und 7 zusätzliche exponierte Punkte dar, sie sind aber nicht zwangsläufig global eindeutig; ein Angreifer kann also nicht durch die Kontrolle eines oder weniger Knoten ein gesamtes VPN am Konvergieren hindern. Ähnlich wie beim Aufbau eines zusammenhängenden VPN durch das Bootstrapping einzelner Netzwerkschnittstellen, ist das Verfahren allerdings auf Szenarien beschränkt, in denen Netzübergänge durch VPN-Gateways erfolgen. Würde ein zentraler Router oder Switch an dieser Stelle ausfallen, kann keine Konvergenz garantiert werden.

Stochastische Suche nach Ringnachbarn

Um Overlay-Strukturen auch in Situationen konvergieren zu lassen, bei denen die Partitionierung allein auf das Transportnetz beziehungsweise auf einen weiterleitenden VPN-Knoten mit nur einer Netzwerkschnittstelle zurückzuführen ist, werden in SOLID zusätzliche stochastische Suchen nach Ringnachbarn durchgeführt. Ähnlich wie im Fall des zentralen VPN-Gateways wäre es dabei theoretisch ausreichend in regelmäßigen Abständen einen zufälligen Nachbarknoten mit einer Suche nach Ringnachbarn zu beauftragen. Allerdings wird aus Verfügbarkeitsgründen durch eine konstante Anzahl vorhergehender weiterer randomisierter Routing-Schritte sichergestellt, dass durch den partiellen Ausfall einiger Knoten die Konvergenz nicht verhindert werden kann. Wird also ein Grayhole-Angriff auf Knoten 2 und 7 aus dem Beispiel in Abbildung 4.14 durchgeführt, können Knoten 9 und 10 einander langfristig durch den Algorithmus trotzdem finden.

Wie häufig die randomisierte Suche durchzuführen ist, und wie viele Routing-Schritte notwendig werden, hängt dabei von der VPN- und Transportnetztopologie ab und bedarf einer genaueren quantitativen Untersuchung.

4.6.4 Anpassung an die Transportnetzumgebung

Wie bereits bei der Vorstellung der Latenzoptimierung in Abschnitt 4.5.3 angedeutet, benötigen Knoten ein möglichst gutes Abbild ihrer Umgebung, um effektiv einschätzen zu können, inwiefern sich eine Delegation von Weiterleitungsaufgaben auf andere VPN-Knoten lohnen kann. Da eine der wichtigsten Metriken in diesem Fall die Minimierung der Gesamtlatenz ist, besteht die Hauptaufgabe darin, diese zwischen anderen Knotenpaaren abschätzen zu können.

Eine relativ naive Lösung besteht darin, die Kandidatenknoten mit Latenzmessungen zu beauftragen und gegebenenfalls die Weiterleitung auf Basis dieser Ergebnisse zu reorganisieren. Allerdings ist der Aufwand solcher Messungen nicht zu vernachlässigen. Besitzt jeder der Knoten eines VPN $\mathcal{O}(\log n)$ Nachbarknoten, fallen insgesamt $\mathcal{O}(n \log^2 n)$ Messungen an, da jeder Knoten jeden seiner Nachbarn zumindest beauftragen muss, die Latenzen zu den anderen Nachbarknoten zu bestimmten. Zudem sind für eine Anpassung an neue Transportnetzgegebenheiten die Messungen periodisch zu wiederholen. Bei einer schrittweisen Latenzoptimierung auf Basis lokalen Wissens ist es außerdem wichtig, die Nachbarknoten so zu wählen, dass durch die Optimierung nicht nur ein suboptimales, lokales Minimum erreicht wird. Ein reiner Greedy-Ansatz, der Verbindungen zu Knoten mit niedriger Latenz aufbaut, führt beispielsweise nicht zu zwangsläufig zu guten Ergebnissen. Unter der Annahme, dass Latenzen die euklidischen Distanzen zwischen Knoten repräsentieren, zeigt Abbildung 4.15 ein Negativbeispiel. Die Knoten 1 und 6 können den optimalen Weiterleitungspunkt 4 nicht ermitteln, obwohl Sicherheitsbeziehungen zu allen bezogen auf die Latenz dichten Knoten aufgebaut sind. Zusätzlich müssten die Sicherheitsbeziehungen 1-4 und 4-6 aufgebaut werden.

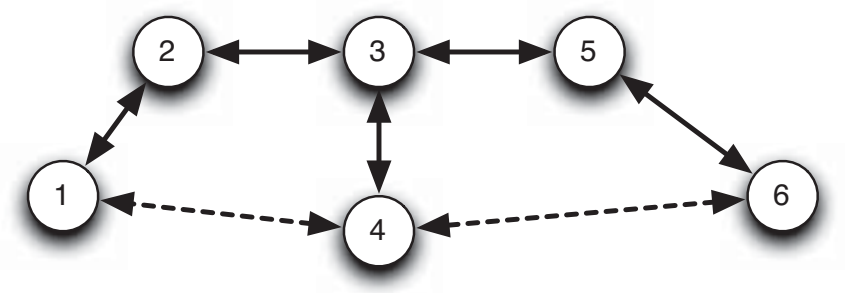


Abbildung 4.15: Beispiel für lokale Minima bei der Latenzoptimierung

Aus diesem Grund wurden Netzwerkkoordinatensysteme und im Speziellen die bereits in Teilabschnitt 2.4.4 vorgestellte, vollständig verteilte Variante Vivaldi entwickelt. Diese erlaubt eine Berechnung künstlicher Koordinaten auf Basis von per Piggybacking betriebenen Latenzmessungen. Allerdings lässt sich eine Auswahl von Nachbarknoten zur optimalen Aufteilung des Koordinatenraumes, wie sie in euklidischen Räumen durch Delaunay-Triangulation realisiert wird, in Vivaldi nicht einfach umsetzen. Grund hierfür ist die zusätzlich eingeführte Manhattan-Distanz für die Höhenkoordinate. Beispielsweise müssen Knoten, die über eine geringe Distanz in der x-y-Ebene verfügen, nicht benachbart sein, wenn sie durch hohe Latenzen der Zugangsnetze eine große Höhe besitzen.

Der in SOLID verfolgte Ansatz nimmt daher keine global eindeutige Aufteilung der Knoten vor sondern berechnet für jeden Knoten individuell seine Umgebung mittels einer modifizierten Delaunay-Triangulation. Wie in Abbildung 4.16 dargestellt, werden dazu die Höhen der einzelnen Knoten zurück in den euklidischen Raum transformiert, indem die x-y-Positionen um die Höhe vom berechnenden Knoten weg bewegt werden. Knoten mit großer Höhe werden, wie in der Abbildung ebenfalls dargestellt, dabei nicht als Nachbarn ausgewählt und nicht für eine Weiterleitung in Erwägung gezogen. Da es sich nicht länger um eine symmetrische Beziehung handelt, und die eigene Höhe nicht in die Berechnung eingeht, können Knoten mit hoher Latenz trotzdem Nachbarn wählen und so an der Optimierung teilnehmen.

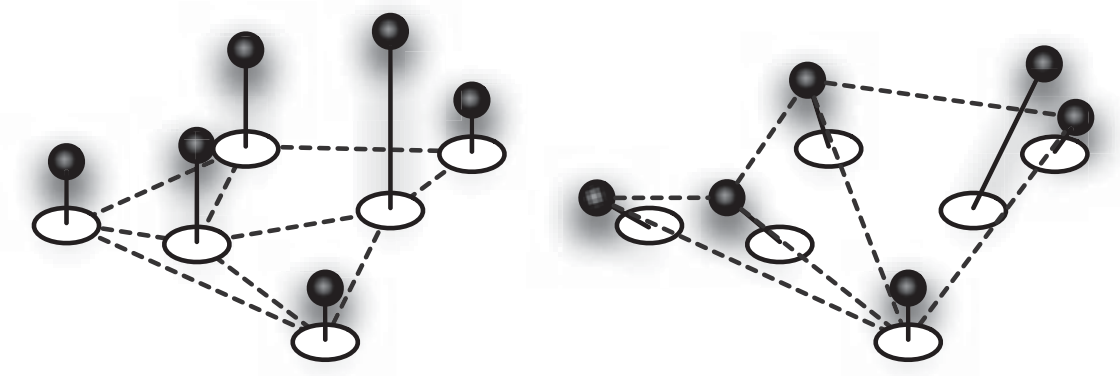


Abbildung 4.16: Anpassung der Delaunay-Triangulation auf eine Kombination aus euklidischer und Manhattan-Distanz

Zusätzlich eröffnet sich die Möglichkeit durch eine künstliche Anpassung der Höhe weitere Optimierungsfaktoren, wie die Auslastung eines Knotens, transparent in die Berechnung einfließen zu lassen. Auf diese Weise werden Knoten, welche über niedrige Kapazitäten zur Weiterleitung verfügen, durch eine größere Höhe weniger häufig als Nachbarn gewählt, ohne dass die Gefahr besteht, ein lediglich lokales Minimum zu erreichen.

4.6.5 Fazit

Insgesamt hat die Topologiekontrolle die komplexe Aufgabe, proaktiv Verbindungen aufzubauen, die für eine effektive Suche in der Overlay-Struktur notwendig sind. Zusätzlich ist es erforderlich Verbindungen herzustellen, um Inkonsistenzen durch netztopologische Aspekte und Partitionierungen zu vermeiden. Der Aufbau einer weiteren, auf einer modifizierten Delaunay-Triangulation beruhenden Struktur ermöglicht Latenzoptimierungen, welche auch bei der Weiterleitung von Multicast-Verkehr im folgenden Abschnitt eine Rolle spielen.

4.7 Multicast–Routing

Da die durch SOLID aufgebauten Sicherheitsbeziehungen über individuelle Schlüssel verfügen, kann keine direkte Behandlung von Multicast-Verkehr innerhalb des VPN erfolgen. Stattdessen wird durch ein gesondertes Subsystem [Hei09, HRS10] ein Multicast-Dienst bereitgestellt, der auf Prinzipien des ALM aufbaut, um auch in Transportnetzen ohne eigene Multicast-Unterstützung die Weiterleitung des Verkehrs unter den VPN-Knoten zu verteilen. In den folgenden zwei Abschnitten wird auf die Realisierung von ALM-Topologien in SOLID und deren Optimierung eingegangen.

4.7.1 Topologien für die Verteilung von Multicast–Daten

Für die Verteilung von Daten für Gruppenkommunikation über ALM wird im Allgemeinen zwischen Pull- und Push-basierten Ansätzen unterschieden. Erstere haben den Vorteil, dass sie durch ihre vermaschte Struktur im Prinzip robuster mit Ausfällen umgehen können; während letztere durch ihre effizienten Baumtopologien besser geeignet sind, um mit wenigen Overlay-Verbindungen Daten schnell auszuliefern. Es ist allerdings davon auszugehen, dass in VPN übertragene Multicast-Daten auch verzögerungskritische Datenströme enthalten, wie etwa bei der Videoüberwachung einer Bank. Daher bietet sich zunächst der Aufbau von Baumstrukturen an, welche bei härteren Verfügbarkeitsanforderungen gegebenenfalls durch Redundanzen abgesichert werden müssen.

Einen weiteren großen Einfluss auf die effiziente Verteilung von Multicast-Daten hat die angenommene Anzahl der Quellen im System. Würde beispielsweise jeder

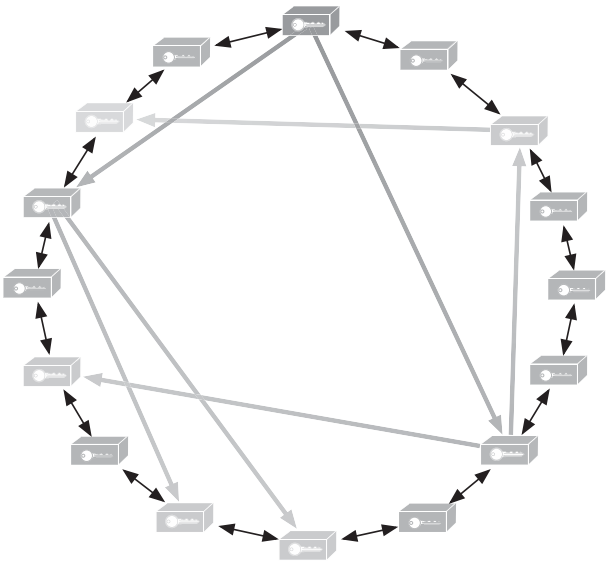


Abbildung 4.17: Einbettung eines ALM-Baumes in eine SOLID Topologie

Knoten über DNS-SD (vormals Bonjour beziehungsweise Rendezvous [CK10]) seine Dienste innerhalb des VPN annoncieren, so skalieren Topologien, die pro Quelle einen unabhängigen Verteilbaum aufbauen, möglicherweise nicht ausreichend. Jedoch ist der Gewinn eines ALM-Ansatzes in solchen Szenarien de facto nicht vorhanden, da dies auf eine paarweise Kommunikation zwischen allen Teilnehmern hinausläuft. Außerdem ist die Gefahr im VPN vergleichsweise hoch, dass Multicast-Mechanismen zur Verstärkung von bandbreitenerschöpfenden Angriffen eingesetzt werden, wenn alle Teilnehmer Multicast-Pakete versenden können. Daher wird im Folgenden von einem Szenario mit einer oder wenigen Quellen pro Multicast-Gruppe ausgegangen.

Es ergibt sich eine Overlay-Struktur wie sie in Abbildung 4.17 beispielhaft dargestellt ist. Die VPN-Knoten bauen unabhängig von ihrer Ringposition pro Gruppe einen eigenen ALM-Baum auf, bei dem sich die jeweilige Quelle an der Wurzel befindet. Durch die Verwendung des SOLID-Systems für Routing und Datenweiterleitung spielt die direkte Erreichbarkeit in dieser Topologie keine Rolle. Bei partiellen Ausfällen oder verschachtelten VPN-Topologien können daher auch indirekte Verbindungen zur Weiterleitung von Multicast-Inhalten verwendet werden. Alle vorgestellten Optimierungsmaßnahmen werden ebenfalls transparent angewendet.

Um auf einen gesonderten Bootstrap-Dienst zu verzichten, wird für die SOLID-Struktur für das Einfügen neuer Knoten in den ALM-Baum verwendet. Zur Gewährleistung eines effizienten Knoteneinstiegs besteht dabei prinzipiell die Möglichkeit, die Quelle an einem exponierten Punkt im VPN zu registrieren, oder alternativ alle Knoten mit freien Weiterleitungskapazitäten auffindbar zu machen. In beiden Varianten fügen sich die betreffenden Knoten in den SOLID-Ring an der durch die Multicast-Adresse bestimmten Stelle ein und werden über reguläre Overlay-Suchen in logarithmisch vielen Overlay-Schritten erreicht. Für eine Realisierung des Einfügevorganges über die Multicast-Quelle sprechen dabei vor allem drei Gründe: Zum ersten müssen auf diese Weise weniger Sicherheitsbeziehungen als im Alternativansatz aufgebaut werden. Knoten mit freien Kapazitäten müssten sich sonst immer erst durch den Aufbau

zweier Sicherheitsbeziehungen an die entsprechende Stelle in den Ring einfügen, um möglicherweise kurz darauf wieder entfernt zu werden. Zum zweiten können Knoten, die von ihrem Vaterknoten getrennt wurden, beim Einfügen über die Quelle der Gruppe einfach erneut beitreten. Im Alternativansatz besteht die Chance, dass sie einen freien Knoten finden, welcher ein indirekter Nachfolger von ihnen selbst ist, auf diese Weise ihrem eigenen Teilbaum beitreten und so bis zu einer Zeitüberschreitung die Weiterleitung im gesamten Teilbaum blockieren. Drittens erlaubt der über die Quelle einfügende Ansatz durch rekursives Herunterreichen der Anfrage bereits beim Einfügen Optimierungen vorzunehmen, wie sie im folgenden Abschnitt vorgestellt werden.

4.7.2 Optimierungsmaßnahmen

Auch wenn die Verteilung von Multicast-Inhalten durch ein einfaches Einfügen von VPN-Knoten in die Baumstruktur funktioniert, können Effizienz, Skalierbarkeit und Sabotageresistenz durch kontinuierliche Optimierungen verbessert werden. Als topologisches Ziel wurde dabei bereits unter anderem in [Str07, RSS07] begründet, dass die Baumhöhe minimiert werden soll. Dies ist zum einen durch eine möglichst gute Balancierung des Verteilbaumes und zum anderen durch ein Einfügen von Knoten mit hohem Ausgangsgrad in obere Ebenen zu erreichen. Ferner wird das Ziel formuliert, dass Knoten der oberen Ebenen möglichst lange am System teilgenommen und somit ihre Stabilität gezeigt haben sollen. Daher bewertet jeder Multicast-Knoten bei Topologieoptimierungen seine Nachfolger mithilfe folgender Funktion:

$$rating = c_1 \left(1 - \frac{\#succ}{\max \#succ}\right) + c_2 \frac{capacity}{\max capacity} + c_3 \frac{age}{\max age} + c_4 \left(1 - \frac{latency}{\max latency}\right)$$

Dabei stehen #*succ* für die Anzahl der Nachfolger, *capacity* für die freie Weiterleitungskapazität, *age* für die Dauer der Nachfolgerschaft, *latency* für die mit Vivaldi geschätzte Latenz zu einem Zielknoten und $c_{1\ldots4}$ für individuelle Wichtungsfaktoren. Im Einzelnen wird die Bewertungsfunktion in vier Mechanismen verwendet:

- **Einfügen neuer Knoten:** Wie bereits angedeutet, werden neue Knoten im Regelfall über die Quelle eingefügt, welche mithilfe der Bewertungsfunktion einen möglichst guten Teilbaum für die Weiterleitung auswählt und die Nachricht an diesen weiterleitet. Der Vorgang setzt sich rekursiv fort bis ein Knoten mit ausreichend Kapazität gefunden wird. Da SOLID allerdings auch verschachtelte Topologien erlaubt, kann dies zu sehr ineffizienten Baumstrukturen führen, wenn diese von der Transportnetztopologie stark abweichen. Daher können SOLID-Knoten zusätzlich Multicast-Einfügenachrichten abfangen und den Knoten in den eigenen Teilbaum einfügen.

- **Balancierung von Multicast-Bäumen:** Regelmäßig ausgetauschte Nachrichten erlauben es jedem weiterleitenden Knoten die Größe seiner Teilbäume und so den Grad der Balancierung zu bestimmen. Weichen die Größen zu stark ab, werden Blattknoten aus einem Teilbaum in einen anderen eingefügt. Da lediglich

Blattknoten verschoben werden, verbleiben alte Knoten in ihren Positionen und die Stabilität des Multicast-Baumes wird nicht wesentlich verringert.

- **Lastanpassung:** Sollte ein Knoten feststellen, dass er beispielsweise durch das Ausscheiden direkter Nachfolger über freie Bandbreite verfügt, kann er einen vollständigen Teilbaum eines direkten Nachfolgers unter sich einfügen. Der Vorgang setzt sich rekursiv fort und hilft so die Baumhöhe unter Beachtung des Knotenalters zu reduzieren.
- **Positionstausch:** Knoten vergleichen regelmäßig ihren eigenen Wert mit dem ihrer Nachfolger. Stellt ein Knoten trotz der per Definition größeren Anzahl an Nachfolgern und des höheren Alters fest, dass einer seiner Nachfolger besser zur Weiterleitung geeignet ist, kann er einen Positionstausch initiieren und ein Nachfolger von diesem werden. Auf diese Weise können auch ältere Knoten auf lange Sicht von oberen Positionen des Baumes verdrängt und die Hierarchie flacher gestaltet werden.

Eine Vielzahl weiterer Optimierungsmaßnahmen ist denkbar, um Stabilität und Effizienz der Multicast-Verteilung weiter zu erhöhen. Allerdings sind diese nicht zwangsläufig VPN-spezifisch, für einen Machbarkeitsnachweis nicht nötig und daher nicht weiter Fokus dieser Arbeit.

4.8 Fazit

Von relativ einfachen VPN-Modellen und Annahmen ausgehend, wurde in diesem Kapitel ein umfassendes System zur Autokonfiguration von VPN entworfen. Im Einzelnen wurde auf Bootstrapping, Adresszuweisung, Discovery von VPN-Knoten, Routing und Weiterleitung von Datenpaketen sowie auf Topologiekontrolle für Uni- und Multicast-Verkehr eingegangen. Das entworfene System *SOLID* verspricht durch seine verteilte Struktur einen hohen Grad an Sabotageresistenz und durch den Einsatz von Peer-to-Peer-Prinzipien werden darüber hinaus auch Skalierbarkeit und Effizienz berücksichtigt. Hierbei konnte gezeigt werden, dass mithilfe vergleichsweiser kleiner Stichproben Querverbindungen nahezu optimal aufgebaut werden können, und dass so zur Realisierung variabler Subnetzmasken auf Hashing der Identifikatoren verzichtet werden kann. Ferner wurden Mechanismen entwickelt, die auch bei der Einbettung strukturierter Overlays in verschachtelte Transportnetzstrukturen Konvergenz und Wiederanlauf nach Ausfällen sichern.

Allerdings ist das Potential in Bezug auf Sabotageresistenz noch nicht voll ausgeschöpft. Die Topologiekontrolle kann durch den Aufbau besonders stabiler Topologien in der Ebene des direkten VPN die Gesamtstabilität weiter erhöhen. Diese Möglichkeit wird im folgenden Kapitel näher untersucht werden.

All problems in computer science can be solved by another level of indirection.

(David Wheeler)

Kapitel 5

Betrachtungen zur Sabotageresistenz

Durch die eingesetzten verteilten Algorithmen und den Verzicht auf exponierte Punkte haben Sabotageangriffe auf SOLID-VPN lediglich lokale Auswirkungen, sofern keine kritischen Ausfallpunkte auf Anwendungs- oder Transportnetzebene bestehen. Nicht betroffene Netzteile können unabhängig weiter operieren und betroffene Systeme über andere Transportnetzanbindungen schnell reintegriert werden. Allerdings erlaubt das Verfahren keine Vorhersagen über den maximalen Schaden, welcher selbst durch kurzzeitige Angriffe im ungünstigsten Fall ausgelöst werden kann. Daher wird im Folgenden auf analytische Methoden zur Schadensvorhersage und –begrenzung eingegangen, die zum Teil auf Vorarbeiten in [BRS09, Bra10] zurückzuführen sind. Mit einer Beschreibung der auf Basis dieser Betrachtungen entstandenen SOLID-Erweiterungen schließt das Kapitel.

5.1 Modellierung von VPN

Die direkten Sicherheitsbeziehungen eines VPN können im einfachsten Fall als Kanten eines Graphen zwischen den VPN-Knoten aufgefasst werden. Für die Einschätzung bandbreitenerschöpfender DoS-Angriffe, wie sie in Abschnitt 3.1.4 als größte Bedrohung für die Verfügbarkeit herausgearbeitet wurde, reicht der Detailgrad dieser Modellierung jedoch nicht. Diese Art von DoS-Angriffen hat nur auf betroffene Netzwerkschnittstellen einen direkten Einfluss, der Rest des VPN-Knotens und insbesondere mögliche andere Netzanbindungen sind nicht betroffen. Wie in Abbildung 5.1 beispielhaft dargestellt, werden daher VPN-Knoten durch einen Teilgraphen modelliert, in welchem jede Netzwerkschnittstelle durch einen Knoten im Graph repräsentiert wird. Die einzelnen Schnittstellen des VPN-Knotens werden zu einer Clique verbunden. Anschließend ist es möglich, VPN mit bekannten Metriken [BFG+09] wie dem Knoten- oder Kantenzusammenhang zu untersuchen.

Sollen die Auswirkungen interner Angriffe durch Bandbreitenüberlastung analysiert werden, können Sicherheitsbeziehungen zusätzlich mit Kantenkapazitäten annotiert werden. Der maximale Durchsatz der Netzwerkschnittstellen beziehungsweise der beschränkenden „letzten Meile“ kann durch Knotenkapazitäten ausgedrückt werden. Kanten zwischen den Netzwerkschnittstellen werden mit unendlicher Kapazität angenommen.

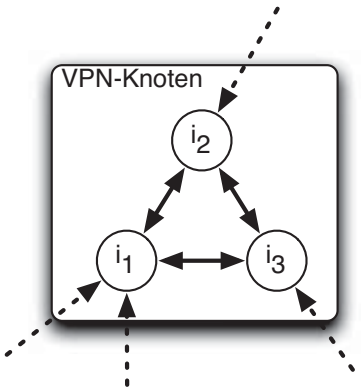


Abbildung 5.1: Beispiel für die Modellierung eines VPN-Knotens mit drei Netzwerkschnittstellen und vier Sicherheitsbeziehungen

Für die realistischere Abschätzung von Bedrohungen müssen weitere Wichtungsfaktoren hinzugezogen werden. So ist es zum Beispiel vergleichsweise einfacher, ein Gerät eines Außendienstmitarbeiters zu kompromittieren oder zumindest dessen Kommunikation mit dem restlichen VPN zu beobachten. Gleichzeitig hat der Ausfall des VPN-Gateways eines zentralen LDAP-Servers schwerere Folgen als die erfolgreiche Sabotage eines einzelnen Heimarbeitsplatzes. Somit müssen die Wahrscheinlichkeit mit der Teile des VPN beobachtet werden oder die Kosten, die beim Angreifer für die Identifikation relevanter VPN-Knoten entstehen sowie gegebenenfalls der „Wert“ einzelner Sicherheitsbeziehungen für den VPN-Betreiber einkalkuliert werden. Die Kosten für den eigentlichen DoS-Angriffsverkehr zu berücksichtigen ist in der gegenwärtigen Bedrohungslage nur beschränkt sinnvoll, da durch gemietete Bot-Netze leicht Verkehr in Größenordnungen mehrerer Gigabit pro Sekunde generiert werden kann [Oll09, Bro10].

Im folgenden Kapitel bezieht sich die Verwendung von Beobachtungswahrscheinlichkeiten auf die Wahrscheinlichkeit mit der ein Angreifer eine erfolgreiche Beobachtung eines Knotens innerhalb eines definierten Zeitintervalls durchführen kann. Dabei wird immer von unabhängigen Ereignissen ausgegangen, da Angreifer beispielsweise physischen Zugang zu den entsprechenden VPN-Knoten benötigen. Gleiches gilt für Kosten, die für das Kompromittieren einzelner Knoten entstehen. Knoten eines physischen Standortes, die der Argumentation nach gemeinsam beobachtet werden können, müssen vor den Berechnungen zu einem virtuellen Knoten mit mehreren Netzwerkschnittstellen zusammengefasst werden.

5.2 Abschätzung der Bedrohung durch externe DoS–Angriffe

Mithilfe des erstellten Netzwerkmodells kann auf relativ einfache Weise vorhergesagt werden, welche Teile des VPN durch die Beobachtung von einem oder mehreren bestimmten Knoten identifiziert werden können. Diese sind dann prinzipiell gefährdet, Opfer von DoS-Angriffen zu werden, und alle betroffenen Netzwerkschnittstellen

sind nicht länger in der Lage Pakete weiterzuleiten. Abbildung 5.2 zeigt ein einfaches Beispiel, in welchem die Beobachtung von einem der vier Knoten für die Vorbereitung eines Angriffs genutzt wird.

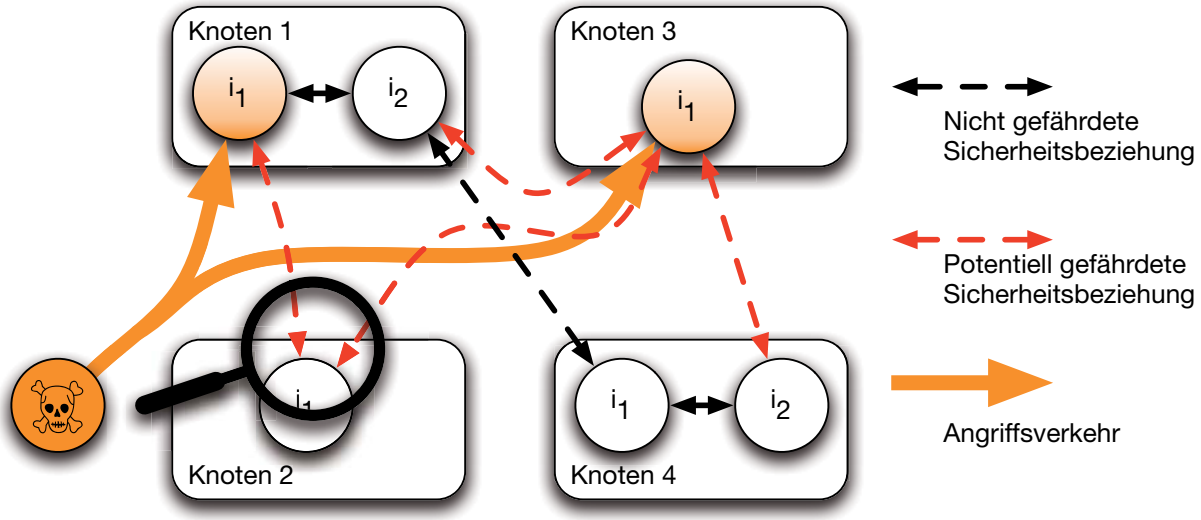


Abbildung 5.2: Beispiel für die Auswirkungen eines DoS-Angriffes der durch Beobachtung von Knoten 2 vorbereitet wurde

Für Worst-Case-Betrachtungen reicht es allerdings nicht aus, lediglich das Kompromittieren einzelner oder Gruppen bestimmter Knoten zu berücksichtigen. Stattdessen ist von einem intelligenten Angreifer auszugehen, der innerhalb eines festgelegten Budgets für die Beobachtung von Knoten eine bestimmte Zielfunktion optimal zu erfüllen versucht. Beispielsweise können die durch DoS-Angriffe betroffenen Sicherheitsbeziehungen, gewichtet mit ihrem Wert für den VPN-Betreiber, durch den Angreifer maximiert werden:

$$\max \sum_{e \in E} c_e a_e$$

Dabei entspricht E der Menge der Sicherheitsbeziehungen des VPN, c_e dem Wert einer bestimmten Sicherheitsbeziehung für das VPN und a_e einer binären Variable, die angibt ob die Kante durch den Angriff betroffen ist. Um den Angriff vorzubereiten, muss der Angreifer innerhalb seines Budgets (K) eine Reihe von Knoten beobachten, deren Kommunikationspartner er erfasst und angreift:

$$\sum_{v \in V} k_v x_v \leq K$$

Die Variablen k_v repräsentieren dabei die Kosten eines Angreifers, einen Knoten zu beobachten, und x_v ist eine weitere binäre Variable, die kodiert, ob der Knoten v tatsächlich beobachtet wird. Allerdings entspricht bereits dieses Auswahlproblem, welche Knoten unter optimaler Nutzung des Budgets angegriffen werden sollten, dabei dem einfachen Rucksackproblem ohne Gewichte. Weiterhin müssen die durch den DoS-Angriff betroffenen Sicherheitsbeziehungen auch zu den ausgewählten Knoten korrespondieren:

$$e \in E : a_e \leq \sum_{v \in V} i_{v \to e} \; x_v$$

Somit kann a_e nur einen Wert größer null annehmen, wenn mindestens ein beobachteter Knoten ebenfalls mit einer Netzwerkschnittstelle kommuniziert, welche für die Sicherheitsbeziehung verwendet wird. Die korrespondierenden Werte $i_{v \to e}$, die angeben ob die Beobachtung eines Knotens v die Sicherheitsbeziehung e gefährdet, werden dabei statisch vorberechnet. Auf das Beispiel in Abbildung 5.2 bezogen, bedeutet dies, dass für die Beobachtung von Knoten 2 gilt $i_{2 \to 1.2,3} = i_{2 \to 1.1,2} = i_{2 \to 2,3} = i_{2 \to 3,4.2} = 1$ und $i_{2 \to 1.2,4.1} = 0$. Die Variablen, zu denen Knoten mit mehreren Netzwerkschnittstellen korrespondieren, sind dabei jeweils in der Form *Knoten.Schnittstelle* kodiert. Es ergibt sich das folgende binäre, lineare Optimierungsproblem $attack_{budget}$:

$$\max \sum_{e \in E} c_e a_e$$

$$\sum_{v \in V} k_v x_v \leq K$$

$$\forall\, e \in E : a_e - \sum_{v \in V} i_{v \to e}\, x_v \leq 0$$

$$\forall\, e \in E : 0 \leq a_e \leq 1$$

$$\forall\, v \in V : x_v \in \{0, 1\}$$

Auch wenn a_e kontinuierlich optimiert werden kann, da sich durch die Optimierung die Werte 0 oder 1 implizit einstellen (a_e muss zwischen 0 und 1 liegen, gebrochene Werte stellen keine Extremalpunkte dar), verbleiben $|V|$ viele binäre Variablen im System. Aufgrund der diskreten Werte ist die optimale Lösung mit diesem Berechnungsmodell NP-schwer zu ermitteln, jedoch lässt sich bereits für das Ursprungsproblem zeigen, dass es NP-vollständig ist [Bra10]. Trotz ihrer NP-Vollständigkeit bietet die lineare Optimierung mit Binärzahlen aufgrund der gut erforschten Methoden und Werkzeuge eine Möglichkeit für kleine VPN optimale Vorhersagen zu treffen. Bei größeren VPN können durch die Korrelation von Primal- und Duallösung zumindest Approximationen mit garantierten Güten berechnet werden.

Allerdings berücksichtigt das skizzierte budgetbasierte Angreifermodell keine kaskadierenden Ausfälle. Sind beispielsweise lediglich sehr wenige gestörte Verbindungen notwendig, um ein VPN in mehrere Teile zu partitionieren, wird dieser Umstand nur berücksichtigt wenn die Kosten der Kanten auch entsprechend gewählt sind. Realistischer ist daher die verbleibende Konnektivität im VPN zu messen, und die Kosten des Angreifers für eine Attacke mit einem festgelegten Mindestschaden M zu minimieren. Da die Messung der Konnektivität im Residualgraphen nicht direkt in einem linearen Programm handhabbar darzustellen ist, verbleibt bei Beibehaltung dieser Methode nur ein Ausschluss ungültiger Angriffe, die zu wenig Schaden verursachen. Dazu wird, wie in Abbildung 5.3 gezeigt, für jeden möglichen Angriff bestimmt, ob er beispielsweise mehr als 50% des VPN betrifft. Ist dies nicht der Fall, muss zusätzlich mindestens einer der nicht beobachteten Knoten beobachtet werden, sodass für den Beispielangriff $x_0 + x_1 + x_2 + x_3 + x_5 + x_6 \geq 1$ als Nebenbedingung hinzugefügt wird. Über die Variable x_4 wird an dieser Stelle keine Aussage getroffen, da nicht klar ist, ob der korrespondierende Knoten in einem optimalen Angriff beobachtet wird. Der besseren Übersicht wegen wird dabei im Folgenden auf eine Modellierung

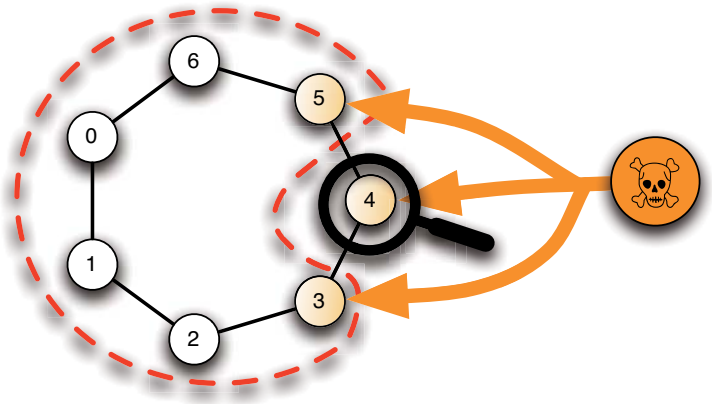


Abbildung 5.3: Beispiel für einen Angriff bei dem 50% des VPN verfügbar bleiben. Mindestens ein Knoten der umrandeten Menge muss durch zusätzliche Bobachtung aufgedeckt und angegriffen werden.

der Netzwerkschnittstellen verzichtet, und pro Knoten von genau einer Schnittstelle ausgegangen.

Wenn $\mathcal{P}(V)$ die Potenzmenge der Knoten und $\mathring{Y}$ die Menge der ungültigen Angriffe bezeichnen, ergibt sich folgendes binäres, lineares Optimierungsproblem $attack_{partition}$:

$$\min \sum_{v \in V} k_v x_v$$

$$\mathring{Y} \subseteq \mathcal{P}(V) \text{ so dass } Y \in \mathring{Y} \Leftrightarrow$$

$$\text{Angriff auf alle } v \in Y \text{ trennt nicht } M \text{ Ende-zu-Ende-Verbindungen}$$

$$\forall A \in \mathring{Y} : \sum_{a \in V \setminus A} x_a \geq 1$$

$$\forall v \in V : x_v \in \{0,1\}$$

Das entstehende Programm besitzt zwar lediglich $\mathcal{O}(|V|)$ binäre Variablen, allerdings müssen im schlimmsten Fall $\mathcal{O}(2^{|V|})$ Nebenbedingungen erzeugt werden, sodass keine vertretbare Laufzeit zu erwarten ist. Allerdings sind die Ungleichungen zum Teil redundant, sodass nicht alle vollständig formuliert werden müssen, sondern sich eines Schnittebenenverfahrens bei der Optimierung bedient werden kann. Dabei wird dynamisch während der Berechnung ein Programm zur Überprüfung der Residualkonnektivität aufgerufen und gegebenenfalls verletzte Bedingungen dem linearen Programm hinzugefügt. Der aufgerufene Algorithmus könnte dabei auch eine Wichtung der Kanten, je nach Wert für den Betreiber, vornehmen. Aus Gründen der Übersichtlichkeit wird darauf allerdings im Folgenden verzichtet.

Die Laufzeit zur Berechnung des linearen Programms ist je Struktur des Graphen und Wahl von M für mittlere VPN-Größen von bis zu 400 Knoten akzeptabel. Sehr viel schnellere Lösungsverfahren sind nicht zu erwarten, da das Angreifermodell ein abgewandeltes Vertex-Cover-Problem darstellt. Um ein Originalproblem zu lösen, müssen lediglich M auf die Anzahl der Kanten des Graphen und alle k_v auf den gleichen positiven Wert gesetzt werden. Anschließend wird auf jeder Kante ein zusätzlicher Knoten eingefügt, dessen Kosten die Gesamtkosten aller Originalknoten übersteigen. Durch

den Lösungsalgorithmus wird nun sichergestellt, dass je einer von zwei benachbarten Knoten des Originalgraphen überdeckt wird. Da die Reduktion in polynomieller Zeit erfolgen kann, ist das Problem NP-schwer.

Sollen an Stelle von Beobachtungskosten die Wahrscheinlichkeiten für erfolgreiche Identifikationen (p_v) verwendet werden, verändert sich die Optimierungsfunktion zu:

$$\max \prod_{v \in V} (1 - (1 - p_v)x_v)$$

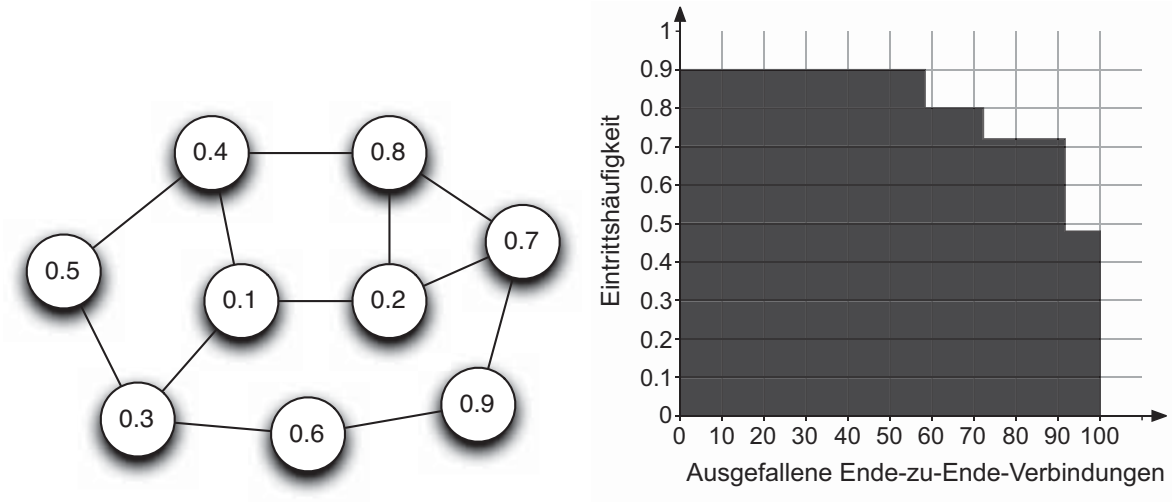
Der Angreifer versucht also, so wenige Knoten wie möglich und mit möglichst geringer Resistenz $(1 - p_v)$ zu beobachten. Die Gegenwahrscheinlichkeiten, welche eine gleichzeitige Beobachtung weiterer Knoten ausschließen, dürfen an dieser Stelle nicht mit einfließen. Es kann nicht vernachlässigt werden, dass ein Angriff erfolgen kann, obwohl das Budget des Angreifers eine unwahrscheinlichere Attacke zulässt.

Das entstandene Optimierungsproblem kann in zwei Schritten in ein lineares mit identischen Nebenbedingungen umgewandelt werden. Zunächst müssen alle Knoten mit einer Beobachtungswahrscheinlichkeit von $p_v = 0$ aus dem Beobachtungsvektor X entfernt werden, weil sie ohnehin nicht observiert werden können. Da nun alle verbleibenden p_v echt größer Null sind, kann das Problem durch Umformung auf ein lineares Optimierungsproblem zurückgeführt werden:

$$\begin{aligned}
\max \prod_{v \in V} (1 - (1 - p_v)x_v) \Leftrightarrow \max \log \prod_{v \in V} (1 - (1 - p_v)x_v) \\
&= \max \sum_{v \in V} \log (1 - (1 - p_v)x_v) \\
&= \max \sum_{v \in V} \begin{cases} \log p_v & \text{falls } x_v = 1 \\ 0 & \text{falls } x_v = 0 \end{cases} \\
&= \max \sum_{v \in V} x_v \log p_v
\end{aligned}$$

Auf die Beispieltopologie des hypothetischen VPN aus Abbildung 5.4(a) bezogen, ergeben sich so verschiedene optimale Angriffsstrategien: Zunächst kann ein Angreifer mit relativ hoher Wahrscheinlichkeit den Verkehr des Knoten mit Beobachtungswahrscheinlichkeit 0,9 analysieren und so bereits einen beträchtlichen Teil des VPN stören (21 der 36 Ende-zu-Ende-Beziehungen auf Overlay-Ebene). Andererseits ist die Wahrscheinlichkeit, dass die Knoten 0,9, 0,7 und 0,5 beobachtet werden zwar insgesamt relativ gering, allerdings würde das gesamte VPN ausfallen, da alle anderen Knoten überdeckt sind.

In Abhängigkeit von Angreifer und VPN können eine Reihe von Angriffen als optimal zur Erreichung eines Mindestschadens betrachtet werden. Für das Beispiel ergibt sich die Graphik von Abbildung 5.4(b), bei der Schaden eines Angriffs gegenüber seiner Eintrittshäufigkeit, also dem verbrauchten „Budget“ der Einzelwahrscheinlichkeiten, aufgetragen ist. Die Fläche unter der Kurve kann dabei als Anfälligkeit eines VPN



(a) (b)

Abbildung 5.4: Beispieltopologie mit Knoten und ihren jeweiligen Beobachtungswahrscheinlichkeiten (links) und die Effizienz der korrespondierenden Angriffe (rechts)

gegenüber dem beschriebenen Angriffsszenario aufgefasst werden. Die Gesamtfläche ergibt sich dabei durch einfaches Summieren der Teilflächen. Sowohl in Hub-and-Spoke- als auch vollvermaschten Szenarien ist diese Fläche beispielsweise sehr groß, da sie sich aus der Beobachtungswahrscheinlichkeit des exponiertesten Knotens und dem Ausfall aller Verbindungen ergibt. Wird das Beispiel-VPN mit einer dieser Topologien realisiert, beträgt die Anfälligkeit demzufolge $0,9$ — im Gegensatz zum Wert der Beispieltopologie von $0,82$ also ein wenig höher. Mit der Anfälligkeit gegen Angriffe korrespondiert die Resistenz, welche die Fläche über der Kurve bildet.

Der Berechnungsaufwand für Anfälligkeit beziehungsweise Resistenz hängt stark von der Struktur der untersuchten VPN-Topologie ab, allerdings macht sich der exponentielle Aufwand oft erst bei der Abschätzung optimaler Angriffe bemerkbar, welche auf eine nahezu vollständige Störung des VPN abzielen. Die Berechnung kann in manchen Topologien jedoch durch einen Vorberechnungsschritt beschleunigt werden, indem Knoten entfernt werden, welche bei optimalen Angriffen ohnehin nicht beobachtet würden. Die Beobachtung ist dabei genau dann unnötig, wenn eine Menge anderer Knoten existiert, die zum einen eine geringere Gesamtwahrscheinlichkeit als die Einzelwahrscheinlichkeit des Knoten besitzt, und zum anderen alle Nachbarn des Knoten sowie den Knoten selbst überdeckt.

Bereits diese noch einfachen statischen Angreifermodelle, welche weder Reaktionen des VPN noch komplexere Metriken wie die Reduzierung des VPN-Durchsatzes berücksichtigen, erlauben allerdings aufgrund ihrer Berechnungskomplexität nur sehr beschränkte Aussagen. Zum einen sind sie für große Netze nur noch näherungsweise zu berechnen. Zum anderen können daraus, wie im folgenden Abschnitt näher ausgeführt, nicht direkt allgemeine Gegenmaßnahmen für die Konstruktion besonders stabiler VPN abgeleitet werden.

5.3 Schutz vor externen DoS–Angriffen

Die im letzten Abschnitt vorgestellte Bewertung potentieller DoS-Angriffe stellt eine Grundlage dar, VPN-Topologien überhaupt erst in Bezug auf ihre Angriffsresistenz zu unterscheiden. Die sich ableitende Konstruktion besonders stabiler Topologien wird im Folgenden vorgestellt. Dabei wird zunächst das Problem erläutert, optimale Angriffsresistenz zu erreichen, bevor auf verschiedene heuristische Modelle eingegangen wird.

5.3.1 Optimale VPN–Topologien

Die Konstruktion optimal stabiler VPN-Topologien aus konkreten Angreifermodellen führt auf ein Stackelbergspiel (nach Freiherr von Stackelberg) [Dem02]. Dabei gibt ein so genannter Stackelbergführer oder *Leader*, in diesem Fall der VPN-Betreiber, eine Situation vor, auf welche der Stackelbergfolger (*Follower*) in Bezug auf eine Optimierungsfunktion reagiert. Der Angreifer würde also in Bezug auf die vorgegebene Topologie mit einem optimalen Angriff reagieren. Es ergibt sich folgendes diskretes Bileveloptimierungsproblem, bei dem der Betreiber versucht den Angriffsgewinn unter gleichzeitiger Berücksichtigung von Kosten (beispielsweise die Anzahl der aufgebauten Overlay-Verbindungen) optimal einzustellen:

$$\text{Betreiber: } \min_{x} \{attackGain(x,y) + c \cdot costs(x), \text{ für zulässige Topologien } x\}$$
$$\text{Angreifer: } \max_{y} \{attackGain(x(y),y), \text{ für zulässige Angriffe } y\}$$

Die Struktur des Bilevelproblems ist für eine diskrete Fragestellung vergleichsweise einfach: Beide Teilprobleme sind linear, und durch das Auftauchen der Zielfunktion des Angreifers in der des Betreibers kann eine optimistische Lösung des Problems erfolgen. Daher genügt es die Topologie gegen einen bestimmten der jeweils besten Angriffe zu sichern, und es müssen nicht erst alle möglichen Angriffe mit gleichem optimalen Angriffspotential berechnet werden. Dennoch zeigt der aktuelle Stand der Forschung [Bar99, Dem02, Lö10], dass diese Probleme für die benötigte Anzahl an Variablen aktuell nicht in angemessener Zeit zu berechnen sind.

5.3.2 Schutz durch Verfügbarkeitszonen

Bedingt durch die hohe Berechnungskomplexität des idealen Modells müssen andere Mechanismen geschaffen werden, die es einem Betreiber garantieren, dass bestimmte Bereiche des VPN nicht durch die Beobachtung anderer aufzudecken und somit klar definierte Schutzziele zu erfüllen sind. Beispielsweise ist in Abbildung 5.5 eine Situation dargestellt, bei der es einem Außendienstmitarbeiter nicht erlaubt sein soll, auf einen Verzeichnisdienst direkt zuzugreifen. Der Zugriff muss immer indirekt über einen Knoten erfolgen, dessen Ausfall eher akzeptabel ist.

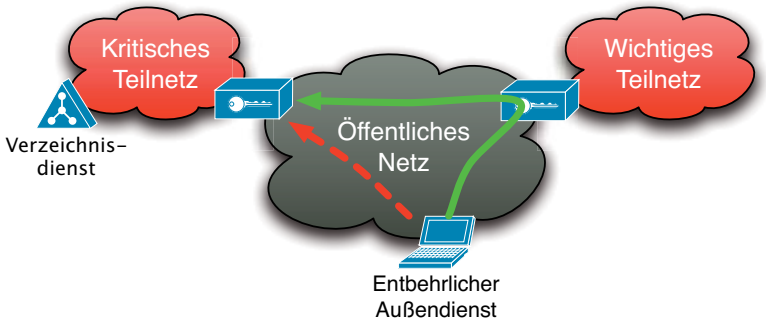


Abbildung 5.5: Anwendungsszenario Außendienstmitarbeiter soll keinen direkten Zugriff auf kritische Dienste haben

Um diese Anforderungen zu modellieren und beherrschbar zu machen, werden die VPN-Knoten nach ihrer Gefährdung und Relevanz für das VPN gruppiert und somit in *Verfügbarkeitszonen* eingeteilt. Analog zu den Schutzstufen der Multi-Level-Sicherheitssysteme, beispielsweise nach Bell-LaPadula [BL73] oder dem Biba-Modell [Bib75], können diese eine Ordnung und konkrete Namen besitzen, um komplexe Anforderungen in einfachere, noch zu beherrschende Modelle umzuwandeln. Für das Beispiel aus Abbildung 5.5 würde sich eine Aufteilung in KRITISCH > WICHTIG > ENTBEHRLICH anbieten. Nach erfolgter Aufteilung wird der direkte Verbindungsaufbau zwischen Knoten unterbunden, wenn diese sich nicht in der gleichen Zone oder in einer laut Ordnung benachbarten Zone befinden.

Durch diesen Ansatz sind Beziehungen jedoch im Gegensatz zur Informationsflusskontrolle der Multi-Level-Sicherheitssysteme bidirektional. Es wird ferner nicht in Subjekte und Objekte unterschieden, und es muss bei der Kommunikation zwingend über Knoten jeder Zwischenzone weitergeleitet werden, da es sich nicht länger um eine transitive Eigenschaft handelt. Des Weiteren ergibt sich zumindest die theoretische Möglichkeit, Zonenübergänge auch ohne feste Ordnung zu erlauben, indem jeder Verfügbarkeitszone Nachbarzonen über eine symmetrische Relation explizit vorgegeben werden. Es ergibt sich eine direkte VPN-Ebene, wie sie in Abbildung 5.6 beispielhaft illustriert ist.

Als direkte Konsequenz folgt aus dem Modell eine klar beschränkte Sicht, die durch das Beobachten von Knoten einzelner Zonen im schlimmsten Fall entstehen kann. Je nach Anwendungsfall kann der VPN-Betreiber durch Anzahl der Verfügbarkeitszonen, die Nachbarschaftsrelation und die Zuordnung der VPN-Knoten die Beobachtbarkeit frei einstellen. Allerdings impliziert eine reduzierte Beobachtbarkeit auch tendenziell einen geringeren Zusammenhalt der direkten VPN-Ebene, und einen erhöhten Kommunikationsaufwand durch vermehrte Paketweiterleitung.

Planung von Zonenzuordnungen

Insbesondere der angedeutete geringere Zusammenhalt durch sehr viele Verfügbarkeitszonen bietet wiederum Raum für potentielle DoS-Angriffe, da das VPN leichter partitioniert werden kann. Die optimale Zergliederung eines bestehenden Netzes ist

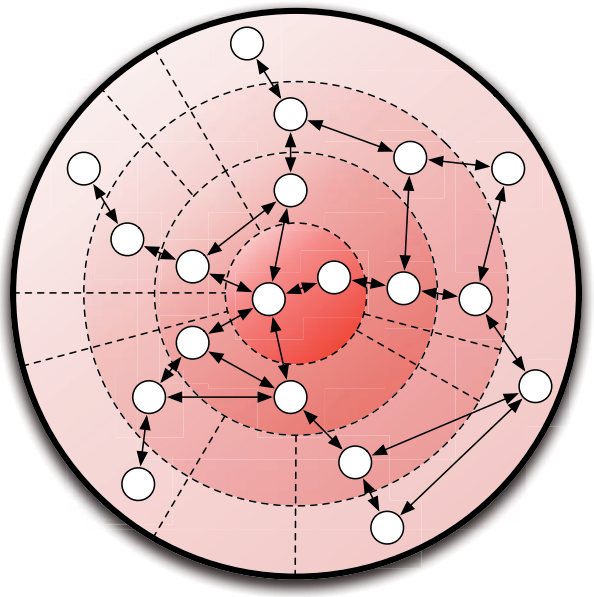


Abbildung 5.6: Beispieltopologie für ein VPN mit Verfügbarkeitszonen: direkte Kommunikation ist nur innerhalb benachbarter Zonen möglich, das heißt Kommunikation zwischen Knoten über mehrere gestrichelte Linien hinweg muss über die VPN-Overlay-Ebene erfolgen

daher ebenfalls ein Optimierungsproblem, welches durch die Einflussnahme auf die Topologie des VPN diskret ist. Für ein ähnlich strukturiertes Optimierungsproblem wurde in [Bra10] gezeigt, dass es NP-vollständig ist. Allerdings eignet sich die folgende Formulierung dennoch für kleinere VPN und um daraus Tendenzen für stabile Netze abzuleiten.

Ein mögliches Optimierungsziel stellt die Minimierung der Angriffsfläche dar, welche sich durch die Summe der Beobachtungswahrscheinlichkeiten der einzelnen VPN-Verbindungen $e_{i,j}$ ergibt:

$$\min \sum_{i \in V} \sum_{j \in V} \left(1 - (1 - p_i)(1 - p_j)\right) e_{i,j}$$

Eine Totalordnung der Verfügbarkeitszonen vorausgesetzt, würde eine naive Aufteilung der Knoten über die Zuweisung von Ganzzahlen erfolgen, welche die Zonen kodieren. Allerdings ist diese Zuordnung nicht linear, da eine Kommunikation bei einer Distanz von maximal 1 möglich sein soll, darüber hinaus jedoch nicht mehr. Daher erfolgt die Klassifizierung von Knoten in Verfügbarkeitszonen ($z \in Z$) über eine One-Hot-Kodierung [Mar07] mithilfe der Matrix C, das heißt es darf jeweils nur ein $c_{i,z}$ pro Zeile den Wert „1“ annehmen.

$$\forall\, i \in V,\, z \in Z : c_{i,z} \in \{0, 1\}$$
$$\forall\, i \in V : \sum_{z \in Z} c_{i,z} = 1$$

Die erste Regel sichert dabei zu, dass die Verteilung diskret erfolgt, während die zweite Regel die Zuordnung zu genau einer Zone erzwingt.

Die Sicherheitsbeziehungen des VPN werden genau dann erzeugt, wenn die beteiligten Knoten in gleichen oder benachbarten Zonen liegen. Wird ohne Beschränkung der Allgemeinheit eine Totalordnung für die Nachbarschaftsrelation vorausgesetzt, ergibt sich für alle $i, j \in V$ mit $i \neq j$:

$$(c_{i,0} \wedge (c_{j,0} \vee c_{j,1})) \vee (c_{i,1} \wedge (c_{j,0} \vee c_{j,1} \vee c_{j,2})) \vee \cdots \vee (c_{i,|Z|-1} \wedge (c_{j,|Z|-2} \vee c_{j,|Z|-1})) = e_{i,j}$$

Die Sicherheitsbeziehung $e_{i,j}$ wird demzufolge aufgebaut, wenn Knoten i und j sich in der gleichen Zone befinden oder in einer Zone nebeneinander. Durch diese Nebenbedingungen wird $e_{i,j}$ gleichzeitig implizit binär, und kann durch kontinuierliche Optimierung ermittelt werden. Einzig der Wertebereich ist zur schnelleren Optimierung eingeschränkt: $0 \leq e_{i,j} \leq 1$. Da die verwendeten logischen Verknüpfungen nicht notwendigerweise zu linearen Beziehungen führen, erfolgt eine Umformulierung:

$$c_{i,0}(c_{j,0} + c_{j,1}) + c_{i,1}(c_{j,0} + c_{j,1} + c_{j,2}) + \cdots + c_{i,|Z|-1}(c_{j,|Z|-2} + c_{j,|Z|-1}) = e_{i,j}$$

Die UND-Verknüpfungen werden einfach durch die äquivalente Multiplikation und die ODER-Operationen durch Additionen ersetzt. Letztere Ersetzung ist lediglich möglich, da jeweils nur einer der Teilterme durch die eindeutige Klassenzuordnung wahr werden kann, und die Addition in diesem Fall somit ebenfalls äquivalent ist. Eine Multiplikation kann beim Einsatz linearer Optimierungsmethoden nicht direkt berücksichtigt werden. Aus diesem Grund erfolgt eine Substitution der einzelnen Produkte durch die neu eingeführten Variablen $r_{i,j,z}$:

$$r_{i,j,0} + r_{i,j,1} + \cdots + r_{i,j,|Z|-1} = e_{i,j}$$

Diese werden durch die folgenden Nebenbedingungen zum einen ebenfalls implizit binär und zum anderen genau dann 1 wenn alle Faktoren 1 sind. Es gilt für alle $i, j \in V$ mit $i \neq j$ und $z \in Z$:

$$r_{i,j,z} - c_{i,z} \leq 0$$

$$r_{i,j,z} - c_{j,z-1} - c_{j,z} - c_{j,z+1} \leq 0$$

$$c_{i,z} + c_{j,z-1} + c_{j,z} + c_{j,z+1} - r_{i,j,z} \leq 1$$

Die aktuell definierten Nebenbedingungen erzeugen zwar bereits gültige VPN-Konfigurationen, allerdings wird kein zusammenhängendes Overlay gefordert, sodass eine Aufteilung laut Optimierungsbedingung am Besten durch die Partitionierung in möglichst viele Teilnetze erfolgt. Daher muss ein gewisser Kantenzusammenhang $\mathcal{K}$ des Netzes für gültige Lösungen gefordert werden. Um diesen zwischen zwei Knoten $i, j \in V, i \neq j$ innerhalb eines linearen Programms zu bestimmen, kann jeweils ein Flussnetzwerk verwendet werden. Es gilt für alle Flüsse u des Netzwerks:

$$\forall\, k, l \in V : 0 \leq u_{i,j,k,l} \leq e_{k,l}$$

Ein Fluss der Kapazität 1 kann genau dann zwischen Knoten l und k fließen, wenn die Kante im VPN vorhanden ist. Die Variablen $u_{i,j,l,k}$ werden dadurch ebenfalls implizit binär. Es folgt die Regel zur Flusserhaltung:

$$\forall k \in V : \sum_{l=1}^{n} u_{i,j,k,l} - \sum_{l=1}^{n} u_{i,j,l,k} \begin{cases} \geq \mathcal{K} & \text{falls } i = k \wedge j \neq k \\ \leq -\mathcal{K} & \text{falls } j = k \wedge i \neq k \\ = 0 & \text{sonst} \end{cases}$$

Die Summen der eingehenden und ausgehenden Flüsse müssen sich für alle Knoten entsprechen. Lediglich bei Quelle und Senke muss die Summe der Flüsse den geforderten Mindestzusammenhang erreichen. Durch die maximale Kantenkapazität von 1 müssen mindestens $\mathcal{K}$ kantendisjunkte Pfade zur Weiterleitung verwendet werden, wodurch $\mathcal{K}$-Kantenzusammenhang gewährleistet sein muss.

Um auch $\mathcal{K}$-Knotenzusammenhang zu fordern, genügt es mithilfe von Nebenbedingungen den maximalen eingehenden Fluss für jeden Knoten außer der Senke auf 1 zu beschränken:

$$\forall i, j, k \in V, j \neq k : \sum_{l=1}^{n} u_{i,j,l,k} \leq 1$$

Anschließend kann das insbesondere für die Robustheit gegen Ausfälle interessante Maß des Knotenzusammenhangs auf ein Mindestniveau festgesetzt werden.

Die Anzahl der benötigten Variablen und Nebenbedingungen kann auf nahezu die Hälfte reduziert werden, indem die Symmetrie der Klassenbeziehungen ausgenutzt wird. Aufgrund derer ist $e_{i,j} = e_{j,i}$ und $j \leq i$ müssen daher nicht betrachtet werden. Ferner müssen $e_{i,j}$ und die korrespondierenden $u_{i,j,k,l}$ nur definiert werden, falls im Transportnetz überhaupt eine direkte Verbindung möglich ist.

Trotz dieser Vereinfachungen besteht das Optimierungsproblem im Allgemeinen aus $\mathcal{O}(|Z| \cdot |V|)$ binären Variablen und $\mathcal{O}(|Z| \cdot |V|^2 + |V|^2 + |V|^4)$ implizit binären Variablen, sodass es selbst für kleine Probleminstanzen schwer zu berechnen ist.

Aus diesem Grund wurde ein leicht modifizierter Ansatz zum Erzwingen eines minimalen Zusammenhaltes umgesetzt. Dazu werden zunächst keine Flussbedingungen erstellt und somit $\mathcal{O}(|V|^4)$ Variablen eingespart. Stattdessen wird jeder vom Optimierungsalgorithmus vorgeschlagene Graph durch ein imperatives Programm in randomisierter Reihenfolge auf $\mathcal{K}$-Zusammenhang überprüft.

Ist ein Mindestzusammenhang gegeben, ist der Graph optimal und die Optimierung wird beendet. Wird jedoch eine Verletzung der Bedingung erkannt, erfolgt dynamisch das Erstellen einer weiteren Nebenbedingung, welche die vorgeschlagene Lösung explizit ausschließt. Wie in Abbildung 5.7 illustriert, wird zur Identifikation geeigneter Nebenbedingungen die vorgeschlagene VPN-Topologie in gerichtete Flussgraphen umgewandelt.

Bei der Umwandlung werden außer der Senke und der Quelle alle Knoten des Ursprungsproblems in zwei neue Knoten zerteilt, welche mit einer gerichteten Kante verbunden werden, um so eine Knotenkapazität zu modellieren. Alle Kanten des VPN werden als gerichtete übernommen. Anschließend kann durch Standardalgorithmen

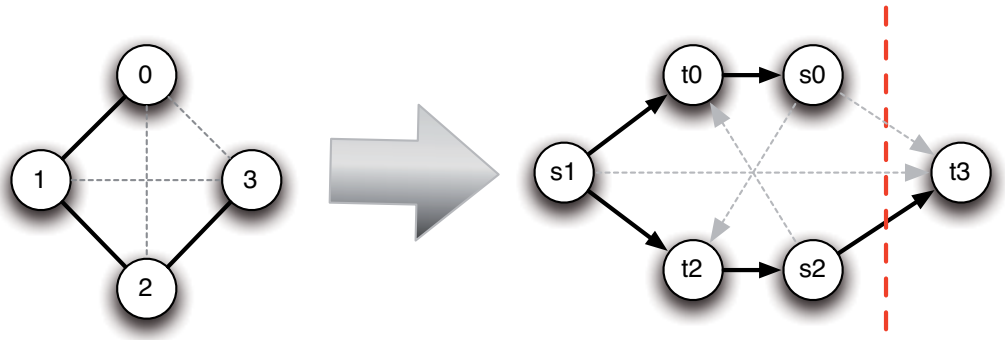


Abbildung 5.7: Umformung einer Beispieltopologie in ein gerichtetes Flussproblem zur Berechnung minimaler Schnitte

ein minimaler Schnitt berechnet werden, welcher die Knoten und damit auch die Knoten des Ursprungsproblems in zwei Teile S und $\bar{S}$ partitioniert. Da der maximale Fluss unterhalb der geforderten Mindestmenge $\mathcal{K}$ liegt, wird folgende Nebenbedingung eingefügt:

$$\sum_{i \in S} \sum_{j \in \bar{S}} e_{i,j} \geq \mathcal{K}$$

In der Beispieltopologie aus Abbildung 5.7 könnte dies beispielsweise $e_{0,3} + e_{1,3} + e_{2,3} \geq 2$ sein. Eine Besonderheit tritt beim Schnitt durch Kanten auf, welche die Kapazität eines Knoten des Ursprungsproblems repräsentieren: Diese können einer der beiden Schnittmengen zugeordnet werden, da eine Verschiebung des Schnittes auf ihre Ein- beziehungsweise Ausgangskanten keine Auswirkungen hat.

Angewendet auf die Knoten aus dem Beispielnetzwerk in Abbildung 5.4(a) ergeben sich durch die Optimierung VPN-Topologien, wie die in Abbildung 5.8 dargestellten. Über die Anzahl an Verfügbarkeitszonen und den geforderten Zusammenhang kann dabei zwischen Effizienz und Resistenz abgewogen werden. Für das erste Beispiel durften drei Verfügbarkeitszonen verwendet werden, und es war ein Mindestzusammenhang von 2 gefordert. An dieser Stelle ist durch die Separierung der besonders anfälligen Knoten gegenüber einem vollvermaschten Szenario bereits eine verbesserte Resistenz zu erkennen. In letzterem hätte die Beobachtung eines beliebigen Knotens und der damit korrespondierende Angriff bereits zu einem vollständigen Ausfall geführt.

Bei der Berechnung des zweiten Beispiels war die Verteilung auf 20 Verfügbarkeitszonen erlaubt; dabei war ebenfalls ein Mindestzusammenhang von 2 gefordert. Der scheinbare Widerspruch neun Knoten auf 20 Zonen aufzuteilen, liegt in der festen Anordnung der Zonen begründet, die vor der Optimierung definiert werden muss. Dadurch ist es unter Umständen sinnvoll, bestimmte Zonen freizulassen. Trotz des durch die Zonenmatrix reduzierten Freiheitsgrades beim Aufbau von Verbindungen und trotz der vorgenommenen Vereinfachung in der Berechnung der Zonenzugehörigkeiten, besitzt dieses VPN eine geringfügig niedrigere Angriffsanfälligkeit von $0,77$, welche sich analog zu Abschnitt 5.2 durch Summieren der Einzelflächen in Abbildung 5.8(d) bestimmt.

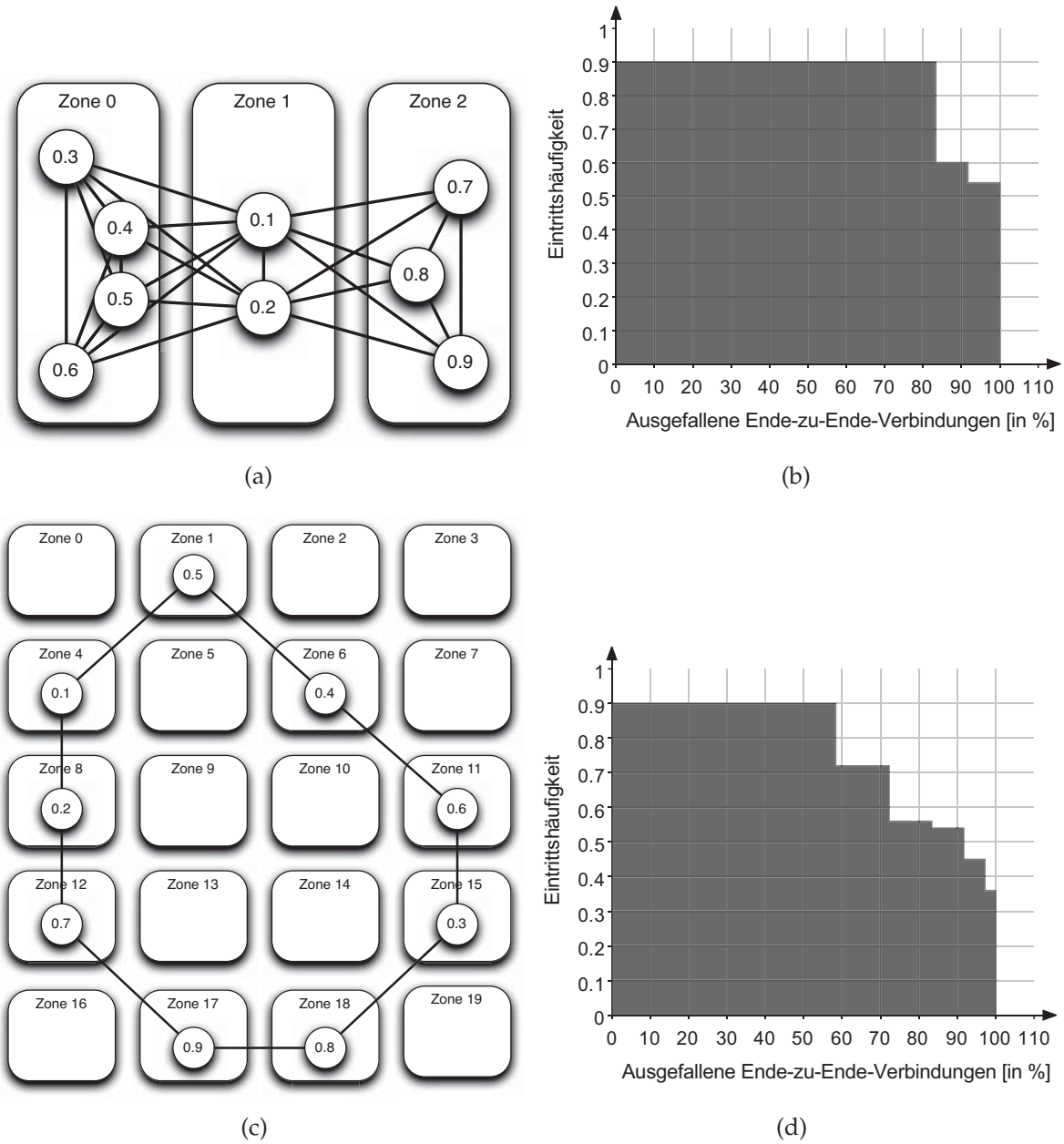


Abbildung 5.8: Optimierte Versionen der Beispieltopologie (links) und die Effizienz der korrespondierenden Angriffe (rechts)

Für reale VPN, die teilweise mehrere tausend Knoten besitzen, wird die Planung optimaler Zonenzugehörigkeiten in absehbarer Zukunft aufgrund der damit verbundenen Berechnungskomplexität nicht möglich sein. Dennoch lassen sich einige Heuristiken aus Beispielen, wie den vorgestellten, ableiten:

- Während herkömmliche auf Verfügbarkeit ausgelegte Netze sich durch eine hohe Konnektivität auszeichnen, ist dies für flexible DoS-resistente VPN nicht länger gegeben. Die optimale Konnektivität hängt hier von Anzahl und Anordnung von Verfügbarkeitszonen sowie der Verteilung der Beobachtungswahrscheinlichkeiten ab; sie steigt aber tendenziell mit der Knotenanzahl langsam an.

- Die Knoten einer Verfügbarkeitszone sollten ungefähr gleiche Beobachtungswahrscheinlichkeiten besitzen. Andernfalls bildet der Knoten mit der höchsten Wahrscheinlichkeit ein bevorzugtes Ziel für Angreifer. Indem bei großen VPN Äquivalenzklassen von Knoten gleicher Beobachtungswahrscheinlichkeit gebildet werden, kann dieser Umstand auch verwendet werden, um den Suchraum bei der Zonenzuordnung zu verkleinern.
- Verfügbarkeitszonen mit Knoten, die über eine hohe Beobachtungswahrscheinlichkeit verfügen, sollten über Zonen verbunden werden, in denen ausschließlich besser geschützte Knoten die Weiterleitung übernehmen. Eine geringe Anzahl von Knoten in den weiterleitenden Zonen sichert dabei zu, dass Angreifer sich ohne größere Schäden vom Rest des Netzes separieren.
- Bezogen auf die niedrige Anzahl an Verbindungen und den damit verbundenen niedrigeren Effekt von Beobachtungen, sind analog zu herkömmlichen Netzen Ringe sehr gut zur Realisierung resistenter Overlay-Strukturen geeignet, da beliebige einzelne Knoten entfernt werden können, ohne dass eine Partitionierung auftritt.
- Knoten mit vergleichsweise hohen Beobachtungswahrscheinlichkeiten werden somit insgesamt in Randbereiche beziehungsweise gleiche Ringsegmente gedrängt. Die Knoten, welche über die niedrigsten Beobachtungswahrscheinlichkeiten verfügen, sollten die zentralen Positionen der Zonenstruktur einnehmen.

Auch wenn optimale Zonenzuordnungen nicht berechnet werden, können Administratoren oder zukünftig zu entwickelnde Werkzeuge anhand der Regeln eine heuristische Aufteilung der VPN-Knoten vornehmen, und so eine signifikant bessere DoS-Resistenz erreichen.

Monotones Routing

Die Einteilung in Verfügbarkeitszonen erlaubt zunächst eine Entkoppelung von VPN-Bereichen. Allerdings gewährleistet sie nicht zwangsläufig einen Zusammenhang der Zonen, falls die Knoten einer anderen Zone ausfallen. So können beispielsweise zwei Knoten der Zone KRITISCH durch die Transportnetzumgebung so von einander getrennt sein, dass sie lediglich mithilfe einer Weiterleitung von Knoten niedrigerer Verfügbarkeitszonen miteinander kommunizieren können. Da bei der zonenübergreifenden Kommunikation jedoch tendenziell häufiger mit Ausfällen zu rechnen ist, kann dies der Intuition des Begriffs „Zone“ widersprechen, welcher sprachlich einen gewissen Zusammenhang suggeriert. In Szenarien bei denen sich nicht alle Knoten direkt erreichen können, kann es also erforderlich sein, ein erzwungenes Routing der Daten in monotoner Zonenfolge durchzusetzen (siehe Abbildung 5.9). Auf diese Weise kann sichergestellt werden, dass Ausfälle in anderen Zonen keine Auswirkungen haben.

Die Berechnung optimaler statischer Topologien, die ein monotones Routing erlauben, kann dabei analog zur Berechnung im letzten Abschnitt erfolgen. Es muss beim

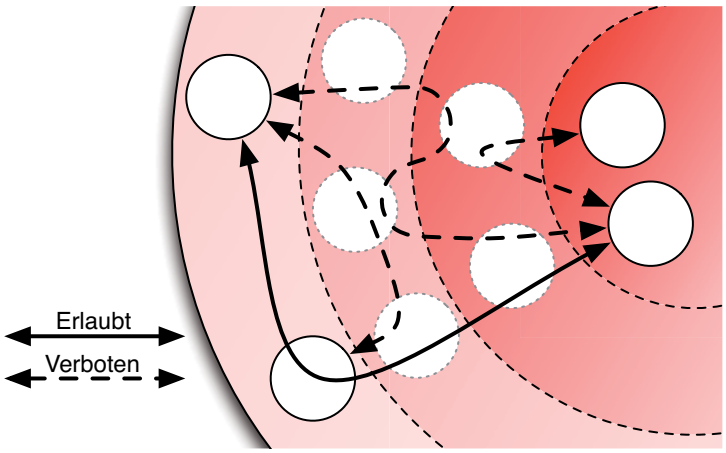


Abbildung 5.9: Beispiel für ein Durchsetzen eines bezogen auf die Zonenzuordnungen monotonen Routing

dynamischen Hinzufügen von Nebenbedingungen lediglich gewährleistet werden, dass neben dem Mindestzusammenhang im Gesamtgraphen auch jeweils ein bestimmter Zusammenhang in den einzelnen Zonen erreicht wird.

5.4 Schutz gegen interne DoS–Angriffe

Interne Angreifer können ebenfalls sehr viel Netzwerkverkehr generieren und so beispielsweise DoS-Angriffe auf Teile des VPN verüben, die aus dem Transportnetz für externe Angreifer nicht zu erreichen wären. Dabei existieren zwei Teilszenarien: Zum einen könnten Klienten innerhalb der roten Netze kompromittiert werden. Zum anderen können ein oder mehrere VPN-Knoten direkt von Angreifern kontrolliert werden. Während der erste Fall gut durch Traffic-Shaping im korrespondierenden VPN-Gateway zu beherrschen ist, muss bei der Berücksichtigung kompromittierter VPN-Knoten praktisch von Verkehrsflüssen ausgegangen werden, welche nur durch die Transportnetzkapazität der mit dem Angreifer direkt verbundenen Knoten begrenzt werden. Wie in Abbildung 5.10 dargestellt ist, kann ein interner Angreifer seine eigene Netzbandbreite vergrößern, indem er die aktuell verwendeten Schlüssel weiteren vom Angreifer kontrollierten Systemen mitteilt und dort ebenfalls kryptographisch validen Verkehr erzeugen lässt.

Im Folgenden werden zwei verschiedene Ansätze zur Realisierung von Schutzmaßnahmen gegen solche Angriffe vorgestellt: ein Mechanismus zum proaktiven Schutz vor DoS-Angriffen und eine flexiblere, reaktive Lösung. Falls neben Klienten auch kompromittierte Knoten berücksichtigt werden sollen, sind beide Konzepte allerdings ergänzend zur Einschränkung durch Verfügbarkeitszonen oder statische VPN-Konfiguration zu sehen, da interne Angreifer andernfalls alle VPN-Teilnehmer direkt kontaktieren können.

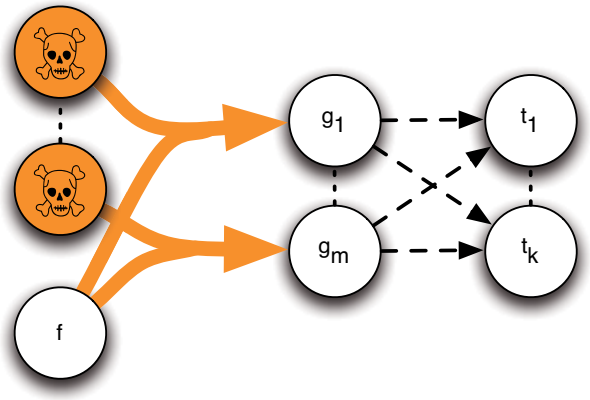


Abbildung 5.10: Interner DoS-Angriff: Knoten f teilt seine Schlüssel einem Bot-Netz mit, sodass über alle direkt verbundenen Knoten $g_{1\ldots m}$ ein Verkehrsfluss zu den Zielen $t_{1\ldots k}$ entstehen kann

5.4.1 Garantierte Vorhersagen der Grenzen interner DoS–Angriffe

In Szenarien mit sehr hohem Schutzbedarf kann es erforderlich sein, dass ein gewisses Kontingent an Netzwerkbandbreite zu jedem Zeitpunkt zur Verfügung steht, beispielsweise um die Fernadministration von Systemen gewährleisten zu können. Statische Garantien erfordern allerdings auch statische Topologien und festgelegte Obergrenzen für Verkehrsflüsse.

Da bei internen Angreifern nicht klar ist, welche Systeme kompromittiert werden, müssen die weitergeleiteten Verkehrsflüsse in jedem einzelnen Knoten beschränkt werden. Für die Sicherheitsbetrachtungen ergibt sich ein Tensor, welcher den maximal erlaubten Fluss beschränkt, den der Verkehr zwischen zwei VPN-Knoten auf dem Pfad zwischen zwei anderen Knoten induzieren darf. Wie in [BRS09] näher erläutert, kann dieser Tensor anschließend auf Sicherheitseigenschaften untersucht werden. Beispielsweise mithilfe von Multicommodity-Flow-Modellen, können so beim gleichzeitigen Einsatz von Verfügbarkeitszonen Aussagen über den maximalen Schaden getroffen werden, den kompromittierte Systeme in anderen Zonen verursachen können. Die statischen Zuweisungen bedeuten unter Umständen aber eine nicht akzeptable Ressourcenverschwendung, da in sicheren Umgebungen die reservierten Bandbreiten nicht anderweitig verwendet werden können.

Neben der statischen Zuweisung von Flusskapazitäten, sprechen auch die realisierbaren Routing-Modelle gegen dieses Verfahren. Mathematisch vergleichsweise einfach zu erfassen sind lediglich statische Verkehrspfade oder ein perfektes Multipath-Routing auf Basis der angesprochenen Flussnetzwerke. Beide Verfahren unterscheiden sich aber signifikant von praktisch einsetzbaren Verfahren wie OSPF oder auch SOLID, sodass deutliche Fehler bei der Einschätzung zu erwarten sind.

Gleichzeitig erlaubt die statische Beschreibung des Netzes theoretisch auch die Planung optimaler Angriffe beziehungsweise besonders angriffsresistenter Topologien und Tensoren. Dies führt aber bereits im Fall des Angreifers auf schwer zu lösende Bileveloptimierungsprobleme, da dieser die Reaktionen des Routing mit einbeziehen muss. Im Gegensatz zu den Bileveloptimierungsproblemen aus Teilabschnitt 5.3.1

sind diese zwar kontinuierlich, dennoch ist eine praktische Relevanz auf Basis des derzeitigen mathematischen Wissensstandes in Bezug auf Lösungsverfahren für solche Probleme nicht in der näheren Zukunft zu erwarten.

5.4.2 Reaktive Regelung von Datenflüssen

Innerhalb von VPN besteht neben einer statischen Regelung auch die Möglichkeit dynamisch auf interne DoS-Angriffe zu reagieren. Im Gegensatz zum Internet können hier die weiterleitenden Knoten beeinflusst, und so ein bedarfsgesteuertes Traffic Shaping durchgeführt werden. In drahtgebundenen Netzen muss ein VPN-Knoten dazu lediglich durch den Vergleich der Paketzähler und Sequenznummern seiner Sicherheitsbeziehungen eine Stausituation feststellen, und kann anschließend direkt benachbarte Knoten anweisen die Bandbreite der entsprechenden Flüsse zu reduzieren. Diese können eine entsprechende Drosselung vornehmen und sie über alle Pfadknoten entlang bis zu den Quellknoten signalisieren. Anschließend wird über einen einfachen Regelungsmechanismus die maximal pro Sicherheitsbeziehung zu übertragene Bandbreite langsam angehoben oder weiter abgesenkt.

Der Aktionsraum eines internen Angreifers wird durch den Mechanismus nicht vergrößert. Dieser kann zwar gezielt über ihn weitergeleitete Flüsse drosseln, und diese somit sabotieren, allerdings geht die Wirkung nicht über einen Grayhole-Angriff auf verschlüsselte TCP-Pakete hinaus. Die Staukontrolle von TCP sorgt hier ebenfalls für ein Absenken der Datenrate.

Ebenso wie bei TCP kann auch bei der reaktiven Regelung von Datenflüssen die Annahme, dass verworfene Pakete auf Stausituationen hindeuten, in drahtlosen Umgebungen zu Fehlschlüssen führen. An dieser Stelle könnte ein ähnliches System nur durch zusätzliches Wissen über die Umgebung konstruiert werden, beispielsweise indem vorher die aktuell ausgehandelte, maximale Bandbreite des Links mit dem eingehenden Datenstrom verglichen wird.

Weitere Probleme können theoretisch bei Halbduplex-Verbindungen und asymmetrischen Pfaden auftreten. Bei ersteren kann die Drosselungsnachricht möglicherweise nicht mehr übertragen werden, bei letzteren muss diese nicht zwangsläufig den gleichen Pfad nehmen. Beide sollten allerdings in realen VPN keine praktische Relevanz haben: Die gängigen Halbduplex-Übertragungsverfahren achten auf Fairness, sodass der Versand von Steuernachrichten kein Problem darstellen sollte. Asymmetrische Pfade werden zumindest bei den bekannten VPN-Autokonfigurationssystemen nicht verwendet. Unter Robustheits- und Verfügbarkeitsgesichtspunkten scheint der Einsatz auch nicht sinnvoll.

Im Gegensatz zum statischen Ansatz können durch eine einfache Verkehrsflussregelung keine echten Garantien gegeben werden. Allerdings scheint die reaktive Behandlung von DoS-Angriffen zumindest für die meisten VPN das adäquatere Mittel zu sein, da sie Bandbreiten besser ausnutzen kann und somit erheblich effizienter ist.

5.5 Verfügbarkeitszonen in SOLID-VPN

Die Realisierung von Verfügbarkeitszonen innerhalb flexibler SOLID-Netze erscheint zunächst vergleichsweise einfach. Durch die Unterteilung in direkte VPN-Ebene und VPN-Overlay können direkte Verbindungen nahezu beliebig eingeschränkt werden. Solange das entstehende VPN zusammenhängend ist, können prinzipiell alle VPN-Teilnehmer miteinander kommunizieren. Allerdings muss das Konzept in allen Teilsystemen umgesetzt werden, um zu verhindern, dass externe Beobachter oder interne Angreifer Kenntnis von nun schützenswerten äußeren IP-Adressen erlangen können.

5.5.1 Bootstrapping und Verbindungsaufbau

In Bezug auf das Bootstrapping bedeutet die Umsetzung des Ansatzes, dass Knoten unterschiedlicher Verfügbarkeitszonen auch unterschiedliche Dienste zum initialen Auffinden anderer Knoten der gleichen Verfügbarkeitszone verwenden. Die Knoten müssen ferner in der Lage sein, über das Bootstrapping die Knoten kompatibler Verfügbarkeitszonen abzufragen. In SOLID wird dies realisiert, indem die Knoten Bootstrap-Informationen über die eigene Zone und alle kompatiblen Zonen erhalten, welche im Vergleich die höhere Beobachtungswahrscheinlichkeit besitzen. Nachdem die IP-Adresse eines geeigneten Kommunikationspartners ermittelt wurde, erfolgt ein herkömmlicher kryptographischer Schlüsselaustausch und damit der Aufbau einer Verbindung.

Bei direkten Verbindungen muss während des Schlüsselaustausches erzwungen werden, dass lediglich Knoten kompatibler Zonen diesen Aufbau erfolgreich abschließen können. Problematisch erweist sich, dass weder IKEv2 noch TLS in der Lage sind die Identitäten der beteiligten Knoten ausreichend zu sichern. Auch bei einer Inkompatibilität der Identifikatoren kann wenigstens einer der beteiligten Knoten Rückschlüsse auf die Identität des anderen ziehen. Beim Einsatz von IKEv2 wird zumindest ein gewisser Grundschutz zugesichert: Hier muss der Initiator sein Zertifikat als erstes präsentieren. Dadurch ist das Protokoll zumindest resistent gegen systematische Suchen nach Knoten bestimmter Zonen. Beispielsweise durch Man-in-the-Middle-Angriffe oder durch von anderen Knoten initiierte Sicherheitsbeziehungen ist es allerdings dennoch möglich, die Zone eines Knoten ohne passendes Zertifikat zu ermitteln.

Letztlich zu beheben ist diese Schwachstelle nur durch ein Schlüsselaushandlungsprotokoll mit einer Zero-Knowledge-Authentisierung. Motiviert durch diese konkrete Problemstellung laufen zurzeit Untersuchungen zu effizienten Protokollen auf Basis bilinearer Pairings in elliptischen Kurven [Kö10, KRS11]. Die ersten Resultate sind vielversprechend; allerdings sind die zugrundeliegenden kryptographischen Primitive nicht hinreichend lange untersucht, um bei allen VPN-Betreibern als sicher angesehen zu werden. Ferner birgt die Entwicklung und Implementierung neuer kryptographischer Protokolle das Risiko von sicherheitskritischen Fehlern, sodass im Folgenden weiterhin vom Einsatz herkömmlicher Schlüsselaushandlungsprotokolle ausgegangen und dieser Aspekt in zukünftigen Arbeiten untersucht wird.

5.5.2 Topologiekontrolle und Routing

In Bezug auf die Auswahl proaktiv aufgebauter Sicherheitsbeziehungen und dynamischer Pfadwahl müssen zunächst nur wenige Modifikationen vorgenommen werden, da für SOLID die Einschränkung der Verfügbarkeitszonen mit Transportnetzeinschränkungen vergleichbar ist. Als einzige direkte Unterscheidung muss jetzt vor jeder Optimierungsmaßnahme geprüft werden, ob eine Kommunikation in Bezug auf Verfügbarkeitszonen möglich sein wird. Andernfalls würde möglicherweise die IP-Adresse eines Knoten in anderen Zonen bekannt. Weiterhin dürfen auch die periodisch übertragenen Verwaltungsinformationen keine äußeren IP-Adressen und Vivaldi-Koordinaten enthalten, wenn sie mehr als eine Zonengrenze passieren sollen.

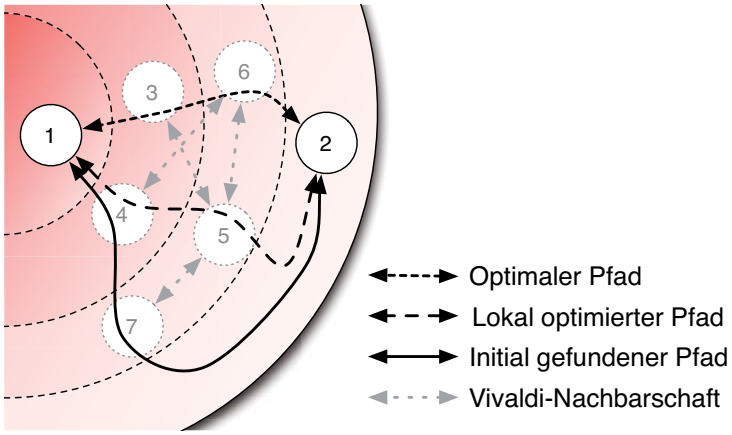


Abbildung 5.11: Beispiel für ein von Verfügbarkeitszonen verursachtes Fehlschlagen der lokalen Latenzoptimierung: Weder Knoten 4 noch Knoten 5 würden eine Optimierung initiieren, da die Pfade $2 \to 5 \to 3 \to 1$ und $2 \to 6 \to 4 \to 1$ jeweils eine Verschlechterung bedeuten. (Aus zeichnerischen Gründen wurde lediglich die relevante Vivaldi-Nachbarschaft dargestellt.)

Die Aufteilung der Knoten in Verfügbarkeitszonen hat jedoch einen Einfluss auf die Routing-Effizienz: So wird die in Abschnitt 4.5.3 vorgestellte Routing-Optimierung auf Basis von Vivaldi-Koordinaten sehr viel stärker zum Einsatz kommen, um Weiterleitungspunkte zwischen Knoten verschiedener Zonen so auszuwählen, dass die Gesamtverzögerung minimiert wird. Bei einfachen Transportnetzausfällen kann diese Optimierung lokal erfolgen. Wie in Abbildung 5.11 dargestellt ist, werden beim Einsatz von Verfügbarkeitszonen durch die potentiell längeren Pfade möglicherweise lediglich lokale Optima erreicht. Um dieses Problem für den allgemeinen Fall zu lösen, müssten die optimierenden Knoten über eine globalere Sicht verfügen. Dies ist zum einen aus Skalierbarkeitsgründen nicht anzustreben, und führt zum anderen auch dazu, dass Knoten Kenntnis von Vivaldi-Koordinaten und Sicherheitsbeziehungen nicht kompatibler Verfügbarkeitszonen erhalten müssen. Letzteres erlaubt potentiellen Angreifern möglicherweise Rückschlüsse auf Standorte und lohnende Ziele für DoS-Angriffe, sodass auf weitere Optimierungen an dieser Stelle verzichtet wird. Neben einer quantitativen Evaluierung des Nutzens von Verfügbarkeitszonen für SOLID-VPN,

sind somit die genauen Auswirkungen auf die Ende-zu-Ende-Latenzen zu untersuchen.

Ein ähnlich gelagertes Problem entsteht, wenn durch das Nichtbesetzen von Zonen Zyklen eingeführt werden (wie im Beispiel aus Abbildung 5.8 gezeigt). In diesem Fall kann durch das Vivaldi-Koordinatensystem keine Optimierung erfolgen, und auch durch SOLID's Routing-Mechanismus ist nicht garantiert, dass der kürzeste Weg bevorzugt wird. Auch in diesem Fall kann ein Finden optimaler Pfade lediglich durch eine globalere Sicht entstehen, sodass hierfür Informationen über aufgebaute Sicherheitsbeziehungen ausgetauscht werden müssen. Insgesamt unterscheidet sich dieser Umstand allerdings von seiner Struktur her nicht von nicht optimal aufzulösenden Zyklen, die aufgrund des Transportnetzes hervorgerufen werden.

Zusätzlich kann die Realisierung einer zonenmonotonen Knotensuche innerhalb von SOLID nur erfolgen, wenn keine durch unbesetzte Zonen entstehenden lokalen Minima oder Zyklen vorhanden sind. Aber auch anderenfalls kann ein erheblicher Mehraufwand entstehen: Wenn nicht bereits durch Aufteilung der inneren Netzadressen und die Struktur der Netzwerkkoordinaten sichergestellt wird, dass nur monotone Weiterleitungen erfolgen, müssen zusätzliche Sicherheitsbeziehungen aufgebaut werden, die dies ermöglichen. Das bedeutet, dass jeder Knoten für jede kompatible Zone proaktive Sicherheitsbeziehungen zu den bezüglich Ring- und Vivaldi-Struktur relevanten Nachbarknoten aufbauen muss. Dieser Aufwand dient aber lediglich einer monotonen Suche; für die eigentliche Weiterleitung der Nutzdaten wird durch die vom Umlenkungsalgorithmus vorgenommene Optimierung ohnehin ein monotoner Pfad entstehen, da dieser als einziger in Bezug auf die Hops optimal ist. Somit wird die zonenmonotone Suche nur vom Zeitpunkt einer Störung durch DoS-Angriffe bis zum Aufbau von Sicherheitsbeziehungen zu den Ringnachbarn des Residualnetzwerks benötigt. Für die meisten VPN ist dieses Risiko wahrscheinlich akzeptabel und lediglich wenn sichergestellt werden muss, dass auch die Suche bei DoS-Angriffen zu jedem Zeitpunkt bestmöglich erfolgen muss, ist der zusätzliche Aufwand zu betreiben.

5.6 Fazit

Der Aufbau von Strukturen, die sich durch eine besondere Resistenz gegen DoS-Angriffe auszeichnen, ist bisher durch Autokonfigurationsansätze trotz bekannter Defizite im Bereich der Sabotageresistenz nicht betrachtet worden. Aus diesem Grund wurde in Abschnitt 5.2 ein formales Modell für die Bewertung von DoS-Angriffen auf VPN-Infrastrukturen eingeführt. Da sich aufgrund der hohen Berechnungskomplexität daraus nicht direkt Methoden zur Konstruktion stabiler VPN ableiten lassen, erfolgt mit der Einführung von Verfügbarkeitszonen eine Vereinfachung, die es Administratoren und Netzwerkplanern ermöglichen soll, heuristisch möglichst sabotageresistente VPN aufzubauen.

Auf diese Weise lassen sich auch in Kombination mit einer flexiblen Einrichtung von VPN Aussagen über den Schaden treffen, der in bestimmten Bereichen des VPN

auftreten kann. Zumindest beim Einsatz von SOLID muss dabei abgewogen werden, inwiefern Zusatzaufwand betrieben wird, um ein optimales Routing in diesen Netzen bereitzustellen. Prinzipiell verursacht eine Umsetzung wegen der flexiblen VPN-Struktur jedoch vergleichsweise wenig Integrationsaufwand.

Bisher erfolgt die Konstruktion DoS-resistenter Topologien aufgrund der besseren Analyseeigenschaften ausschließlich auf Basis statischer Modelle. In Zukunft könnten möglicherweise dynamische Erweiterungen die DoS-Resistenz flexibler an die Gegebenheiten des Transportnetzes anpassen. Beispielsweise ist es denkbar, VPN-Knoten aufgrund ihrer geographischen Nähe zu gruppieren oder Knoten wichtiger werden zu lassen, wenn sie sich als besonders stabil erweisen. Umgekehrt ist es sinnvoll, mobile Knoten mit häufig wechselnden Aufenthaltsorten in Randbereiche des VPN zu verlagern.

```
/* FIXME: forward and
output firewalls used to
be called here. What do we
do with netfilter? - RR */
```

(Linux Kernel Code)

Kapitel 6

Implementierung

Um das entworfene SOLID-System genauer zu untersuchen, aber natürlich auch um es einzusetzen, ist unter anderem im Rahmen von [Ste08b, RSSM09] ein erster Prototyp entstanden, dessen Funktionsprinzipien im Folgenden vorgestellt werden. Für die Implementierung haben sich dabei zusätzliche Anforderungen herauskristallisiert, welche zunächst kurz dargestellt werden, bevor auf die eigentliche Software-Architektur eingegangen wird. Die anschließend illustrierten Details der Implementierung sind nach ihrer Realisierung in einzelne Dienste gegliedert.

6.1 Zusätzliche funktionale Anforderungen

Für den Einsatz des Prototypen sind primär zwei Arbeitsumfelder vorgesehen: zum einen der Einsatz in Forschung und Lehre zur Evaluierung sowie der Weiterentwicklung von SOLID-VPN und zum anderen eine perspektivische Nutzung in einem realen Produkt. Aus diesen Umfeldern ergeben sich die folgenden zusätzlichen Anforderungen:

1. **Evaluierbarkeit komplexer und großer Szenarien:** Um auch komplexe und sehr große VPN-Szenarien untersuchen zu können, beziehungsweise eine isolierte Betrachtung von Problemen zu ermöglichen, soll der Großteil des Prototyps in einem Netzwerksimulator zu untersuchen sein.

2. **Minimale Veränderungen an vorhandener Software:** Aus Gründen der Wartbarkeit, aber auch um möglichst wenige Anpassungen bei einer Umsetzung im Rahmen eines kommerziellen Projektes vornehmen zu müssen, soll an der gewählten Software-Umgebung möglichst wenig verändert werden. Insbesondere sollen möglichst wenige proprietäre Änderungen vorgenommen werden.

3. **Transparenz:** Die durch SOLID bereitgestellten zusätzlichen Dienste zur sicheren Übertragung von Unicast- und Multicast-Daten sollen keine Veränderungen an existierenden Klienten oder Applikationen erfordern. SOLID-Gateways müssen daher die Schnittstellen gewöhnlicher Internet-Router aufweisen.

Die Anforderungen bestimmen auch die in den nächsten Abschnitten vorgestellte Software-Architektur.

6.2 Einbettung in Simulator & Prototyp

Zur Untersuchung von SOLID in einem Netzwerksimulator bietet sich aufgrund der klaren modularen Struktur, der hohen Ausführungsgeschwindigkeit und der Vielzahl an implementierten Protokollen aus dem IP-Umfeld das OMNeT++/INET Framework [VH08] an. Bedingt durch diese Entscheidung ergibt sich zum einen die Festlegung auf C++ als Programmiersprache und zum anderen die Realisierung des Prototypen mittels einer ereignisbasierten Architektur.

Letztere zeichnet sich durch eine klare Gliederung der Funktionalität in Module aus, die nur ausgeführt werden, wenn sie eine Nachricht erhalten. Um periodische Aufgaben zu erledigen oder auf Zeitüberschreitungen zu reagieren, müssen sich die Module selbst eine Nachricht mit einer voreingestellten Verzögerung senden. Diese zunächst stark erscheinende Einschränkung bietet allerdings auch im Prototyp den Vorteil, dass auch komplexe Aufgaben mithilfe eines einzigen Threads umzusetzen sind. Race conditions und damit verbundene potentiell sicherheitskritische Fehler werden somit vermieden.

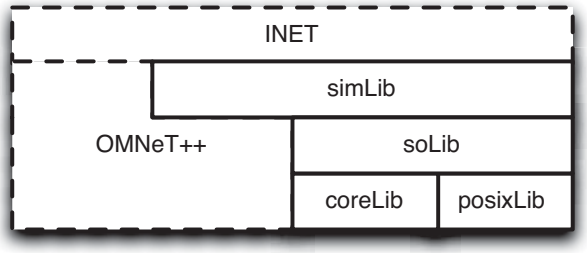


Abbildung 6.1: Software-Architektur zur Realisierung von SOLID in OMNeT++

Für die Realisierung von SOLID im Rahmen einer Simulation ergibt sich die in Abbildung 6.1 dargestellte Architektur. In das auf OMNeT++ aufbauende INET-Framework wird eine Simulationsbibliothek eingebunden, die neben Protokollimplementierungen, die von INET nicht direkt bereitgestellt werden, primär simulationsspezifische Anpassungen von SOLID enthält. Diese Bibliothek bindet wiederum `soLib` ein, welche alle Kernfunktionen von SOLID inklusive des Overlay-Routing, der Topologiekontrolle und verschiedener Bootstrapping-Mechanismen enthält. Alle SOLID-unabhängigen oder auf POSIX basierenden Funktionalitäten sind aus Gründen der Wiederverwendbarkeit noch einmal gekapselt.

Zur Realisierung in einem Prototyp existieren im Prinzip, wie in Abschnitt 2.3 bereits dargelegt, mit DTLS und IPsec zwei konkurrierende Netzsicherheitsprotokolle zur Einrichtung von VPN-Beziehungen. Beide setzen dabei ähnliche Mechanismen ein, sodass die Implementierung im Nutzerprozess oder Betriebssystemkern ein wesentliches Unterscheidungsmerkmal darstellt. Letzteres stellt in herkömmlichen Betriebssystemen allerdings ein nach wie vor nicht unerhebliches Effizienzkriterium dar, sodass aufgrund der Gateway- und Weiterleitungsfunktionalität von SOLID der Einsatz von IPsec Geschwindigkeitsvorteile verspricht. Außerdem können so existierende Implementierungen der IP-Verarbeitung des Betriebssystems weiterverwendet werden, und müssen nicht im Nutzerprozess erneut implementiert werden.

Als Software-Umgebung zur Umsetzung des IPsec-basierten SOLID-Systems wurde Linux gewählt, da es sich durch seinen umfangreichen Netzwerk-Stack zum einen hervorragend für den Einsatz auf Routern eignet. Zum anderen wird durch die offene und freie Architektur die IPsec-Implementierung nachvollziehbar und gegebenenfalls vergleichsweise leicht zu erweitern. Weiterhin wird *Charon*, der IKEv2-Daemon des strongSwan-Projektes [Ste08a], genutzt, um die Parameter für IPsec-Beziehungen auszuhandeln. Auch dieser zeichnet sich durch eine offene, modulare Architektur aus, besitzt einen großen Funktionsumfang und wird zudem noch aktiv weiterentwickelt.

Wie in Abbildung 6.2 dargestellt, werden in dieser Software-Umgebung die gleichen Bibliotheken zur Umsetzung von SOLID eingesetzt. Die Architektur ist aber allein in Bezug auf ihre Schnittstellen wesentlich komplexer. Neben Übergängen in den Linux-Kern zur Beeinflussung von Routing, Firewalls und IPsec-Politiken, existieren beispielsweise Informationsflüsse zu einem DHCP-Server-Daemon und zu strongSwan's Charon-Dienst. Bevor die Schnittstellen im Einzelnen in Bezug auf die mit ihnen realisierten Aufgaben vorgestellt werden, erfolgt jedoch zunächst ein kurzer Gesamtüberblick über die interne Software-Architektur von SOLID.

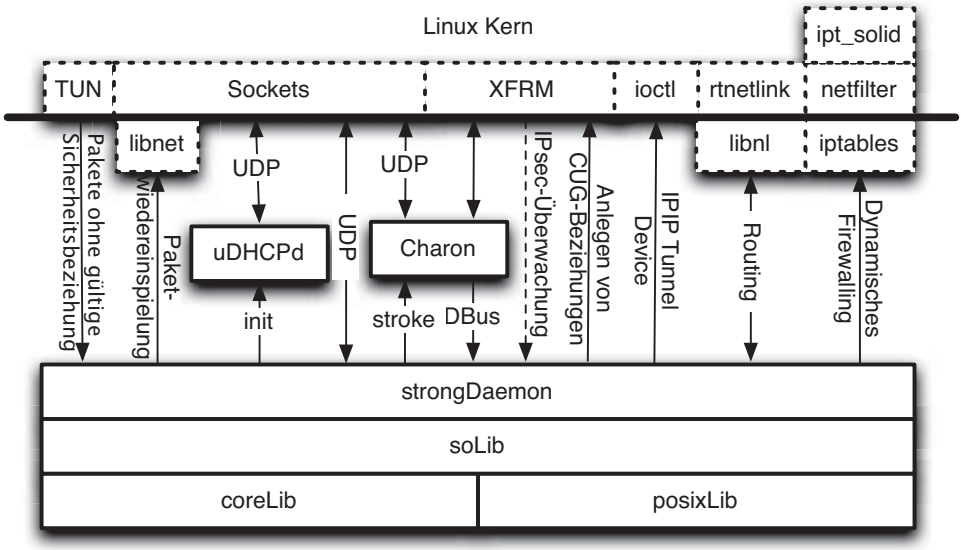


Abbildung 6.2: Software-Architektur zur Realisierung von SOLID in Linux/strongSwan (der Betriebssystemkern ist nur aus zeichnerischen Gründen als oberste Schicht dargestellt)

6.3 Überblick über die Software–Architektur

Die Aufteilung in verschiedene Software-Bibliotheken spiegelt sich auch in der Ableitungshierarchie und den Abhängigkeiten der realisierten Klassen wider. In Abbildung 6.3 sind einige der wichtigsten Zusammenhänge gezeigt. Der zentrale Koordinator eines jeden Overlay-Knotens ist die `Node`-Klasse, welche einfach je nach Einsatzszenario abgeleitet und teilweise unterschiedlich implementiert wird. Darüber

hinaus existieren einige Klassen, deren Objekte in einer 1:1 Beziehung zum `Node` stehen. Diese sind primär zur besseren Strukturierung ausgelagert und setzen die eigentliche Funktionalität wie Bootstrapping, Topologiekontrolle und Routing um. Ein zentraler `EventDispatcher` pro Knoten bewahrt die entstandene hierarchische Struktur vor dem Entstehen zu vieler Querbeziehungen. Interessierte Objekte können sich mithilfe des verwendeten Publish-Subscriber-Systems einfach durch Registrieren bei dem Dispatcher von Ereignissen, wie dem Hinzufügen oder Entfernen von Netzwerkschnittstellen, informieren lassen.

Ein weiteres zentrales Objekt jedes Knotens ist eine Instanz der `SelfPeer`-Klasse; in diesem Objekt werden die zentralen Eigenschaften des Knotens hinterlegt, wie seine IP-Adressen, Vivaldi-Koordinate oder Multicast-Mitgliedschaft. Die Klasse implementiert darüber hinaus das so genannte *Memento*-Entwurfsmuster [GHJV94]. Die entstehenden Objekte lassen sich somit serialisieren, über das Netzwerk übertragen und von der Gegenseite zu `Peer`-Objekten zusammensetzen, welche die Sicht anderer SOLID-Instanzen auf die jeweiligen Knoten repräsentieren. Die übertragenen `Peer`-Objekte enthalten dabei nur relevante Informationen, wie äußere IP-Adressen, rote Netze und die aktuellen Vivaldi-Koordinaten des anderen Knotens.

Jeder Knoten verfügt über drei verschiedene Datenstrukturen zur Verwaltung der Informationen über andere Knoten:

- Das `GossipStorage`-Objekt speichert alle `Peer`-Informationen, die ein Knoten über andere Knoten erhalten hat. Diese Datensätze werden unter anderem verwendet, um die stochastischen Querverbindungen optimal zu platzieren.
- In der `SAStorage`-Struktur werden Informationen über alle aktuell aufgebauten Sicherheitsbeziehungen gehalten. Insbesondere erfolgt hier eine Zuordnung des Verkehrsflusses bezüglich Volumen und Anzahl der Pakete auf die einzelnen VPN-Knoten.
- Die eigentliche Zuordnung von Sicherheitsbeziehungen und Knoteninformationen erfolgt in der `SPEStorage`-Klasse. Objekte der Klasse speichern alle Verbindungen, welche zu anderen Knoten existieren oder aufgebaut werden sollen.

Das Zusammenspiel der vorgestellten Einzelkomponenten wird in den folgenden Abschnitten, anhand der Realisierung der einzelnen in Teilabschnitt 4.1.3 vorgestellten Dienste, erläutert.

6.4 Bootstrapping

In Umgebungen mit normalem Sicherheitsbedarf funktioniert der Bootstrapping-Mechanismus relativ einfach: Der `BootstrapManager` ruft periodisch eine Reihe verschiedener Module auf, welche jeweils eines der in Abschnitt 4.2 vorgestellten Verfahren implementieren. Die ermittelten IP-Adressen können anschließend über ein einfaches Request-Response-Protokoll schnell auf mögliche Erreichbarkeit getestet werden. Je

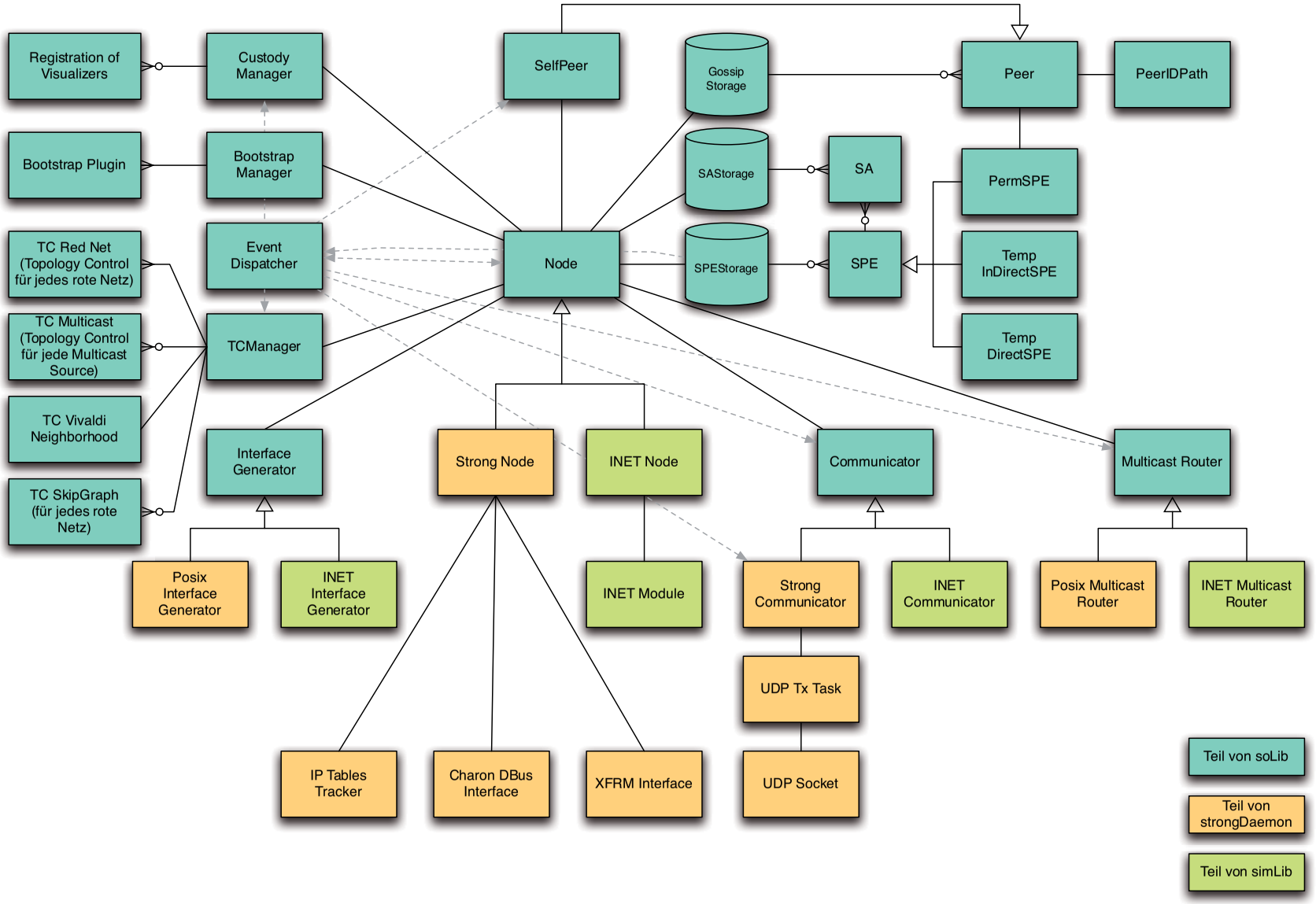


Abbildung 6.3: Vereinfachtes Klassendiagramm des SOLID-Prototypen

nach VPN-Szenario kann das Bootstrapping über eine Selektion der Module an die Umgebung angepasst werden.

Komplexer wird das Verfahren, wenn aus Sicherheitsgründen keine Pakete ohne IPsec-Verschlüsselung in das schwarze Netz gesendet werden dürfen. Da in diesem Fall auch die durch das Bootstrapping-Verfahren übermittelten Pakete zumindest mithilfe statischer Sicherheitsbeziehungen geschützt werden müssen, können komplexere Verfahren wie das Abfragen anderer Overlay-Netze oder DNS-Server nicht länger zum Einsatz kommen. Daher ist SOLID auf das direkte Kontaktieren anderer System des gleichen VPN beschränkt, was durch statisch vorkonfigurierte Adressen, Broad- und Multicast erfolgen kann. Alternativ können Systeme auch versuchen IP-Adressen zu kontaktieren, die bereits von VPN-Knoten benutzt wurden.

Zur Absicherung dieser Nachrichten werden statische IPsec-Beziehungen im Transportmodus und auf Basis symmetrischer Gruppenschlüssel eingerichtet. Um auch Pakete von vormals unbekannten Geräten entgegenzunehmen, muss zusätzlich der von IPsec vorgesehene Schutz vor wiedereingespielten Paketen deaktiviert werden. Dieser wird von SOLID für Bootstrapping-Pakete ersatzweise auf Applikationsebene durch den `BootstrapManager` mittels eines zweistufigen Mechanismus realisiert: Stark verzögert wiedereingespielte Nachrichten werden mittels eines eingefügten Zeitstempels erkannt, der aber aufgrund der lediglich losen Synchronisation der Systemzeiten vergleichsweise ungenau sein kann. Daher wird zusätzlich ein auf *Nonce*-Werten basierender Mechanismus eingesetzt, in dem jeder Bootstrap-Anfrage ein Zufallswert hinzugefügt wird. Dieser Zufallswert muss zum einen in der Antwort enthalten sein, und zum anderen wird keine Anfrage mit einem doppelten Wert akzeptiert. Da je nach angenommener Synchronisation zwischen den VPN-Knoten, der VPN-Größe und dem Bootstrapping-Intervall, möglicherweise eine Vielzahl von Zufallswerten zwischengespeichert werden muss, erfolgt eine Erfassung empfangener Zufallswerte in einem Bloom-Filter. Um den Filter periodisch zu löschen, existieren zwei Bloom-Filter von denen in jeweils einen geschrieben wird. Der inaktive Filter wird noch bis zur maximal tolerierten Zeitdrift abgefragt. Anschließend wird er zurückgesetzt, und die Funktion beider Filter getauscht.

Da das Autokonfigurationsverfahren auch in Umgebungen mit NAT einzusetzen sein soll, muss auch dieser Umstand beim Bootstrapping berücksichtigt werden. Aus diesem Grund wird bei der IPsec-Beziehung NAT-Traversal [HSV+05] aktiviert, sodass ESP-Daten aus Kompatibilitätsgründen zusätzlich in UDP gekapselt werden. Wie in Abbildung 6.4 angedeutet, wird die Quelladresse dabei möglicherweise durch das NAT-Gateway umgeschrieben. Dieser Umstand ist insofern problematisch, als die IPsec-Verarbeitung diese Verwaltungsinformation ohne weitere Benachrichtigung verwirft. Daher öffnet SOLID einen so genannten *RAW*-Socket, der alle eingehenden Pakete mit dem durch NAT-Traversal festgelegtem Ziel-Port 4500 und gültigem Security Parameter Index (SPI) ohne weitere Bearbeitung an die Applikation weiterreicht. So kann nach Empfang des entschlüsselten Paketes eine korrekte ausgehende IPsec-Policy angelegt und die Antwort ausgeliefert werden.

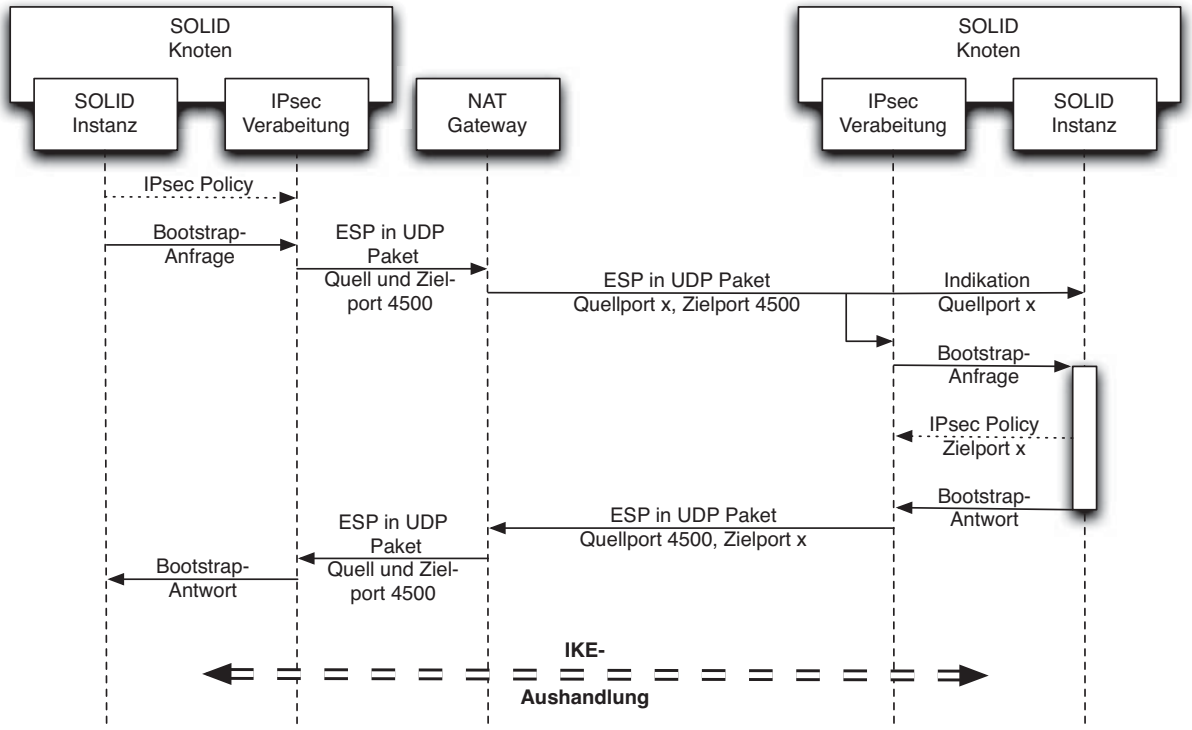


Abbildung 6.4: Ablauf des Bootstrappings in einem NAT-Szenario mit aktivierter Zwangsverschlüsselung

Ein potentielles Problem des Verfahrens sind lediglich unter Umständen eintretende Race-Conditions, falls mehrere Knoten gleichzeitig über dasselbe NAT-Gateway Anfragen versenden. In diesem Fall kann es passieren, dass die Zuordnung von verschlüsselten und unverschlüsselten Paketen nicht vollständig gelingt. Dies ist allerdings zu vernachlässigen, weil das Bootstrapping zum einen periodisch erfolgt. Zum anderen reicht ein korrekt beantwortetes Paket aus, um eine Sicherheitsbeziehung zu einem der Knoten aufzubauen und so alle Knoten hinter dem NAT-Gateway in das globale VPN einzubinden.

Unabhängig davon, ob ein NAT-Gateway verwendet wird, kann ein VPN-Knoten, wenn er eine kompatible Antwort erhalten hat, eine dynamische IPsec-Beziehung zwischen beiden Knoten initiieren, durch die weitere Kommunikationsvorgänge geschützt werden. Dazu wird beim Start von SOLID ein generischer Eintrag in der Sicherheitspolitik erzeugt, der alle eingehenden Sicherheitsbeziehungen für eine bestimmte Zertifizierungsinstanz zulässt.

6.5 Adresszuweisung

Wie in Abschnitt 4.3 bereits herausgearbeitet wurde, müssen die jeweiligen privaten IP-Adressbereiche der VPN-Knoten kryptographisch nachweisbar sein, um innerhalb eines dynamischen Netzes Pakete eindeutig zu authentisieren. Im Fall von herkömmlichen IPsec-VPN werden die gültigen inneren IP-Adressen in Konfigurationsdateien

hinterlegt; ein Mechanismus, der dem Ansatz von Access Control Lists (ACLs) ähnelt. Im Prinzip ist es möglich, dieses Verfahren auch beim Einsatz von Autokonfigurationsverfahren beizubehalten, indem beispielsweise zunächst über eine Sicherheitsbeziehung im Transportmodus Konfigurationsdaten und das Adresszertifikat ausgetauscht werden. Allerdings ist der Aufbau solcher Beziehungen im Fall von NAT nicht möglich und der Aufbau von Sicherheitsbeziehungen allein für Konfigurationszwecke ineffektiv.

Daher werden in SOLID bereits bei der IKE-Aushandlung die von der Gegenseite beanspruchten IP-Adressen anhand des IPsec-Zertifikates überprüft, sodass ein *Capability*-ähnliches Konzept zum Einsatz kommt. Die Umsetzung erfolgt über eine Erweiterung von strongSwan, die aber inzwischen Teil des offiziellen Entwicklungszweiges ist. Dabei werden IP-Adressen in einer ehemals für S-BGP vorgesehenen X.509-Erweiterung [LKS04] abgelegt, die eine hierarchische Vergabe der IP-Adressbereiche ausgehend von einer Wurzel-CA erlaubt. Wie im letzten Abschnitt bereits angedeutet, können die angelegten Sicherheitspolitiken nun einfach den vollständigen IP-Adressbereich erlauben, welcher dann jeweils auf Basis der tatsächlich beanspruchten Adressen sicher eingeschränkt wird.

Die Zuweisung von IP-Adressen innerhalb des roten Netzes erfolgt aus Kompatibilitätsgründen über DHCP, wobei die Adressbereiche dem Zertifikat des SOLID-Gateways entnommen werden.

6.6 Discovery und Topologiekontrolle

Discovery-Vorgänge werden durch Pakete eines Klienten oder vom VPN-Knoten selbst ausgelöst, wenn für sie keine geeignete Sicherheitsbeziehung aufgebaut ist. Wie es für einen typischen Verlauf in Abbildung 6.5 illustriert wird, empfängt der SOLID-Prozess dieses Paket über eine virtuelle Netzwerkschnittstelle und reagiert mit einer Suchnachricht. Zwischensysteme, welche die Suchnachricht erhalten und bisher keine gültige Route zum Ursprungssystem besaßen, vermerken diese und leiten die Nachricht entlang der Overlay-Struktur zum Ziel-Gateway. Dieses beantwortet die Suche, und über die gültige Route wird anschließend eine IKE-Aushandlung vollzogen. Sobald die IPsec-Beziehung aufgebaut wurde, kann das ursprüngliche Paket vom Quell-Gateway wiedereingespielt und durch die IPsec-Beziehung geschützt übertragen werden. Kann kein korrespondierender VPN-Knoten gefunden werden, wird das ursprüngliche Paket verworfen und automatisch eine Fehlernachricht mittels Internet Control Management Protocol (ICMP) generiert. Aus Gründen der Transparenz ist es dabei wichtig, dass der Ablauf des Discovery-Verfahrens innerhalb eines möglichst kurzen Zeitraums geschieht. Daher wird in die eingeführte Verzögerung quantitative zu untersuchen sein.

Die in SOLID implementierten Mechanismen zur Topologiekontrolle sind ähnlich wie beim Bootstrapping modular strukturiert und können je nach Bedarf aktiviert werden. Die Module werden vom Auf- und Abbau relevanter Sicherheitsbeziehungen, sowie

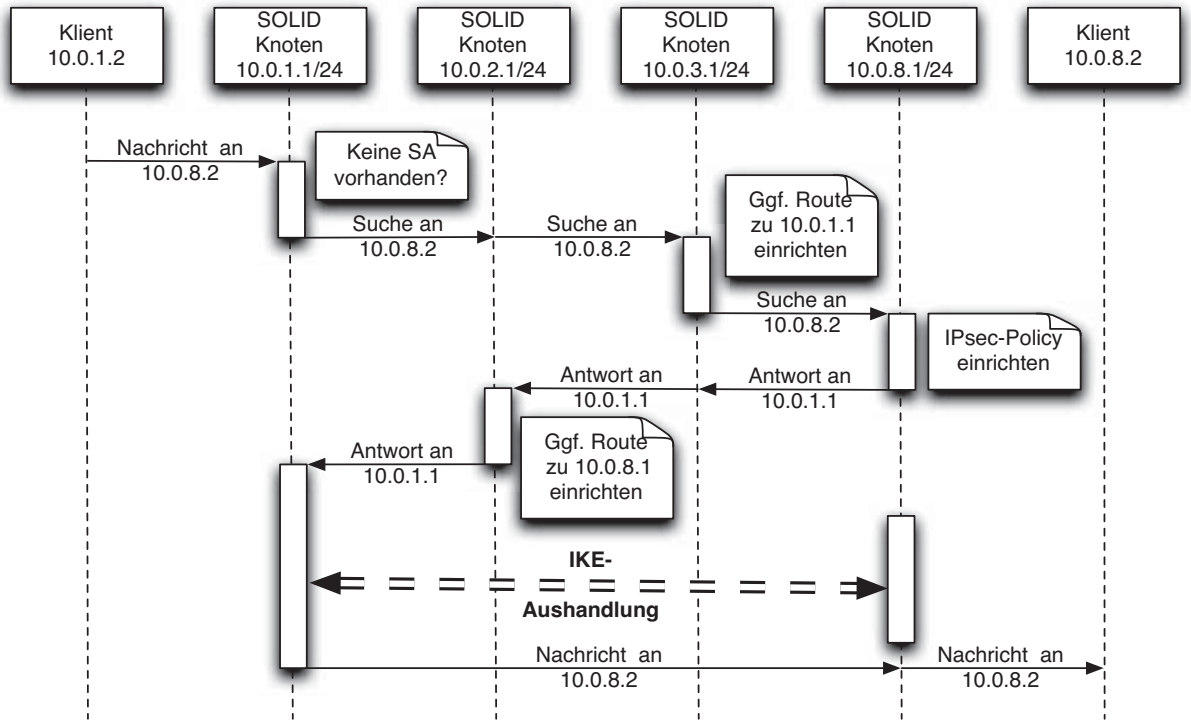


Abbildung 6.5: Beispiel eines Discovery-Ablaufs zum reaktiven Finden von VPN-Gateways

von Änderungen in der Gossip-Datenbank informiert und können so zum einen Verbindungen markieren, die aufrecht erhalten werden sollen, und zum anderen den Aufbau neuer Verbindungen proaktiv initiieren. Ferner werden periodische Suchnachrichten versandt, um beispielsweise Ringnachbarn zu finden. Wird ein passender Knoten gefunden, verläuft der Aufbau einer Sicherheitsbeziehung völlig analog zur Discovery-Phase.

Sind Verbindungen zumindest aus Sicht der Topologiekontrolle nicht länger interessant, werden sie als abbaubar markiert. SOLID nutzt die *XFRM*-Schnittstelle des Linux-Kerns, um festzustellen, inwiefern die zu einer solchen Verbindung korrespondierenden Sicherheitsbeziehungen noch für den Transport von Nutzdaten verwendet werden. Wird eine als abbaubar markierte Verbindung eine Zeit lang, beispielsweise zehn Minuten, nicht verwendet, wird die entsprechende Sicherheitspolitik automatisch gelöscht.

6.7 Routing und Paketweiterleitung

Nachdem durch Topologiekontrolle oder Discovery-Vorgänge Endpunkte für Sicherheitsbeziehungen identifiziert wurden, müssen Nutz- und Verwaltungsdaten sicher und effizient dort hingeleitet werden. Die dazu eingesetzten Techniken werden in den folgenden vier Teilabschnitten erläutert. Nach Details zu den eingesetzten Protokollen, wird auf die Interaktion zwischen Routing-Regeln und -Tabellen zur Weiterleitung von

Unicast-Verkehr eingegangen. Der dritte Teil behandelt die Weiterleitung von Multicast-Verkehr, bevor der Abschnitt mit einem kurzen Überblick über die eingesetzten NAT-Traversal-Techniken schließt.

6.7.1 Eingesetzte Protokollarchitektur

In herkömmlichen IPsec-VPN werden Sicherheitsbeziehungen im Tunnelmodus verwendet, um Pakete zwischen Klienten beziehungsweise zwischen Klienten und VPN-Knoten zu vermitteln. Im Gegensatz dazu wird in SOLID ein flexibleres Verfahren verwendet, welches Ähnlichkeiten zu IIPtran aufweist. Wie in Teilabschnitt 2.3.2 beschrieben, sieht letzteres den Ersatz des IPsec-Tunnelmodus durch die Verwendung von IPIP-Tunneln in IPsec-Transportmodusbeziehungen vor.

SOLID setzt zur Sicherung der Kommunikation zwischen benachbarten VPN-Knoten zunächst BEET-Beziehungen ein, wie sie ebenfalls in Teilabschnitt 2.3.2 beschrieben sind. Diese werden sowohl für den Verwaltungsverkehr als auch für die eigentlichen Nutzdaten verwendet. Sollen Daten zwischen einem Klienten eines VPN-Gateways und einem benachbarten VPN-Knoten, beziehungsweise einem Klienten eines benachbarten VPN-Gateways, ausgetauscht werden, erfolgt zunächst eine Kapselung der Daten durch einen IPIP-Tunnel. Anschließend werden die Pakete über die gleiche BEET-Beziehung versendet.

Sind die kommunizierenden VPN-Knoten nicht benachbart, kommt zur Sicherung — der Argumentation aus Abschnitt 4.5.4 folgend — genau eine geschachtelte Beziehung zur Anwendung. Diese wird im Transportmodus betrieben, da innerhalb des VPN weder NAT noch eine lokalitätstreue Adressvergabe stattfinden. Werden Daten von oder zu einem Klienten eines Gateways übermittelt, wird auch innerhalb der inneren Sicherheitsbeziehung ein weiterer IPIP-Tunnel eingesetzt, sodass sich die exemplarische Tunnelstruktur aus Abbildung 6.6 ergibt.

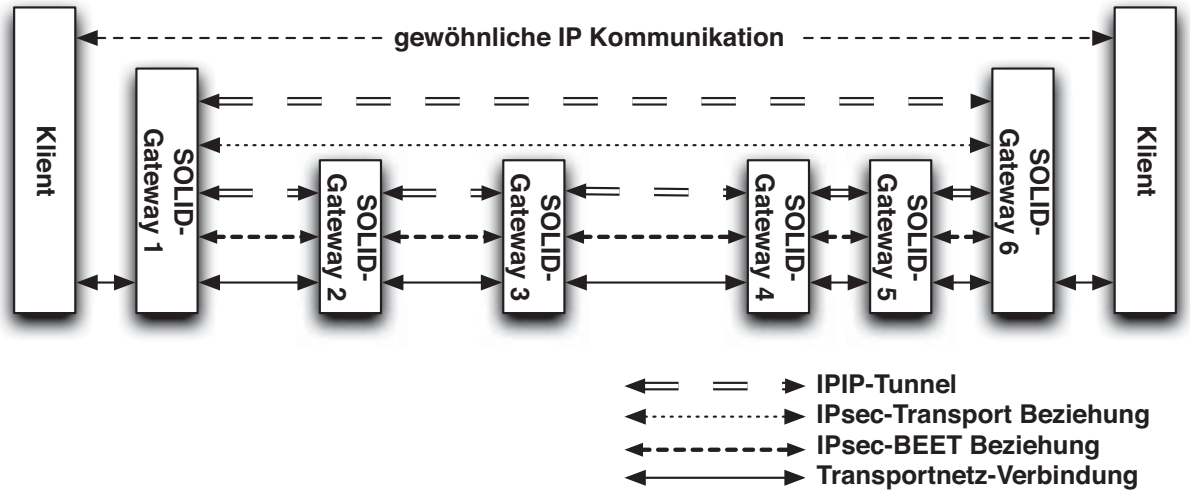


Abbildung 6.6: Verwendete Tunnelstruktur in verschachtelten SOLID-VPN

Eine verbleibende Herausforderung ist die Verknüpfung der IP-Adressen, die innerhalb des IPIP-Tunnels verwendet werden, mit den Adressen für die ein VPN-Knoten oder -Gateway verantwortlich ist. Diese Überprüfung wird normalerweise durch die IPsec-Verarbeitung vorgenommen, und eine Unterlassung würde es kompromittierten VPN-Knoten erlauben, Pakete mit gefälschten IP-Adressen zu versenden. Daher erweitert SOLID die Linux-Firewall um ein eigenes Netfilter-Modul, welches lediglich IPIP-Verkehr für registrierte Sicherheitsbeziehungen erlaubt und eine Zuordnung von Sicherheitskontext und verwendeten roten IP-Adressbereichen vornimmt.

Neben einem effizienten Einsatz der übertragenen Verwaltungsinformationen und der geringen Anzahl benötigter Sicherheitsbeziehungen, ist der Hauptvorteil dieses zunächst relativ komplex anmutenden Verfahrens, dass vor dem IKE-Daemon die Verschachtelung von Sicherheitsbeziehungen verborgen werden kann. Wie alle anderen offenen IKEv2-Daemons für Linux (OpenIKEv2[1], Racoon2[2], IKEv2[3]) beherrscht auch Charon zurzeit die Aushandlung verschachtelter Sicherheitsbeziehungen nicht, da IKE-Pakete von der normalen IPsec-Verarbeitung grundsätzlich ausgeschlossen sind, und so nicht durch einen äußeren IPsec-Tunnel versendet werden können. Weiterhin erlaubt die Nutzung von IPIP-Tunneln die Nutzung einer Sicherheitsbeziehung, auch wenn sich eine variable Anzahl nicht zusammenhängender roter Netze hinter einem VPN-Gateway befinden, oder VPN-Knoten über eine variable Anzahl interner IP-Adressen verfügen können.

6.7.2 Routing–Regeln und –Tabellen

Um Nutz- und Verwaltungsdaten in die jeweiligen Tunnel zu kapseln, werden in SOLID ein komplexer Routing-Regelsatz und drei unterschiedliche Routing-Tabellen verwendet (siehe Übersicht in Abbildung 6.7). Die Einträge der ersten Tabelle (ID 150) enthalten einfache Regeln, um Pakete auf eine virtuelle Netzwerkschnittstelle weiterzuleiten, die für den äußeren Tunnel verantwortlich ist. Da die Tabelle lediglich für IPsec-geschützten Verkehr verwendet werden darf, können Einträge hier aufgrund von Routing-Entscheidungen des SOLID-Prozesses beliebig verändert werden, ohne das ein Verlust von Datenvertraulichkeit, –integrität oder –authentizität zu erwarten ist. Die zweite Tabelle (ID 151) dient der Kapselung von Nutzdatenverkehr sowohl in äußere als auch innere Tunnel und ist damit im Gegensatz zu Tabelle 150 in Bezug auf Vertraulichkeit sicherheitsrelevant. Routing-Einträge werden hier lediglich angelegt, wenn eine gültige Sicherheitsbeziehung zum korrespondierenden VPN-Knoten besteht. In verschachtelten VPN-Topologien kommt unter Umständen noch eine dritte Tabelle zum Einsatz, welche die Weiterleitung zwischen VPN-Knoten regelt, deren schwarze Adressen sich innerhalb desselben roten Netzes befinden. In diesem Fall muss das Routing nicht über das SOLID-Gateway erfolgen, welches für das Netz verantwortlich ist, sondern es können in Übereinkunft mit diesem direkte

[1] `http://openikev2.sourceforge.net`
[2] `http://www.racoon2.wide.ad.jp`
[3] `http://ikev2.zemris.fer.hr`

Sicherheitsbeziehungen eingerichtet werden. Das Routing für diese Beziehungen erfolgt in Tabelle 152.

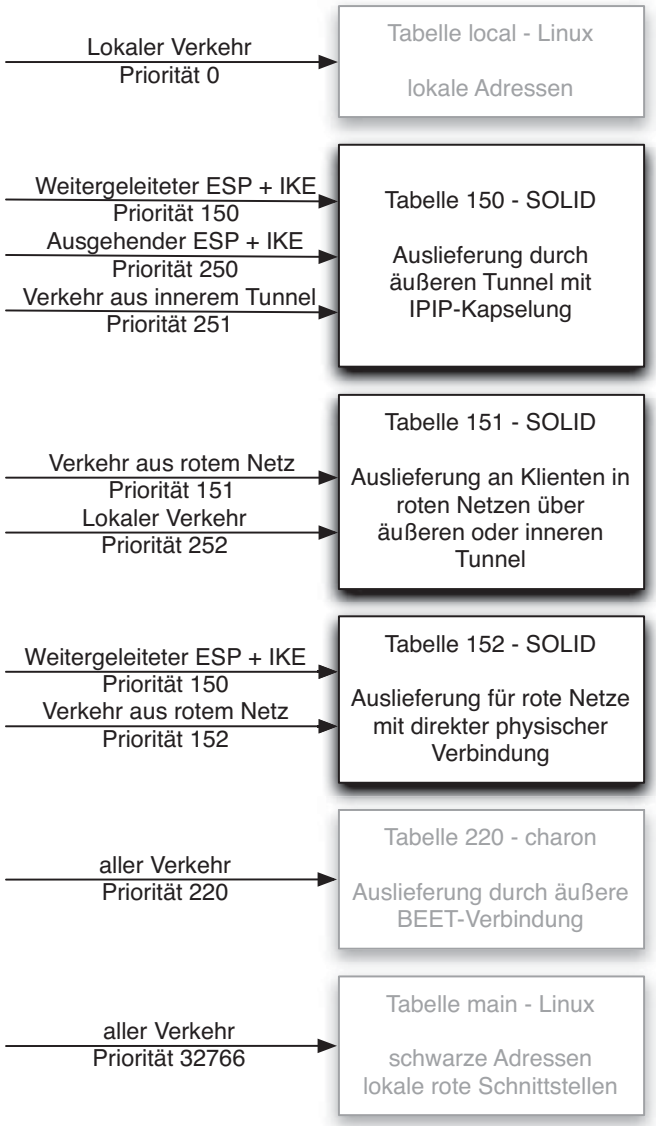


Abbildung 6.7: Von SOLID verwaltete Routing-Tabellen und –Regeln

Das Zusammenspiel der Routing-Regeln wird im Folgenden für einige Verkehrstypen im Detail erläutert:

IKE-Verkehr: Für direkte Sicherheitsbeziehungen greift keine der angelegten Routing-Tabellen, da diese ausschließlich innere Adressen enthalten. Somit erfolgt eine Auslieferung über die *main*-Tabelle des Betriebssystems, welche mit 32766 die niedrigste Priorität aufweist. Bei indirekten Sicherheitsbeziehungen erfolgt über Tabelle 150 zunächst eine Kapselung in einen IPIP-Tunnel, zu einem von SOLID spezifizierten nächsten Knoten. Nach der Kapselung entsteht ein Paket für welches die Routing-Regel nicht noch einmal zutrifft, da es von der virtuellen IPIP-Netzwerkschnittstelle aus versendet wird. Es existiert für das neue Paket jedoch eine direkte Sicherheitsbeziehung und damit ein von Charon angelegter Routing-Eintrag in Tabelle 220. Dieser hat eine höhere Priorität als die Regel für eine weitere Kapselung, und das Paket wird an eine physische Netzschnittstelle ausgeliefert. Die IPsec-Verarbeitung schreibt nun auf Basis des BEET-Modus die innere Adresse in eine externe um, und eine reguläre Auslieferung kann erfolgen. Zwischensysteme leiten so gekapselten Verkehr nach einer Entschlüsselung des äußeren Tunnels auf Basis der Einträge von Tabelle 150 bis zum Zielsystem.

SOLID-Verkehr: Die von SOLID erzeugten Verwaltungsdaten werden im einfachsten Fall von den BEET-Regeln aus Tabelle 220 erfasst und direkt ausgeliefert. Da SOLID-Prozesse auch im indirekten Fall immer nur mit anderen VPN-Knoten kommunizieren, trifft andernfalls automatisch eine IPsec-Regel der inneren Transportmodusbeziehung zu und für das entstandene ESP-Paket wird über Tabelle 150 ein korrespondierender

IPIP-Tunnel zur weiteren Kapselung gesucht. Auslieferung und Weiterleitung erfolgen analog zum IKE-Verkehr.

Nutzdaten: Wie bereits angedeutet, erfolgt die Auslieferung von Paketen roter Netze ausschließlich über IPIP-Tunnel, die Tabelle 151 entnommen werden. Im direkten Fall wird nach der Kapselung eine BEET-Regel in Tabelle 220 gefunden, und das entstandene Paket ausgeliefert. Indirekt übermittelte Pakete werden nach der ersten Kapselung automatisch im IPsec-Transportmodus verschlüsselt. Anschließend erfolgt ein weiterer Durchlauf durch Tabelle 150, welcher die Konfiguration des äußeren Tunnels entnommen wird, bevor über Tabelle 220 eine Auslieferung des BEET-Paketes initiiert wird.

Die Routing-Regeln greifen in der beschriebenen Weise ineinander, um mit einer minimalen Anzahl an Routing-Einträgen die relativ große Anzahl verschiedener Flüsse abzudecken. Würden mehr Tabellen mit teilweise redundanten Einträgen angelegt werden, könnten die voneinander abhängigen Regeln vereinfacht werden. Allerdings ist in diesem Fall ein Effizienzverlust zu erwarten, da Routing-Caches schlechter ausgenutzt würden.

6.7.3 Weiterleitung von Multicast–Verkehr

Durch die Unterstützung von IPIP-Tunneln zur Realisierung von Unicast-Weiterleitungen wird auch die Implementierung des ALM-Ansatzes stark vereinfacht. Der SOLID-Daemon kann durch sie einfach die Multicast-Router-Schnittstelle zum Linux-Kern verwenden, um je nach Overlay-Topologie Multicast-Daten auf eine reale Schnittstelle zu seinen roten Netzen oder über eine virtuelle IPIP-Schnittstelle zu Kindknoten im Verteilbaum weiterzuleiten. Wie in Abschnitt 4.7 gefordert, spielt es bei letzterem keine Rolle, ob die zugrunde liegenden Sicherheitsbeziehungen direkt oder indirekt sind. Eingebettet in ein Beispielszenario ergibt sich die Architektur aus Abbildung 6.8.

Aus Gründen der Transparenz nutzen SOLID-Knoten auch Anwendungsprozessen gegenüber eine standardisierte Schnittstelle: Alle Gateways mit aktivierter Multicast-Funktionalität annoncieren sich in ihren roten Netzen periodisch über Internet Group Management Protocol (IGMP) und treten beim Erhalt gültiger Abonnements dem entsprechenden Verteilbaum bei. Lediglich der Wurzelknoten muss aus Sicherheitsgründen einzelne Multicast-Quellen explizit aktivieren.

6.7.4 NAT–Traversal mit IKE–Mediation

Unabhängig davon ob Multi- oder Unicast-Daten transportiert werden sollen, können bei der Weiterleitung im Transportnetz NAT-Gateways die IP-Adressen des IPsec-geschützten Verkehrs umschreiben. Werden ein oder mehrere NAT-Gateways im Rahmen der IKEv2-Aushandlung identifiziert, kapseln die erstellten Sicherheitsbeziehungen ESP-Verkehr automatisch in UDP-Paketen, sodass die gewöhnlichen NAT-

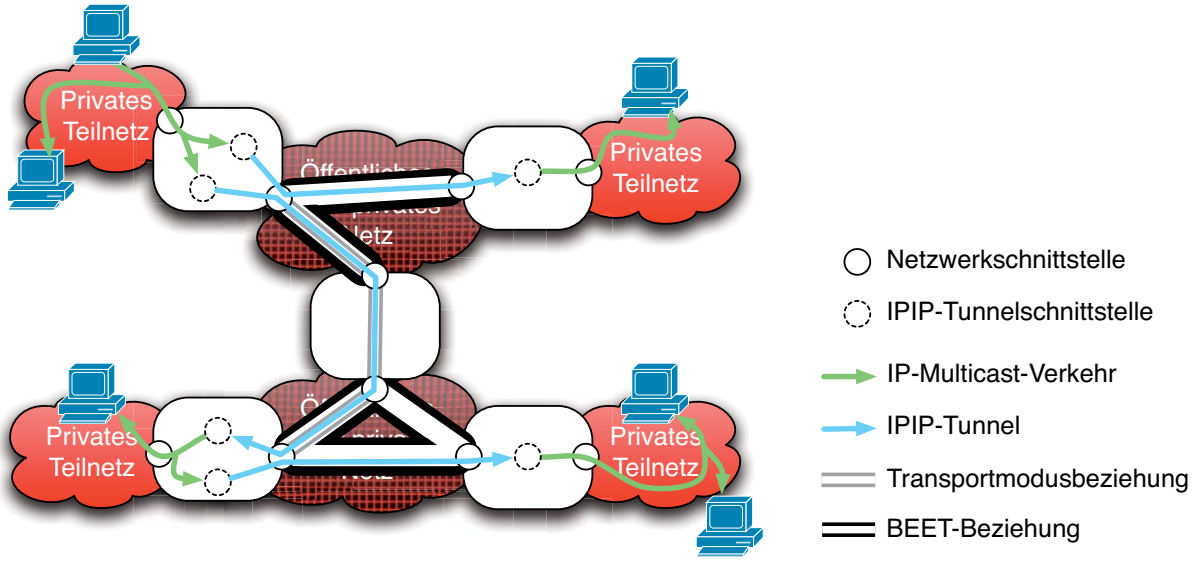


Abbildung 6.8: Beispiel für die Interaktion virtueller und privater Netzwerkschnittstellen bei der Multicast-Weiterleitung

Techniken weiter funktionieren. Durch den Einsatz des BEET-Modus entstehen in diesem Szenario auch keine Adressierungsprobleme, selbst wenn sich mehrere VPN-Knoten hinter demselben NAT-Gateway befinden oder wenn VPN-Knoten sich hinter unterschiedlichen Gateways befinden, aber dieselben privaten IP-Adressen verwenden.

Zur direkten Weiterleitung von Paketen zwischen VPN-Knoten, welche sich beide hinter unterschiedlichen NAT-Gateways befinden, kommt eine Besonderheit zum Tragen: SOLID setzt strongSwans IKE-Mediation-Erweiterung [Bru08] ein, um auch zwischen diesen Knoten mittels Hole Punching direkte Sicherheitsbeziehungen zu etablieren.

Stellen zwei SOLID-Knoten fest, dass sie sich beide hinter unterschiedlichen NAT-Gateways befinden, kontaktiert der Knoten mit der niedrigeren Kennung den Knoten, welcher außerhalb des NAT aufgrund des SOLID-Protokolls die indirekte Weiterleitung übernommen hat. Dieser informiert anschließend den Knoten mit der höheren Kennung über die äußere IP-Adresse des NAT-Gateways und den korrespondierenden IKE-Port des initiierenden Knoten. Daraufhin können beide Knoten eine direkte Sicherheitsbeziehung aufbauen und sind anschließend unabhängig vom weiterleitenden Knoten.

6.8 Fazit

Insgesamt wurde im Rahmen der Arbeit ein komplexer Prototyp entwickelt, welcher zum einen zur Evaluierung von Effekten in großen Netzen simulativ untersucht werden kann. Zum anderen kann er in einer strongSwan/Linux-Umgebung IPsec-

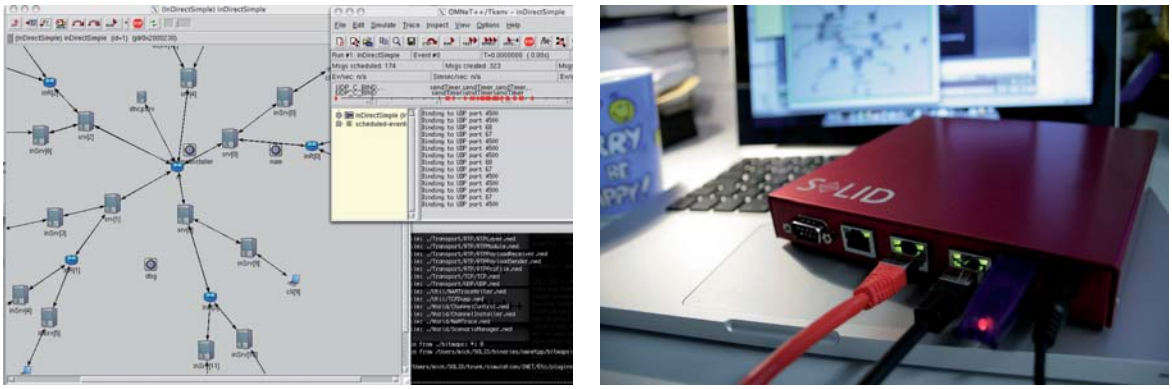


Abbildung 6.9: Einsatzumgebunden von SOLID: innerhalb eines Netzwerksimulators (links) und in Prototyp (rechts)

Infrastrukturen automatisch konfigurieren. Dabei wird aus Effizienz- und Sicherheitsgründen die IPsec-Verarbeitung weiterhin durch den unmodifizierten Betriebssystemkern und einen minimal angepassten Charon-Daemon vorgenommen. Wie in Abbildung 6.9 illustriert, wurde der Prototyp dabei auch auf eingebetteten Systemen installiert und getestet, sodass funktionsfähige SOLID-Gateways untersucht werden können.

Um die Sicherheit gegenüber einer herkömmlichen manuellen Konfiguration nicht zu schwächen, überprüft der Prototyp die angegebenen IP-Adressen anhand der vorgelegten Zertifikate. Die korrespondierenden privaten Schlüssel können dabei, ebenfalls aus Gründen der Sicherheit, durch Smartcards geschützt sein. Durch den Verzicht auf IPsec-Tunnel und den Einsatz von IPIP-Tunneln in Kombination mit Transport- und BEET-Modusbeziehungen erlaubt der Ansatz ferner:

- den Aufbau verschachtelter Tunnel ohne erhebliche Erweiterung und den damit verbundenen Komplexitätszuwachs gängiger IKE-Daemons,
- die Verwendung der Multicast-Routing-Implementierung des Linux-Kerns
- und Effizienzvorteile beim Wechsel des nächsten Hops bei geschachtelten Sicherheitsbeziehungen, da diese nicht neu ausgehandelt werden müssen.
- Weiterhin können in Zukunft so variable rote Netzbereiche unterstützt werden,
- und es wird die Nutzung des flexiblen Linux-Routing für Nutzdatenströme ermöglicht, sodass beispielsweise QoS, alternative Routen oder Multipath-Routing einfacher implementiert werden können.

Die versendeten Pakete entsprechen syntaktisch denen des IPsec-Tunnelmodus. Um sie auch semantisch äquivalent sicher zu machen, wird neben dem BEET-Modus, zur klaren Unterscheidung innerer und äußerer IP-Adressen, ein Firewall-Modul eingesetzt, welches die inneren IPIP-Tunnel mit den ausgehandelten IP-Adressbereichen verknüpft.

Gegenüber Anwendungsprozessen und Klienten in den roten Netzen treten SOLID-Knoten, nicht zuletzt durch das Bereitstellen eines DHCP-Servers und das Annoncieren von IGMP-Nachrichten, wie normale Router auf.

Wo du sicher bist, setze Fragezeichen.

(Wieslaw Brudzinski)

Kapitel 7

Evaluierung & Diskussion

Die Evaluierung der vorgestellten Konzepte und Implementierung erfolgt in zwei Schritten. Zunächst wird die Umsetzung der in Abschnitt 3.1 spezifizierten Anforderungen qualitativ diskutiert. Einige ausgewählte Fragestellungen werden darüber hinaus in Experimenten näher untersucht, deren Anordnung und Resultate im zweiten Teil dieses Kapitels beschrieben werden.

7.1 Qualitative Diskussion

Die in diesem Abschnitt vorgenommene Bewertung gliedert sich strukturell analog zum Anforderungskapitel, und geht somit zunächst auf funktionale Anforderungen ein, bevor nichtfunktionale Anforderungen und insbesondere die Sicherheit des Ansatzes diskutiert werden.

7.1.1 Umsetzung funktionaler Anforderungen

In Bezug auf die aufgestellten funktionalen Anforderungen, stellen sich SOLID's Eigenschaften wie folgt dar:

1. **Selbstkonfiguration:** Die Knoten eines VPN benötigen lediglich eine minimale individuelle Konfiguration. Insbesondere in Bezug auf die nun flexible Overlay-Topologie sind keine individuellen Einstellungen nötig. Konfiguriert werden müssen lediglich:

    - **Kryptographische Parameter:** Jeder Teilnehmer eines SOLID-VPN muss ein eigenes Zertifikat mit korrespondierendem privaten Schlüssel und CA-Zertifikat besitzen, um seine Identität zweifelsfrei nachweisen zu können. Durch die Einbettung von IP-Adressbereichen in das Zertifikat, kann mittels Auswertung des eigenen Zertifikates, der eigene rote Netzbereich automatisch konfiguriert werden. Zum Schutz des Bootstrap-Vorgangs wird zusätzlich ein symmetrischer Gruppenschlüssel verwendet, der dem Knoten bekannt sein muss.

- **Zuordnung roter und schwarzer Schnittstellen:** Bei VPN-Gateways muss für jede Netzwerkschnittstelle eine Zuordnung erfolgen, ob sie zu einem vertrauenswürdigen Netz oder einem Uplink führt. Unter Umständen kann dies pro Gerätetyp statisch durch das verwendete Betriebssystem über die Zuordnung von eindeutigen Bezeichnungen auf physische Schnittstellen geschehen.

- **Bootstrapping:** Wie in Abschnitt 4.2 bereits erläutert, muss das Finden eines ersten VPN-Knotens beim initialen Beitritt je nach Einsatzszenario erfolgen und ist somit zumindest zum Teil individuell zu konfigurieren. In der Regel müssen dabei allerdings lediglich für jedes VPN eine Methode und gegebenenfalls mehrere IP-Adressen eingestellt werden.

- **Internet-Schnittstelle:** Die Internet-Anbindung der Teilnehmer ist zwar nicht Fokus dieser Arbeit, muss aber in Realsystemen möglicherweise individuell konfiguriert werden. Soll auf diese Konfiguration ebenfalls verzichtet werden, muss angenommen werden, dass auf Autokonfigurationsmechanismen der Zugangsanbieter wie TR-069 [BS04] oder DHCP zurückgegriffen werden kann.

Keine der erwähnten Konfigurationsdomänen kann ohne weitere Annahmen oder veränderte Anforderungen weiter vereinfacht werden.

2. **Konfiguration in gebräuchlichen Transportnetzen:** In Bezug auf die unterstützten Topologien von VPN und Transportnetzen werden die folgenden Szenarien berücksichtigt:

   a) **Gateway-Funktionalität:** SOLID kann sowohl VPN-Teilnehmer direkt einbinden, als auch dies transparent über ein Sicherheits-Gateway realisieren.

   b) **Verschachtelte VPN-Strukturen:** Werden VPN-Teilnehmer hinter anderen angeschlossen, so kann SOLID diese trotzdem in das VPN einbinden. Im Fall von baumartig verbundenen Teilnetzen, werden dabei auch optimale Pfade gefunden.

   c) **Verwendung von Unicast-Kommunikation im Transportnetz:** Alle von SOLID ausgehandelten IPsec-Beziehungen verwenden ausschließlich Unicast. Multicast-Verkehr innerhalb des VPN wird automatisch durch den ALM-Ansatz in Unicast-Paketen gekapselt. Lediglich bei Bootstrapping-Komponenten ist der Einsatz von Gruppenkommunikation zum Auffinden anderer SOLID-Knoten im gleichen Teilnetz vorgesehen. Ist dies nicht möglich, kann auf andere Mechanismen zurückgegriffen werden.

   d) **Umgang mit NAT:** Um auch in Gegenwart von NAT-Gateways ein funktionsfähiges und effizientes VPN aufzubauen, werden eine Reihe implementierungstechnische Maßnahmen durchgeführt. Durch eine UDP-Kapselung können die ESP-Pakete umgeschrieben werden, und der BEET-Modus ermöglicht eine Unterscheidung von VPN-Knoten, die sich hinter dem gleichen NAT-Gateway befinden. Durch die explizite Berücksichtigung von

IKE-Mediation können darüber hinaus auch direkte Sicherheitsbeziehungen zwischen von NAT betroffenen Knoten ausgehandelt werden.

In Internet-ähnlichen Umgebungen kann SOLID somit ohne weitere Voraussetzungen eingesetzt werden.

3. **Verwendung privater Adressbereiche:** Die IP-Adressen innerhalb des VPN sind weitestgehend unabhängig von denen auf der schwarzen Seite. Da allerdings Routing-Entscheidungen zu treffen sind, dürfen die Adressen außerhalb des VPN nicht mit den inneren kollidieren. Andernfalls sind die kollidierenden Adressen im schwarzen Bereich durch SOLID nicht kontaktierbar.

4. **Multicast-Unterstützung innerhalb des VPN:** Um innerhalb des VPN Gruppenkommunikation durchzuführen, wird der in Abschnitt 4.7 vorgestellte ALM-Ansatz realisiert. Die direkte Umsetzung von Multicast-Verkehr auf Transportnetz-Multicast ist zurzeit nicht vorgesehen, auch aufgrund der damit verbundenen schlechteren Sicherheitseigenschaften von Gruppenkommunikationsbeziehungen.

5. **Transparenz:** SOLID-VPN sind von gewöhnlichen Netzen kaum zu unterscheiden, da sich Gateways auf der roten Seite durch DHCP-, IGMP-Funktionalität und IP-Weiterleitung analog zu herkömmlichen Routern verhalten. Durch den reaktiven Aufbau von Sicherheitsbeziehungen kann das erste Paket allerdings nur leicht verzögert ausgeliefert werden. Der implementierte Wiedereinspielmechanismus sorgt aber für einen schnellstmöglichen Transfer. Quantitative Untersuchungen zur genauen Verzögerung sind im zweiten Teil des Kapitels ausgeführt (siehe Teilabschnitt 7.3.1). Durch die IPsec-Tunnel sinkt weiterhin die MTU, welche aber über DHCP annonciert wird und zusätzlich durch Path MTU Discovery zu detektieren ist.

7.1.2 Nichtfunktionale Eigenschaften

In Bezug auf die in Teilabschnitt 3.1.2 vorgestellten nichtfunktionalen Anforderungen werden einige generelle Aussagen im Folgenden qualitativ getroffen. Für eine tiefergehende quantitative Betrachtung sei auf die experimentelle Evaluierung im zweiten Teil des Kapitels verwiesen.

1. **Skalierbarkeit:** Alle in SOLID implementieren Verfahren besitzen in Bezug auf die Anzahl der Knoten im VPN einen konstanten oder logarithmischen Speicher- und Rechenaufwand. Einzige Ausnahme bildet der maximal lineare Zuwachs des Speicheraufwandes für die Stichproben über die roten Netzwerkadressen, welcher aber selbst bei sehr großen VPN aufgrund der geringen Gesamtgröße zu vernachlässigen ist.

   Die Anzahl der von jedem Knoten proaktiv aufgebauten Netzwerkverbindungen folgt in den erwarteten Szenarien der Funktion $\mathcal{O}(\log(n) + m)$, wobei n für die Gesamtanzahl der Knoten des VPN steht, und m die Knoten repräsentiert, welche sich im roten Netz des Knotens befinden. Allerdings kann es aufgrund

von Transportnetzbedingungen im ungünstigsten Fall trotzdem passieren, dass ein einzelner Knoten zu allen anderen Knoten eine Sicherheitsbeziehung aufbaut, beispielsweise bei einer physischen Sterntopologie. Gleiches gilt wenn solche Topologien aufgrund von Zonenbeschränkungen erzwungen werden. Aus administrativer Sicht sind solche Szenarien allerdings bereits aus Verfügbarkeitsgründen zu vermeiden, da SOLID dann zu einem zentralen Verfahren degeneriert. Darüber hinaus können überproportional viele Verbindungen zu einem Knoten aufgebaut werden, wenn dieser durch das Netzwerkkoordinatensystem bevorzugt als Nachbar identifiziert wird. Auftreten kann dies beispielsweise, falls ein Knoten direkt am Kernnetz angeschlossen ist, und alle anderen Knoten über ein stark verzögertes Zugriffsnetz angebunden sind. Werden solche Szenarien erwartet, muss SOLID zur Realisierung von Skalierbarkeit so konfiguriert werden, dass die Auslastung in Bezug auf die aktuell aufgebauten Sicherheitsbeziehungen in die Berechnung der Höhenkoordinate einfließt. Wie in Teilabschnitt 4.6.4 angedeutet, würde sich darüber ein lastbalancierender Effekt einstellen.

2. **Robustheit:** Sofern durch die Transportnetztopologie keine dedizierten Ausfallpunkte physisch vorgegeben werden, kann SOLID durch die vollständige Verteilung des Konfigurationsmechanismus — beispielsweise im Gegensatz zu den Hub-and-Spoke-Systemen — partielle Knoten- und Verbindungsausfälle sehr gut tolerieren. Allerdings bleiben diese im Vergleich zu vollständig vermaschten Szenarien aufgrund der in SOLID stattfindenden indirekten Kommunikation auch nicht völlig lokal. Kommt es zu einem Ausfall, gehen auch weitergeleitete Pakete bis zur Behebung des Ausfalls oder der automatischen Umgehung des Problems verloren. Letzteres erfolgt sobald durch das Ausbleiben der regelmäßig übertragenen Heartbeat-Nachrichten der Fehler detektiert wurde, und kann in diesem Fall — im Gegensatz zu vollvermaschten, statischen Szenarien — wieder eine virtuelle Konnektivität herstellen.

   Ähnlich wie bei der Fehlerfortpflanzung verhält es sich auch bei potentiellen Netzpartitionierungen: Die im Gegensatz zu vollvermaschten Szenarien geringere Anzahl proaktiv aufgebauter Sicherheitsbeziehungen begünstigt prinzipiell das Entstehen solcher Situationen. Allerdings kann auch hier durch den Aufbau neuer Sicherheitsbeziehungen dynamisch reagiert werden, und die in Teilabschnitt 4.6.3 vorgestellten Mechanismen garantieren eine anschließende Konvergenz des Overlays.

   Gegenüber zentralen und dezentralen Strukturen ist somit eine stark verbesserte Robustheit zu erwarten. Die im Vergleich zu vollvermaschten Szenarien geringere Anzahl an Sicherheitsbeziehungen begünstigt die Entstehung von Netzpartitionierungen und Fehlerausbreitungen prinzipbedingt. Bei kleineren VPN ohne Skalierbarkeitsanforderungen wäre es daher denkbar die Robustheit zu erhöhen, indem ein modifiziertes SOLID nach dem Aufbau der Ringstruktur proaktiv Sicherheitsbeziehungen zu allen anderen, direkt erreichbaren Knoten aufbaut.

3. **Agilität:** Applikationen und Klienten erhalten in SOLID bereits aus Sicherheitsgründen rote Adressen, die nicht vom Standort abhängig sind, und es genügt

somit einen Knoten nach jedem Wechsel der äußeren Adresse in das VPN zu reintegrieren.

Dabei ist in SOLID durch die Verwendung der logischen Ringstruktur ein Aufbau vergleichsweise weniger Sicherheitsbeziehungen erforderlich: Im allgemeinen Fall genügt es die Beziehungen zu den zwei Nachbarknoten erneut auszuhandeln, um innerhalb des VPN durch andere Knoten gefunden zu werden. Können die Nachbarn nur indirekt erreicht werden, muss vorher noch eine zusätzliche Beziehung zu einem direkt erreichbaren Knoten ausgehandelt werden, welcher dann die Weiterleitung übernimmt. Der verwendete Routing-Algorithmus erlaubt zudem eine Konvergenz, wenn eine begrenzte Knotenzahl nur über eine Verbindung verfügt. Voraussetzung ist lediglich, dass die Beziehung zu dem Ringnachbarn aufgebaut wird, der bezogen auf die Overlay-Metrik näher ist. Dieser muss zudem korrekt in der Overlay-Struktur registriert sein. Der Algorithmus leitet per Definition alle Anfragen zu diesem Ringnachbarn weiter, welcher sie daraufhin ausliefert.

Der selbstinitiierte Aufbau von Sicherheitsbeziehungen ist möglich, sobald eine Verbindung zu einem beliebigen Knoten aufgebaut wurde, der korrekt in das SOLID-VPN eingebunden ist und somit Anfragen weiterleiten kann. Sicherheitsbeziehungen zur Optimierung des Overlays — beispielsweise für Querverbindungen oder zu in Bezug auf Vivaldi nahen Knoten — können nach Bedarf später aufgebaut werden.

4. **Effizienz:** In Bezug auf die Effizienz formulierte Ziele sind: eine Minimierung der Anzahl an Sicherheitsbeziehungen, der Pfadlängen und der Anzahl an ausgetauschten Nachrichten.

   Wie auch im Abschnitt zur Skalierbarkeit bereits dargelegt, beträgt die Anzahl proaktiv aufgebauter Sicherheitsbeziehungen pro Knoten normalerweise $\mathcal{O}(\log(n) + m)$. Inwieweit diese Anzahl tatsächlich erreicht wird, hängt von der jeweiligen Transportnetzstruktur und den ausgetauschten Nutzdaten ab. Gegenüber vollvermaschten Szenarien werden damit, sowohl insgesamt als auch bezogen auf jeden Knoten, in den meisten Fällen weniger Beziehungen aufgebaut. Hub-and-Spoke-Topologien benötigen zwar insgesamt lediglich $\mathcal{O}(n)$ Beziehungen, da aber alle am zentralen Knoten terminieren, wird dieser unter Umständen enorm belastet.

   Die Länge der Pfade durch das VPN wird, wie in Abschnitt 4.5.3 gezeigt, in gängigen Szenarien durch die vorgenommenen Optimierungen minimiert. Bei Störungen können durch weitere Optimierungsmechanismen Verzögerungszeiten weitestgehend minimiert werden. Details zu quantitativen Ergebnissen werden in Teilabschnitt 7.4.4 vorgestellt.

   Die Anzahl der ausgetauschten Nachrichten hängt ebenfalls primär von den Transportnetzstrukturen ab. Eine konvergierte Overlay-Struktur mit maximaler Verschachtelungstiefe d vorausgesetzt, kann aber pro Knoten von $\mathcal{O}(d \log n)$ Nachrichten pro Zeiteinheit ausgegangen werden. Das Einfügen und Entfernen eines Knotens erfolgt insgesamt mit einem Aufwand von $\mathcal{O}(d \log^2 n)$ Nachrichten,

da sich im ungünstigsten Fall logarithmisch viele Knoten durch eine logarithmische Anzahl von Nachrichten neue Querverbindungen aufbauen müssen. Im Gegensatz zu Link-State oder Distanzvektorverfahren, bedingt der Auf- und Abbau einzelner Sicherheitsbeziehungen per se keine weitere Kommunikation des Zustandes.

Nach einer weiteren Diskussion von SOLID in Bezug auf die Sicherheitseigenschaften, werden die erreichten nichtfunktionalen Ziele weitergehend quantitativ untersucht.

7.1.3 Erfüllung der Sicherheits– und Verfügbarkeitsanforderungen

Zur Bewertung der Angriffsresistenz von SOLID wird das in Teilabschnitt 4.1.4 vorgestellte Angreifermodell verwendet. Dabei werden für jeden Angreifertyp die in Abschnitt 2.1 beschriebenen Angriffe mit steigendem Grad an Einflussnahme betrachtet: Abhören und Verkehrsflussanalyse, Gray- und Blackhole-Angriffe, Paketverzögerung, das Wiedereinspielen von Paketen und das Senden gefälschter Pakete.

Externe Angreifer

Durch den erzwungenen IPsec-Schutz des gesamten Datenverkehrs und fast aller Verwaltungsdaten sind die Möglichkeiten potentieller externer Angriffe vergleichsweise gering. Im Einzelnen sind folgende Ziele zu erreichen:

1. **Abhören und Verkehrsflussanalyse:** Bis auf die Bootstrap-Nachrichten, deren Inhalt nicht geschützt werden muss, werden alle Nutzdaten und Verwaltungsnachrichten durch IPsec zwangsverschlüsselt. Das direkte Ermitteln der übertragenen Informationen kann somit als nicht realisierbar angenommen werden.

   Durch das Beobachten möglichst vieler VPN-Knoten und deren Sicherheitsbeziehungen kann aber möglicherweise Rückschluss auf die Intensität und Art der Datenflüsse gezogen werden. Dies verschlechtert jedoch nicht die Situation im Vergleich zu herkömmlichen VPN. Im Gegenteil: durch die Vermittlung indirekten Verkehrs können unter Umständen die eigentlichen Kommunikationsbeziehungen verschleiert werden. Weitergehende Schutzmechanismen erfordern das Senden von künstlichem Cover Traffic, das Einführen von Paddings, den häufigen Wechsel von Routen, und die Einhaltung fester Sendetakte. Diese Methoden sollen im Rahmen dieser Arbeit, wie bereits in den Anforderungen abgegrenzt, nicht weiter verfolgt werden.

   Wird innerhalb des VPN nur wenig Nutzdatenverkehr transportiert, kann ein Angreifer somit die proaktiv aufgebaute Overlay-Struktur erkennen und Rückschlüsse auf den inneren Adressraum ziehen. Diese Erscheinung ist allerdings die direkte Folge jeglicher strukturierter Ansätze, und ein direkter Nutzen für den Angreifer leitet sich daraus nicht ab.

2. **Gray- und Blackhole-Angriffe:** Angreifer, die einen direkten Einfluss auf verwendete Transportnetz-Router oder -verbindungen haben, können IPsec-Pakete vollständig oder gezielt verwerfen. Ersteres stellt kein Problem für SOLID dar; die Sicherheitsbeziehung wird als ausgefallen erkannt und der Verkehr wird soweit möglich umgeleitet. Das gezielte Unterdrücken von Paketen kann schwerwiegendere Folgen haben: Zum einen können ausschließlich Kontrollpakete gezielt durchgeleitet werden, sodass die Verbindung korrekt zu funktionieren scheint und keine Gegenmaßnahmen ergriffen werden. Zum anderen kann durch das periodische, kurzzeitige Unterdrücken von Verkehr, eine Art künstlicher „Wackelkontakt“ entstehen, der Schwingungen im VPN entstehen lässt. Beide Angriffe lassen sich jedoch auf vergleichsweise einfache Art statistisch erkennen (beispielsweise durch Überwachung der IPsec-Sequenznummern), und über möglicherweise nicht betroffene Pfade umleiten. An Grenzen stößt das Verfahren allerdings, wenn durch das Verwerfen beispielsweise die TCP-Staukontrolle aktiviert und so der Durchsatz stark verringert wird; dies unterscheidet sich allerdings nicht von manuell konfigurierten VPN-Ansätzen.

3. **Paketverzögerung:** Das Verzögern von Paketen hat innerhalb des SOLID-Netzes nur begrenzte Wirkung: Die Nutzdaten werden durch IPsec mittels eines Sliding-Window-Mechanismus geschützt, sodass Pakete nur in Übereinstimmung mit dem Fenster verzögert werden können. Das Request-Response-Protokoll für Verwaltungsnachrichten verwirft weiterhin Antworten bei Überschreitung einer maximalen Round Trip Time (RTT) (standardmäßig 10*s*). Nicht völlig auszuschließen ist an dieser Stelle, dass durch Verzögerungen, die innerhalb dieser Zeitspanne bleiben, das Vivaldi-Koordinatensystem zeitweise gestört wird. Im schlimmsten Fall könnten dadurch bei einem groß angelegten Angriff Optimierungen nicht mehr sinnvoll vorgenommen werden. Die Auswirkungen auf Anwendungen der Applikationsschicht, beispielsweise NTP, sind hierbei allerdings als signifikanter einzuschätzen.

4. **Wiedereinspielen von Paketen:** Erneute Übertragungen von IPsec-gesicherten Paketen werden durch den bereits erwähnten Fenstermechanismus effektiv erkannt und Duplikate verworfen. Einzige Ausnahme bildet der Bootstrapping-Mechanismus — hier kann entweder vollständig auf IPsec verzichtet werden, oder zumindest muss der in IPsec vorgesehene Schutz vor Wiedereinspielung wegen der Mehrpunktkommunikation deaktiviert werden. Daher wird durch eine übertragene Zufallszahl (Nonce) eine wiedereingespielte Antwortnachricht erkannt. Die Sicherung der Anfragenachricht ist komplexer: Hier erfolgt eine Sicherung zunächst über ein grobes Zeitfenster (beispielsweise eine Stunde), und eine feingranularere Überprüfung von Nonces in einem Bloom-Filter. Auch dieser Schutz ist nicht zwangsläufig vollständig, da neue Knoten oder solche, welche die Originalnachricht nicht erhalten haben, innerhalb des Zeitfensters auf die wiedereingespielte Nachricht mit einer Antwort reagieren. Allerdings wird dabei kein Zustand außerhalb des Bloom-Filters angelegt und maximal eine Antwort generiert. Insbesondere werden so Angriffe aufgrund wiedereingespielter Bootstrap-Nachrichten an Broadcast-Adressen verhindert (ähnlich Smurf-Attacken [Gar00]).

Mit wiedereingespielten Paketen kann somit pro VPN-Knoten, der das Originalpaket nicht erhalten hat, maximal ein Antwortpaket generiert werden; in allen anderen Fällen wird eine wiedereingespielte Nachricht immer erkannt.

5. **Senden gefälschter Pakete:** Der Schutz vor modifizierten oder frei konstruierten Paketen erfolgt analog zu den wiedereingespielten durch IPsec: Alle Nutzdaten und die Inhalte der meisten Verwaltungsnachrichten werden durch den MAC der ESP-Pakete geschützt. Diese ungültigen Pakete werden vergleichsweise schnell erkannt und verworfen. Werden sehr viele ungültige ESP-Pakete versendet, ist ein ineffizienter DoS-Angriff die Folge, welcher durch die schnelle Überprüfung des MAC in der Regel lediglich die Bandbreite erschöpfen kann, und gegen den somit von Seite des VPN-Betreibers nur das Verbergen der Endpunktadressen schützt.

   Auch bei der Erkennung von Modifikationen spielen die Bootstrapping-Pakete eine Sonderrolle, da sie über einen MAC auf Basis eines Gruppenschlüssels verfügen, der die Geräte eines VPN untereinander authentisiert. Allerdings erstreckt sich dieser nicht über den Quell-Port in NAT-T Szenarien, sodass ein Angreifer durch das Fluten mit extrem vielen Paketen, wegen der im Abschnitt 6.4 beschriebenen potentiellen Race-Condition, den Aufbau von Sicherheitsbeziehungen unter Umständen verhindern kann. Aufgrund der vielen Parameter (IP-Adressen, verwendeter SPI, exakter Zeitpunkt) kann solch ein Angriff allerdings nur erfolgreich verlaufen, wenn die Kommunikation bereits abgehört wird, oder auffällig viele Pakete über einen längeren Zeitraum gesendet werden.

   Wird IKEv2 standardkonform verwendet, das heißt ohne Zwangsverschlüsselung durch eine statische Sicherheitsbeziehung, ergibt sich ein weiteres potentielles Problem: Durch einen Man-in-the-Middle-Angriff beim Aufbau von IKE-Beziehungen können die Zertifikate der VPN-Knoten angefordert, und so deren Identität und inneren IP-Adressbereiche ermittelt werden. Abhilfe schafft hierbei ein Aufteilen der Zertifikate in IP-Adressattestationen und Zertifikate über die Identität, oder eine Zwangsverschlüsselung mit einer statischen Sicherheitsbeziehung. Letzteres kann auch das Fluten mit IKE-INIT-Nachrichten und den damit assoziierten potentiellen DoS-Angriff unterbinden. Sollen lediglich die inneren IP-Adressen einiger Knoten geheim gehalten werden, können diese möglicherweise auch ausschließlich indirekt mit öffentlichen Netzen verbunden und durch den äußeren Knoten geschützt werden.

Kompromittierter Klient

Wird ein System innerhalb eines roten Netzes kompromittiert, kann der Angreifer typischerweise innerhalb dieses Netzes alle Pakete abhören und analysieren. Gray- und Blackhole-Angriffe, Paketverzögerung, das Wiedereinspielen von Paketen und das Senden gefälschter Pakete ist innerhalb des Netzes ebenfalls möglich. Gegen solche Attacken würde lediglich eine IPsec-Zwangsverschlüsselung innerhalb der an sich vertrauenswürdigen Netze helfen.

Die VPN-Kommunikation außerhalb des betroffenen roten Netzes ist von dem Kompromittieren im Prinzip nicht betroffen. Die einzige Möglichkeit der Einflussnahme, ist das Erzeugen vieler Pakete innerhalb des VPN. Erfolgreiche DoS-Angriffe durch Netzwerküberlastungen innerhalb des VPN können aber durch Bandbreitendrosselung verhindert werden. Werden die Pakete an viele unterschiedliche Knoten gesendet, kann ein Angreifer möglicherweise die Smartcard des direkt verbundenen Gateways überlasten und verhindern, dass für andere Klienten Sicherheitsbeziehungen aufgebaut werden können. Wird innerhalb des roten Netzes auch IPsec eingesetzt, kann auch hier eine Ratenkontrolle zum Einsatz kommen. Langfristig wird durch den Aufbau vieler Verbindungen auch auf das Overlay-Routing Einfluss genommen, da tendenziell mehr Suchanfragen über das Gateway vermittelt werden. Ein direkter Nutzen für den internen Angreifer lässt sich daraus allerdings nicht ableiten. Kann zusätzlich der Verkehr des korrespondierenden Gateways auf der äußeren Seite beobachtet werden, können aber durch den Aufbau zusätzlicher Beziehungen auch mehr VPN-Knoten aufgedeckt werden. Im ungünstigsten Fall sind dies alle Knoten kompatibler Verfügbarkeitszonen.

Kompromittierter VPN–Knoten

In der für die Sicherheit ungünstigsten Situation werden ein oder mehrere VPN-Knoten direkt kompromittiert, und es wird so — in Erweiterung zu den bisher aufgeführten Angriffen — beispielsweise ein direkter Einfluss auf das Routing ausgeübt. In Bezug auf die einzelnen Angriffe bedeutet dies:

1. **Abhören und Verkehrsflussanalyse:** Da der Datenverkehr zwischen den VPN-Knoten in SOLID immer Ende-zu-Ende gesichert ist, können lediglich von oder zum kompromittierten Knoten gerichtete Pakete im Klartext mitgelesen werden. Bei allen weitergeleiteten Datenpaketen können ausschließlich der Quell- und Ziel-Knoten identifiziert werden. Darüber hinaus ist in Anfragen zum Aufbau reaktiver Sicherheitsbeziehungen auch die IP-Adresse des gesuchten Klienten vermerkt. Da innerhalb kompatibler Verfügbarkeitszonen auch die äußeren IP-Adressen der Knoten übertragen werden, kann durch einen internen Angreifer möglicherweise auch ein externer DoS-Angriff besser geplant werden. Insgesamt können somit Verkehrsflussanalysen erwartungsgemäß mit besserer Qualität durchgeführt werden; die Vertraulichkeit weitergeleiteter Daten ist jedoch nicht betroffen.

2. **Gray- und Blackhole-Angriffe:** Kompromittierte VPN-Knoten können aufgrund der besseren Verkehrsflussanalysen auch exakter steuern, welche Pakete aus dem weitergeleiteten Datenverkehr verworfen werden sollen. In Bezug auf das Unterdrücken von Verwaltungsnachrichten lassen sich die folgenden Ergebnisse erzielen:

    - **Ringaktualisierungen:** Das Unterdrücken von Nachrichten zum Aufbau der Ringstruktur kann einen negativen Einfluss auf die Konvergenz haben. Wie im Abschnitt 4.6.3 erläutert, werden diese Nachrichten jedoch stochastisch über nicht optimale Pfade versendet, und interne wie externe Angreifer

können umgangen werden, sofern sie nicht alle Pfade des Transportnetzes kontrollieren.

- **Suchanfragen:** Durch das Unterdrücken von Nachrichten zur Suche nach anderen Systemen können möglicherweise zielgerichtet Verbindungen zwischen Systemen verhindert werden. Das Vorgehen ist für den Angreifer im Prinzip effizienter als das Verwerfen des eigentlichen Datenverkehrs, kann aber je nach Transportnetzstruktur und Position der Angreifer durch das stochastische Routen von Suchanfragen auch verhindert werden.
- **Optimierungsnachrichten:** Das Verwerfen von Abkürzungs- und Handover-Nachrichten wirkt sich primär wie ein Sink-Hole-Angriff aus: Der angreifende Knoten kann verhindern, dass über ihn weitergeleitete Verbindungen zu anderen Knoten vermittelt werden. Der Angreifer kann so möglicherweise mehr Verbindungen kontrollieren.

Durch interne Grayhole-Angriffe kann somit — zumindest bis zum Finden alternativer Pfade und je nach Einfluss der Angreifer — gezielt die Performanz des VPN verringert werden.

Würden die Pakete vollständig verworfen, unterscheidet sich der resultierende Blackhole-Angriff nicht von der externen Variante.

3. **Paketverzögerung:** Der zu erreichende Einfluss durch das Verzögern von Daten- und Verwaltungspaketen gestaltet sich nahezu analog zum externen Angreifer. Der einzige Unterschied besteht darin, dass auch hier der interne Angreifer möglicherweise besser planen kann, welche Pakete wie lang verzögert werden sollen.
4. **Wiedereinspielen von Paketen:** Durch die verwendete Ende-zu-Ende-Sicherung ist der Datenverkehr vollständig vor wiedereingespielten Paketen geschützt. In Bezug auf die Ringaktualisierungs- und Suchnachrichten wird das eingesetzte Request-Response-Protokoll durch Nonces geschützt. Ähnlich wie beim Schutz der Bootstrapping-Pakete wird auch hier ein Bloom-Filter eingesetzt, um wiedereingespielte Requests zu erkennen. Bei Optimierungsnachrichten wird von der empfangenen Partei jeweils geprüft, ob diese bezogen auf den aktuellen Zustand der VPN-Topologie sinnvoll erscheint, sodass auf wiedereingespielte Pakete nicht reagiert wird.
5. **Senden gefälschter Pakete:** Modifizierte oder frei konstruierte Pakete bedürfen in Szenarien mit internen Angreifern einer differenzierten Betrachtung:
   - **Nutzdaten:** Auch durch interne Angreifer verursachte Veränderungen können bei weitergeleiteten Datenpaketen durch den Ende-zu-Ende-Schutz erkannt werden, sodass lediglich Daten der direkt angrenzenden roten Netze betroffenen sind. Mögliche Angriffe auf Nutzdaten gehen somit nicht über das Niveau eines kompromittierten Klienten hinaus.
   - **Bootstrapping:** Da der symmetrische Gruppenschlüssel durch Kompromittierung eines einzigen VPN-Knotens bekannt wird, muss in diesem Fall der Schlüssel in allen nicht kompromittierten Geräten geändert werden.

Andernfalls kann der Angreifer VPN-Knoten mit gefälschten Bootstrap-Anfragen fluten, welche unter Umständen für Smurf-ähnliche Reflektionsangriffe genutzt werden können, oder bei einer Ratenlimitierung dazu führen, dass das Einfügen neuer Knoten unter Umständen sehr lange dauert. Darüber hinaus sind allerdings keine signifikanten Auswirkungen zu erwarten. Falls das Kompromittieren von Knoten wahrscheinlich scheint und für die Aufbewahrung der privaten VPN-Schlüssel Smartcards verwendet werden, könnten diese ebenfalls genutzt werden den Gruppenschlüssel zu sichern. Die Bandbreite der Reflektionsangriffe ist so auf die Rate limitiert, mit der die entwendeten Smartcards MAC-Werte generieren.

- **Ringaktualisierungen:** Bis auf wenige Ausnahmen können alle Angaben in Ringaktualisierungsnachrichten durch Zertifikate oder den Kontext überprüft werden. Einzige signifikante Ausnahme stellt die Position im Vivaldi-Netzwerkkoordinatensystem dar, welche erst zur Laufzeit berechnet wird. Zwar existieren verschiedene Systeme zum Filtern unrealistischer Koordinaten, diese können aber ohne zusätzliche Annahmen noch nicht einmal Konvergenz garantieren, wie in dem durch diese Arbeit initiierten Artikel [GRS+10a] bewiesen werden konnte. Im ungünstigsten Fall konvergiert das Koordinatensystem nicht oder ein Angreifer kann es nutzen, die Koordinaten so zu modifizieren, dass er einen Sinkhole-Angriff initiiert und mehr Verbindungen über ihn weitergeleitet werden. Der skizzierte Angriff ist allerdings extrem schwierig zu koordinieren, da verschiedene Sicherungsmaßnahmen implementiert sind, und die Koordinaten beispielsweise lediglich mit Nachbarn ausgetauscht werden, zu denen eine proaktive Sicherheitsbeziehung aufgebaut wurde. Ferner wird primär in Bezug auf Overlay-Hops optimiert, sodass Angriffe gegen Vivaldi lediglich bei der Umgehung gestörter Verbindungen Auswirkungen haben.

- **Suchanfragen:** Die ausgetauschten Nachrichten zum reaktiven Aufbau von Sicherheitsbeziehungen besitzen im Prinzip den gleichen Aufbau wie die Ringaktualisierungen, da sich lediglich das Routing unterscheidet. Die Sicherheitseigenschaften gegenüber Modifikationen sind identisch.

- **Optimierungsnachrichten:** Umleitungs- und Vivaldi-Optimierungsnachrichten können innerhalb von SOLID nicht weitergeleitet werden, sodass der Sender eines Paketes immer klar anhand seiner IP-Adresse identifiziert werden kann. Die Mechanismen sind ferner so ausgelegt, dass ein Knoten nur Pfade über andere Knoten schalten kann, im ungünstigsten Fall kann ein Angreifer Handover zwischen von ihm kompromittierten Knoten verursachen. Sinkhole-Angriffe werden jedoch nicht ermöglicht. Die Plausibilitätsprüfungen sichern außerdem zu, dass Optimierungen immer zu einer Verbesserung der jeweiligen Metrik führen. Dadurch können auch anderen, kompromittierten Knoten nicht beliebige Verbindungen zugespielt werden. Wie bereits angedeutet, stützen sich diese Prüfungen beim Handover-Vorgang allerdings auf Vivaldi-Koordinaten, sodass über gefälschte Koordinaten möglicherweise je nach Szenario und im beschränkten Maße doch Sinkhole-Angriffe realisiert werden können.

- **IKE-Verkehr:** Das eingesetzte IKEv2 Protokoll ist im Prinzip gegen byzantinische Angriffe geschützt. Durch den Einsatz der Mediation-Erweiterung kommt jedoch mit dem Mediator eine weitere Instanz hinzu, der zumindest teilweise vertraut werden muss. Wird ein kompromittierter Knoten von den durch NAT angebundenen VPN-Knoten selektiert, kann dieser ihnen falsche äußere IP-Adressen mitteilen. Im ungünstigsten Fall sind dies die IP-Adressen eines Relays, welches unter Kontrolle des Angreifers ist, und diesem weiterhin eine Kontrolle der Beziehung erlaubt.

In diese Angriffskategorie fallen auch die in den Grundlagen angesprochenen Sybil- und Eclipse-Attacken (Teilabschnitt 2.4.5), die speziell im Kontext von Peer-to-Peer-Systemen ausführlich diskutiert wurden. Beide Angriffe sind jedoch aufgrund der kryptographischen Instanzenauthentisierung in SOLID nicht durchzuführen.

Bei internen Angriffen ist somit zu meist mit Sinkhole-Attacken zu rechnen, die wiederum zur Vorbereitung besserer Verkehrsflussanalysen und Grayhole-Angriffe genutzt werden können. Die Netzüberlastung durch zu viele Pakete kann durch Ratenkontrolle lokal gehalten werden, und unterscheidet sich somit nicht wesentlich von externen Angriffen.

Fazit: Das erreichte Sicherheits– und Verfügbarkeitsniveau

In Bezug auf die spezifizierten Sicherheitsanforderungen bedeutet dies:

1. **Vertraulichkeit:** Sowohl ein Großteil der Verwaltungs- als auch sämtliche Nutzdaten werden durch eine obligatorische IPsec-Verschlüsselung geschützt.

   a) Der Schutz der Nutzdaten erfolgt dabei je nach Bedarf durch verschachtelte Tunnel, sodass eine **Ende-zu-Ende-Sicherung** garantiert werden kann.

   b) Der in IPsec enthaltene Diffie-Hellman-Schlüsselaustausch sichert für alle Beziehungen **Perfect Forward Secrecy (PFS)** zu.

   c) Durch die Zwangsverschlüsselung aller Datenpakete kann eine gewisse **Resistenz gegen verdeckte Kanäle** erreicht werden. Klienten können an äußere Systeme keine Pakete direkt senden. Vielmehr kann eine Kommunikation über einen verdeckten Kanal nur stattfinden, wenn ein Angreifer den Verkehr zwischen zwei VPN-Knoten beobachtet, und der Klient seine Informationen in Sendefrequenz und Paketgröße kodiert. Paddings, Cover Traffic und ähnliche andere Maßnahmen zum weiteren Schutz können bei Bedarf aus herkömmlichen VPN übernommen werden und wurden im Rahmen dieser Arbeit nicht weiter untersucht.

   d) In Bezug auf **Infrastructure Hiding** können die Größen der roten Netze vor Angreifern verborgen werden. Durch die Weiterleitung indirekten Verkehrs

kann außerdem das genaue Verkehrsaufkommen pro Knoten schwerer abgeschätzt werden. Sofern keine weiteren Schutzmaßnahmen implementiert sind, wie die Verschlüsselung von IKE-Verkehr mithilfe eines statischen Gruppenschlüssels, können allerdings durch Man-in-the-Middle-Angriffe die inneren IP-Adressen der Knoten aufgedeckt werden. Ein direkter Nutzen ergibt sich dadurch noch nicht, allerdings können so möglicherweise andere Angriffe besser koordiniert werden. Beispielsweise können Knoten identifiziert werden, welche anschließend mittels einer gezielten DoS-Attacke angegriffen werden.

2. **Datenintegrität und –authentizität:** Alle von IPsec verschlüsselten Daten werden auch automatisch durch einen MAC gesichert, sodass für Nutzdaten sowohl Integrität als auch Authentizität Ende-zu-Ende überprüft werden können. Bootstrap-Nachrichten werden darüber hinaus ebenfalls durch einen MAC geschützt, welcher aber lediglich auf einem Gruppenschlüssel basiert und nur zum Verhindern von DoS-Angriffen dient.

3. **Instanzenauthentisierung:** Alle VPN-Knoten innerhalb eines SOLID-VPN müssen sich durch ein Zertifikat/Schlüsselpaar gegenüber ihren Kommunikationspartnern authentisieren. Die im Zertifikat attestierten IP-Adressbereiche werden dabei zur Einrichtung von Routing-Einträgen und Firewall-Regeln verwendet, sodass rote IP-Adressen zu jedem Zeitpunkt einem VPN-Knoten zugeordnet werden können.

4. **Kontrollierter Zugang:** Bei der Authentisierung wird durch den IPsec-Daemon geprüft, ob das Zertifikat des Gegenübers von einer kompatiblen Zertifizierungsinstanz ausgestellt wurde. Zertifikate anderer Zertifizierungsinstanzen werden zurückgewiesen, sodass eine **statische Zugriffskontrolle** realisiert wird. Ferner findet bei periodischen Reauthentisierungen auch eine **dynamische Zugriffskontrolle** statt. VPN-Knoten deren Smartcard entfernt wurde, können spätestens zu diesem Zeitpunkt vom VPN ausgeschlossen werden. Die Realisierung von *Forward* und *Backward Access Control* wird bei der Verteilung von Multicast-Verkehr implizit erreicht, da die entsprechenden Pakete lediglich über Unicast-Sicherheitsbeziehungen mit Reauthentisierungen weitergeleitet werden.

5. **Minimale sicherheitsrelevante Konfiguration:** Wie bereits bei der Diskussion zur Selbstkonfiguration erwähnt, benötigen SOLID-Knoten sehr wenige individuelle Einstellungen. Sicherheitsrelevant sind im allgemeinen Fall lediglich der öffentliche Schlüssel und das korrespondierende Zertifikat mit dem angegeben IP-Adressbereich, sowie der verwendete Bootstrap-Schlüssel. Werden die eingesetzten Geräte individuell konfiguriert, kann die Zuordnung von vertrauenswürdigen und nicht vertrauenswürdigen Netzwerkschnittstellen ebenfalls sicherheitsrelevant sein.

Ein primärer Fokus des entwickelten Systems ist die Realisierung einer hohen Sabotageresistenz. Dies ist wie folgt erreicht worden:

1. **Zerstörung von Ressourcen:** Gegenüber Angreifern weisen SOLID-Knoten eine sehr geringe Angriffsfläche auf, da alle Pakete integritätsgesichtert und authenti-

siert sind. Somit wird eine Zerstörung durch Software-Fehler und Abweichen von den eingesetzten Protokollen stark eingeschränkt; zumal die dafür eingesetzten IPsec-Protokolle und –Anwendungen in Bezug auf diese Eigenschaften gut untersucht sind. Sollten einzelne Systeme, beispielsweise durch physische Angriffe, dennoch nicht länger zur Verfügung stehen, rekonfiguriert sich das verteilte System automatisch, um den Ausfall im restlichen Netz zu kompensieren. Weiterhin kann der Ausfall einzelner Verbindungen durch Umleitung des Verkehrs toleriert werden.

2. **Erschöpfung von Ressourcen:** Der Schutz gegen Überlastsituationen umfasst folgende Ressourcen:

   - **Rechenzeit:** Aufwendige Rechenoperationen werden von SOLID-Knoten gegenüber unauthentisierten Kommunikationspartnern vermieden. Die Authentizität von Bootstrap-Anfragen und ESP-Paketen erfolgt über eine relativ schnelle MAC-Prüfung. Für die IKE-Aushandlung sind zwar aufwendige asymmetrische Kryptooperationen notwendig, allerdings verfügt IKEv2 über einen Cookie-Mechanismus, welcher vor dieser Operation optional die Erreichbarkeit der Quell-IP-Adresse überprüft und gegebenenfalls die Antworten zum Kommunikationspartner drosselt.

     Interne Angreifer können potentiell DoS-Angriffe auf die Rechenressourcen durchführen, indem sie andere Knoten Signaturen überprüfen lassen. Gegebenenfalls könnten hier einfache Ratenkontrollmechanismen implementiert werden, da die Quell-IP-Adresse zu jedem Zeitpunkt authentisiert ist.

   - **Speicher:** Reservierungen von Hauptspeicher werden ebenfalls nur für authentisierte Geräte vorgenommen. Pro VPN-Knoten ist dies konstant viel, sodass auch interne Angreifer keine protokollbedingte Überlastsituation erzeugen können. Massenspeicher wird lediglich für die Log-Ausgabe verwendet. In einem Produktivsystem müssen Filter an dieser Stelle eine zu hohe Auslastung vermeiden.

   - **Netzwerkbandbreite:** Reaktive Maßnahmen gegen externe Angriffe, die auf eine Überlastung von Netzwerkressourcen zielen, sind ohne Unterstützung der ISP nicht möglich. Daher können in SOLID proaktiv Verfügbarkeitszonen eingerichtet werden, um Auswirkungen von DoS-Angriffen nach Möglichkeit lokal zu begrenzen. Intern kann ein Schutz einfach mittels Ratenkontrolle erfolgen.

3. **Graceful Degradation:** Um die Auswirkungen von Verfügbarkeitsproblemen lokal begrenzt zu halten, wurden alle Subsysteme von SOLID vollständig verteilt entworfen, sodass der Ausfall beliebiger Knoten und Verbindungen toleriert werden kann. Voraussetzung ist lediglich, dass das VPN zusammenhängend ist; andernfalls operieren die Teilnetze unabhängig weiter. Werden einzelne Systeme kompromittiert, können bis auf die Verfügbarkeit auch alle Sicherheitsanforderungen weiterhin vollständig umgesetzt werden. Wie im Abschnitt zu internen Angreifern erläutert, kann aber selbst durch diese die Verfügbarkeit des VPN

nicht nach Belieben negativ beeinflusst werden; der maximale Grad hängt stark vom Transportnetz und der VPN-Topologie zum Zeitpunkt des Angriffs ab.

4. **DoS-Recovery:** Systeme, die von einem DoS-Angriff betroffen sind, können aufgrund der guten Agilitätseigenschaften schnell mit anderen äußeren IP-Adressen in das VPN reintegriert werden, sofern der ISP dies zulässt oder eine redundante Internet-Anbindung vorliegt. Gerade für relativ schmalbandige Verbindungen, die von DoS-Angriffen besonders betroffen sind, ist beispielsweise eine Wiedereinwahl eine gute Option mit dem Problem umzugehen.

Insgesamt realisiert SOLID durch die Ende-zu-Ende-Sicherung, die vollständige Verteilung und den Einsatz von Overlay-Routing-Techniken ein sehr gutes Sicherheits- und Verfügbarkeitsniveau. Die Einführung von Verfügbarkeitszonen soll es erlauben, trotz potentieller Bedrohung durch DoS-Angriffe ein möglichst flexibles und gleichzeitig robustes VPN aufzubauen.

7.2 Quantitativ evaluierte Topologien und Szenarien

Einige der erreichten Eigenschaften werden in den folgenden Unterkapiteln näher untersucht. Aufgrund der Komplexität des vollständigen Systems wurde auf eine analytische Herleitung von Kennzahlen verzichtet, und stattdessen Experimente auf Basis des Prototypen und Netzwerksimulators vorgenommen. Dazu wird zunächst auf die evaluierten Szenarien eingegangen, bevor die eigentlichen Experimente mit ihren Ergebnissen vorgestellt werden. Abschließend werden die einzelnen charakterisierenden Versuche auf die spezifizierten Anforderungen zurückgeführt.

Im Einzelnen sind in den einzelnen Kapiteln der Ausarbeitung die folgenden Punkte identifiziert worden, die durch Experimente näher zu untersuchen sind:

- Für die Einschätzung des *Prototypen* ist es wichtig, die Verzögerung des reaktiven Verbindungsaufbaus und die benötigte Bandbreite für Verwaltungsinformationen zu erfassen.
- Bezüglich des *Discovery*-Mechanismus muss die Verteilung des Knotengrades und die Effizienz des stichprobenbasierten Verfahrens im Vergleich zu Skip-Graphen betrachtet werden.
- Für das *Routing* ist das Verhalten der Optimierungsalgorithmen in nicht-optimalen Transportnetzen sowie bei partiellen Transportnetzstörungen zu evaluieren.
- Im Abschnitt zur *Topologiekontrolle* wurde ferner das Verhalten bei der Wiederherstellung nach Partitionierungen als untersuchenswert akzentuiert.
- In Hinblick auf die *Sabotageresistenz* ist die Beobachtbarkeit von SOLID-VPN und die durch Verfügbarkeitszonen eingeführte zusätzliche Ende-zu-Ende-Verzögerung von Relevanz.

Insgesamt wurden die Themen drei verschiedene Szenarien näher untersucht: kleinere Experimente konnten am Prototypen in einem Labornetz mit 18 funktionsfähigen SOLID-Knoten vollzogen werden. Die Ergebnisse aller anderen Experimente sind aufgrund der VPN-Größe im Netzwerksimulator entstanden. Dabei kamen je nach Versuchsanordnung direkte und verschachtelte Topologien zum Einsatz.

7.2.1 Struktur des Labornetzes

Wie in Abbildung 7.1 illustriert, besitzt das realisierte Laborszenario 18 SOLID-Knoten, welche über Ethernet-Verbindungen vernetzt sind. Bei der Konstruktion der Topologie wurde darauf geachtet, dass möglichst viele Spezialfälle abgedeckt werden. So existieren zwei schwarze Netze, mehrere Gateways in einem roten Netz, tiefe Verschachtelungen und innerhalb des „Intranet“ eine relativ starke Verzögerung von $100ms \pm 5\%$. Die Aufteilung der roten IP-Adressen begünstigt zusätzlich eine logische Partitionierung, falls Knoten 150 temporär nicht verfügbar ist.

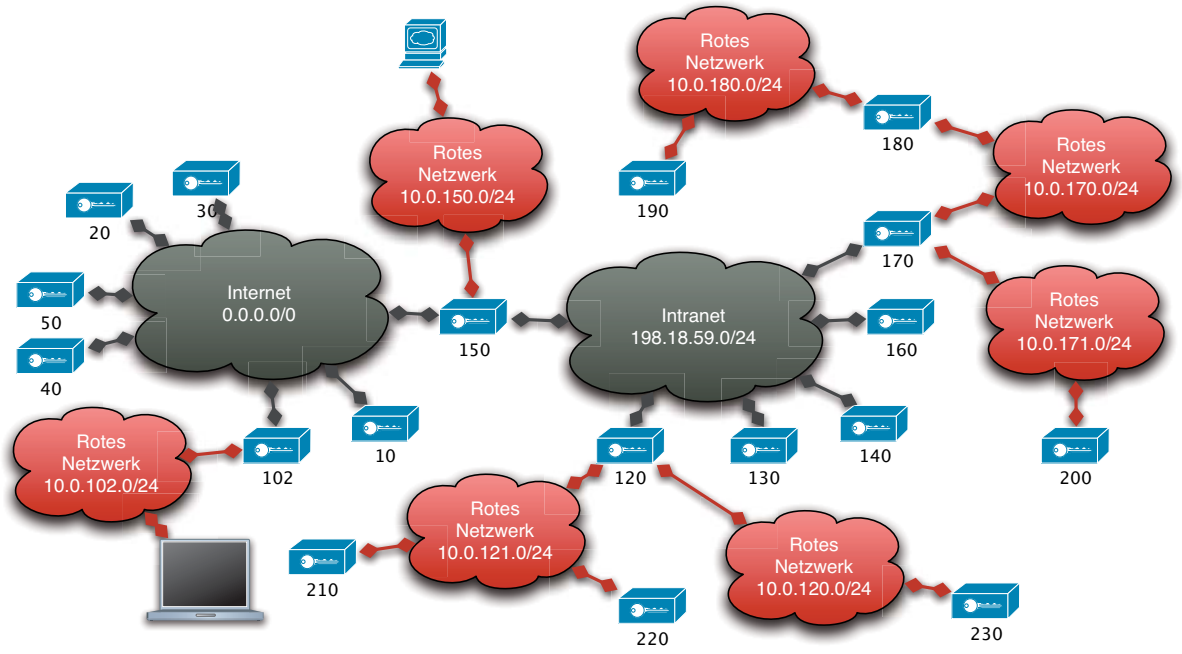


Abbildung 7.1: Evaluiertes Laborszenario

7.2.2 VPN mit direkter Konnektivität

Für einige einfache Simulationen, beziehungsweise zur Untersuchung von Effekten in großen Transportnetzen, kann angenommen werden, dass alle VPN-Knoten eine direkte Verbindung zum Internet haben. Da durch den Konfigurationsmechanismus aufgrund der niedrigen benötigten Bandbreite keine Stausituationen ausgelöst werden, kann die zu untersuchende Topologie stark vereinfacht werden. In diesem *direkten Szenario* sind

alle Knoten mit einem zentralen Router verbunden, der die Weiterleitung übernimmt. Um trotzdem realistische Verzögerungszeiten zu erhalten, werden die Pakete dabei entsprechend einer auf Internet-Messungen basierenden Meridian-Verzögerungsmatrix [WSS05] ausgeliefert.

7.2.3 Verschachtelte VPN–Topologien

Für komplexere VPN sind bisher keine Topologien öffentlich verfügbar. Zum einen liegt dies wahrscheinlich daran, dass heutige Konfigurationsverfahren außer vollvermaschten und Hub-and-Spoke-Netzen keine weiteren Szenarien unterstützen. Zum anderen sind die Strukturen tatsächlich realisierter VPN wegen der Bedrohung durch mögliche Angriffe in der Regel vertraulich.

Aus diesem Grund werden für die Simulationen generische Szenarien auf Basis der Heuristically Optimal Topology (HOT) [LAWD04] verwendet. Dabei handelt es sich um künstliche Topologien, die realen Router-Graphen nachempfunden sind und mithilfe der Werkzeuge des Orbis-Projektes [MHK+07] ohne Verlust ihrer Eigenschaften nahezu beliebig skaliert werden können. Insbesondere lassen sich für die einzelnen Simulationsläufe unterschiedliche Topologien verwenden, sodass auch die Performanz in unterschiedlichen Szenarien evaluiert wird. Da VPN-Gateways im Gegensatz zu herkömmlichen Routern eher am Rand des Netzes eingesetzt werden, wurden lediglich die baumartigen Randstrukturen des HOT-Graphen durch SOLID-Gateways ersetzt und das Kernnetz weiterhin mit Routern realisiert. Es ergeben sich Szenarien, wie das in Abbildung 7.2 dargestellte Beispiel.

Die verwendeten Topologien lassen sich ferner durch die in Tabelle 7.3 gezeigten Eigenschaften charakterisieren. Mit steigender Anzahl an VPN-Knoten wächst beispielsweise die Anzahl der Router zunächst schnell an, um dann in einen nahezu proportionalen Verlauf überzugehen. Während der durchschnittliche Knotengrad dabei in allen Szenarien nahezu identisch bleibt, zeigt sich allerdings auch, dass der maximale Knotengrad in den Netzgrößen sprunghaft anwachsen kann. Insbesondere wegen der hohen Verzögerung bei der Aushandlung von IKE-Beziehungen können die hohen Knotengrade einen signifikanten Einfluss auf die Gesamtperformanz haben. Da diese eine Größenordnung über den Verzögerungszeiten globaler Rechnernetze liegen, stellen sie bei der Konfiguration der Netze den dominierenden Faktor dar. Daher wurde bei der Modellierung der Szenarien auf eine exakte Verteilung der Verzögerungszeiten verzichtet, und alle Verbindungen zwischen Routern sowie die Anbindungen der SOLID-Gateways verfügen über eine Latenz von $10ms$.

7.2.4 Gewählte Standardparameter

Soweit nicht anders angegeben, versenden SOLID-Knoten alle $60s$ Heartbeat-Nachrichten zur Erkennung ausgefallener Kommunikationspartner. Im gleichen Intervall wird

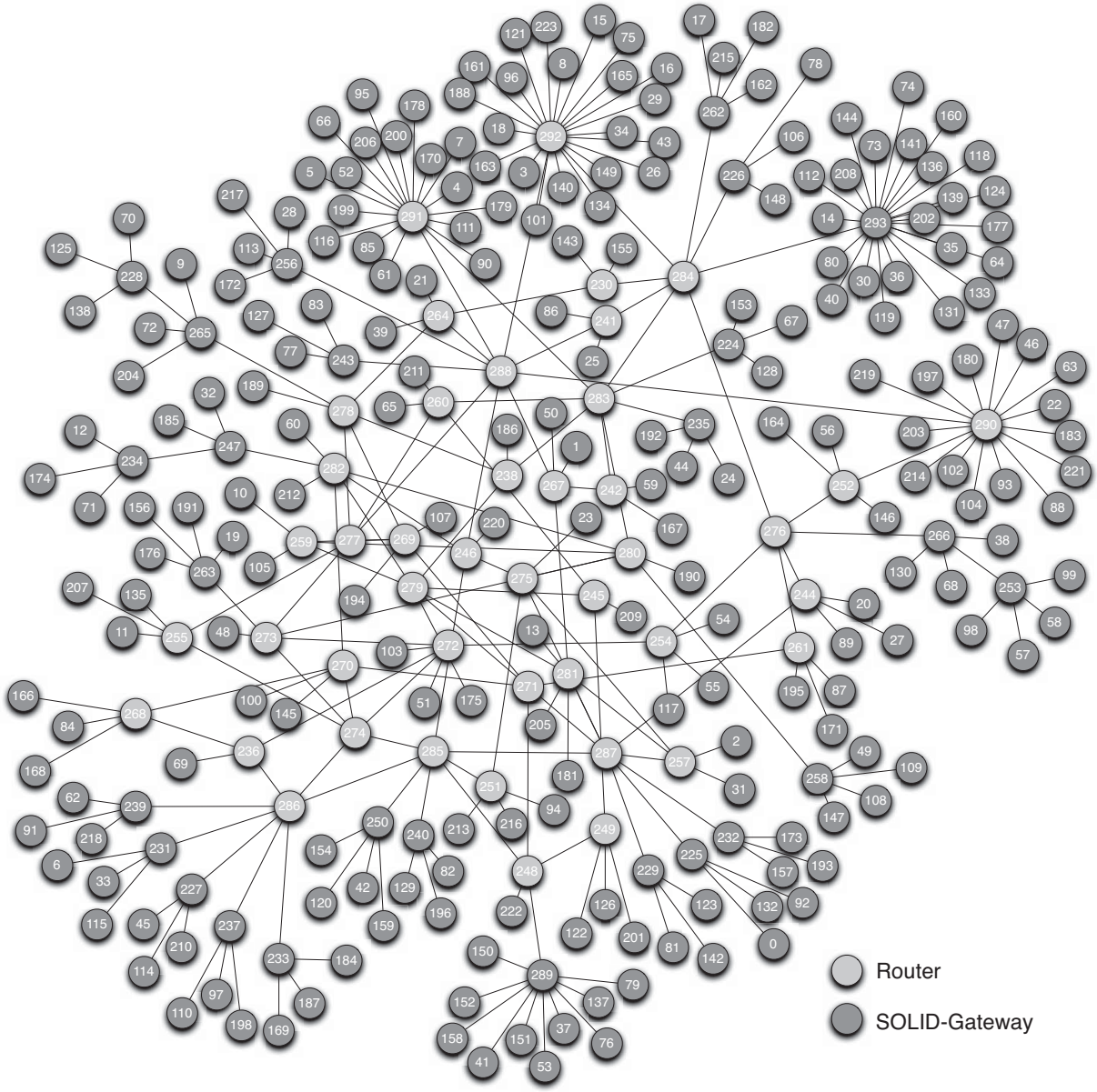


Abbildung 7.2: Beispiel eines aus einem HOT-Graphen generierten Simulationsszenarios mit 250 SOLID-Knoten

Anzahl VPN-Knoten	50	100	150	200	250	300	350	400	450	500
Ø Anzahl Router	7.88	21.19	32.34	38.34	42.44	47.44	49.94	56.12	60.78	65.66
Ø Durchmesser	5.84	8.12	8.88	9.03	9.78	9.75	9.97	10.53	10.44	10.59
Ø Kernnetzdurchmesser	2.91	4.50	5.09	5.28	5.47	5.59	5.94	6.06	6.19	6.47
Ø Knotengrad	2.12	2.25	2.28	2.26	2.23	2.23	2.20	2.20	2.19	2.18
Ø VPN-Knotengrad	1.65	1.55	1.56	1.44	1.52	1.67	1.69	1.65	1.66	1.67
Max. VPN-Knotengrad	24	24	24	24	46	55	59	68	68	75

Tabelle 7.3: Zentrale Eigenschaften der untersuchten HOT-Topologien

auch nach neuen Ringnachbarn gesucht, wobei nach dem Aufbau der ersten Sicherheitsbeziehung sofort eine solche Suche gestartet wird. Mit einer Wahrscheinlichkeit von 5% wird die Aktualisierung der Ringnachbarn, wie in Teilabschnitt 4.6.3 motiviert, über fünf zufällige Beziehungen gesendet, bevor eine zielgerichtete Weiterleitung erfolgt.

Der Aufbau von Sicherheitsbeziehungen zu anderen Knoten wird nach maximal 120s erfolglos abgebrochen. Dieser relativ lange Wert ist auf die Dauer des Aufbaus von IKE-Beziehungen zurückzuführen, welcher in der Standardkonfiguration um 1s verzögert wird. Da sich die Einzelverzögerungen akkumulieren, wird für einen Aufbau von Sicherheitsbeziehungen zu zentral gelegenen Knoten somit ein großer Wert für Zeitüberschreitung benötigt.

Nicht länger benötigte Beziehungen werden in den Experimenten nach 600s abgebaut, und die VPN-Größe beträgt im Standardfall 250 Knoten.

7.3 Evaluierung des Prototypen

Wie bereits in der Beschreibung des Laborszenarios angegeben, eignet sich das Labornetz aufgrund der geringen Größe nur für die Durchführung einiger kleiner Experimente. Im Folgenden sollen vor allem die Einflüsse von SOLID auf den Nutzer, sowie verschiedene Transportnetz- und VPN-Parameter untersucht werden. Alle erläuterten Experimente wurden 100 Mal durchgeführt, um repräsentative Minima, Mittelwerte und Maxima zu ermitteln. Ferner werden in vielen Graphiken die Mittelwerte mit ihren Konfidenzintervallen als rote Bereiche angegeben, wobei sich diese auf ein Signifikanzniveau von 99% beziehen.

7.3.1 Reaktiver Aufbau von Sicherheitsbeziehungen

Ein wesentlicher Ansatzpunkt, die Skalierbarkeit von VPN zu gewährleisten, ist der reaktive Aufbau von Sicherheitsbeziehungen. Diese bedingen allerdings eine kurze Verzögerung, bevor Pakete an vormals unbekannte VPN-Teilnehmer ausgeliefert werden können. Da von der Dauer des Aufbaus die Akzeptanz durch den Nutzer abhängt, sollen neben der Dauer der Signaturgenerierung weitere Faktoren identifiziert werden, die den Aufbau signifikant beeinflussen. Da keine Smartcards eingesetzt wurden, ist aufgrund der vergleichsweise kurzen Latenz der IKE-Signaturerzeugung und der vielen übermittelten Pakete zu erwarten, dass die RTT einen signifikanten Einfluss hat.

Von zwei zentralen Knoten in Inter- und Intranet (20 und 130) wurden dazu ICMP-Echo-Requests versendet und die Antwortzeit beim Ablauf der einzelnen Discovery-Phasen gemessen. Im Einzelnen sind das Eintreffen der Antwort auf die Suchnachricht,

des IKE_INIT, des IKE_AUTH und des ICMP Reply aufgezeichnet worden. Die korrespondierenden Partner befinden sich jeweils in anderen Netzsegmenten, um direkte Pfade auszuschließen.

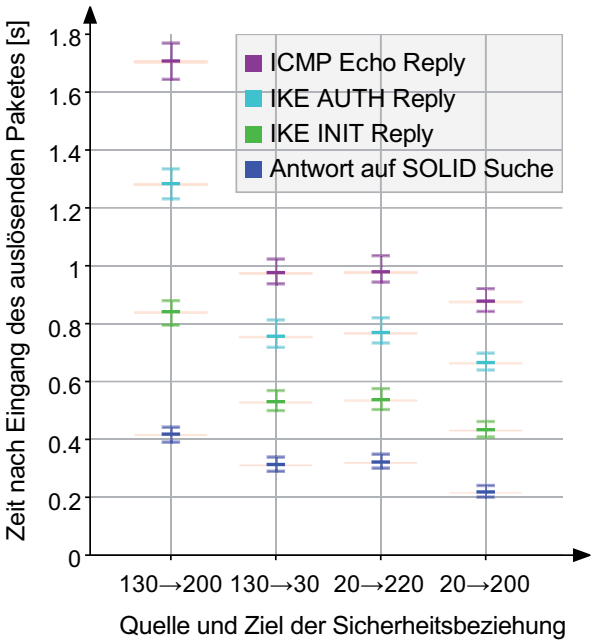


Abbildung 7.4: Durchschnittliche Dauer des reaktiven Aufbaus von Sicherheitsbeziehungen (Laborszenario)

Abbildung 7.4 zeigt die ermittelten Messgrößen: Wie erwartet beträgt der Abstand der Mittelwerte in etwa Vielfachen von 200*ms*, also der RTT durch das Labor-Intranet. Der Aufbau der Beziehung zwischen Knoten 130 und Knoten 200 muss aufgrund physischer Bedingungen Knoten 170 traversieren. Dieser wird über einen der Knoten 140, 150 oder 160 in einem Overlay-Hop erreicht, und leitet die Suchnachricht sowie die IPsec-Pakete weiter. Durch den Zwischenknoten beträgt die durchschnittliche RTT etwas mehr als 400*ms*. Durch die Übertragung von vier Request-Response-Paaren ergibt sich bereits eine Verzögerung von circa 1,6*s*. Die Verarbeitungszeiten in den Knoten beträgt bei den eingesetzten ECDSA-256-Signaturen lediglich 100*ms*. Eine Verkürzung der Routen durch die Optimierungsmechanismen geschieht erst später.

Beim Initiieren einer Beziehung von Knoten 130 zu 30 ist zu erkennen, dass das Intranet drei Mal traversiert wird. Der Grund dafür liegt im Overlay-Routing: Knoten 130 sendet das Paket entlang des Ringes zu Knoten 120, welcher es über Knoten 150 an Knoten 50 weiterleitet. Knoten 150 hat allerdings eine direkte Beziehung zu 130, sodass auf dem Rückweg der kürzere Weg verwendet wird.

Der gleiche Effekt tritt auch beim Verbindungsaufbau zwischen 20 und 220 auf: der initial gefundene Pfad ist $20 \rightarrow 150 \rightarrow 170 \rightarrow 200 \rightarrow 170 \rightarrow 120 \rightarrow 220$. Das Antwortpaket gelangt allerdings bereits auf kürzestem Weg zum Ziel, da Knoten 150 und 120 über eine direkte Verbindung verfügen.

Insgesamt dominiert in Szenarien ohne Smartcards die RTT klar das reaktive Verhalten. Vor dem Aufbau globaler VPN bietet es sich daher an Optimierungen vorzunehmen. Denkbar wäre beispielsweise, wie in Pastry bei Overlay-Suchen den Quotienten aus Overlay- und Transportnetzdistanz zu minimieren, und so den initial gefundenen Pfad zu verkürzen. Ferner können über lange bestehende Beziehungen mit einem herkömmlichen Routing-Protokoll Informationen ausgetauscht werden, sodass eine aufwendige Suche und Optimierung entfällt. Ein Spezialfall von diesem Ansatz ist der Lookahead-Mechanismus in Symphony (siehe Teilabschnitt 2.4.1). Weiterhin könnten die IKE_INIT-Pakete per Piggybacking bereits in der Suchnachricht übertragen

werden. Gleiches gilt für die Nutzdatenpakete und die `IKE_AUTH`-Nachrichten. Neben einer effizienteren Suche, würde sich die Verzögerung so auf zwei RTT-Intervalle reduzieren.

7.3.2 Erzeugter Verwaltungsaufwand

Ein weiteres mögliches Akzeptanzproblem für SOLID ist der höhere Verwaltungsaufwand, den es im Vergleich zu statischen Verfahren verursacht. Dabei spielt vor allem der Mehraufwand in Bezug auf die zusätzlich übertragenen Pakete eine signifikante Rolle, da Rechen- und Speicherressourcen in der Regel kein Problem darstellen. SOLID-Knoten übertragen neben Bootstrapping-Nachrichten, Heartbeats und Ringaktualisierungen kaum Verkehr — diesen allerdings kontinuierlich. Die verwendete Bandbreite muss somit klar von der Anzahl der Sicherheitsbeziehungen abhängen.

Daher wurde in verschiedenen Knoten des Labornetzes die durch proaktive SOLID-Beziehungen induzierte mittlere Bandbreite gemessen. Da die Periode der Heartbeats und Ringaktualisierungen 60*s* beträgt, wurde als Basis jeder Messung ein Mittelwert über 120*s* verwendet, sodass in jedem Zeitfenster zwei Perioden abgedeckt sind.

Erwartungsgemäß zeigt Abbildung 7.5 sehr unterschiedliche durchschnittliche Bandbreiten, welche primär von der Struktur des Transportnetzes und der Position des Knotens darin abhängen. Die für Verwaltungsaufgaben verwendete Bandbreite unterscheidet sich in den Knoten 30, 200 und 230 statistisch nicht und ist etwas geringer als 100 $Byte/s$. Knoten 130 wendet geringfügig höhere Ressourcen auf, um Verwaltungsaufgaben wahrzunehmen. Da er sich direkt im Intranet befindet, erhält er geringfügig mehr Bootstrapping-Nachrichten und umgeleitete Ringaktualisierungen. Die Knoten 120 und 150 verwenden etwa 500 beziehungsweise 450 $Byte/s$, was auf ihre exponierte Lage zurückzuführen ist. Knoten 150 leitet alle Daten zwischen den Netzteilen weiter, und Knoten 120 ist das einzige Gateway für Knoten 210, 220 und 230.

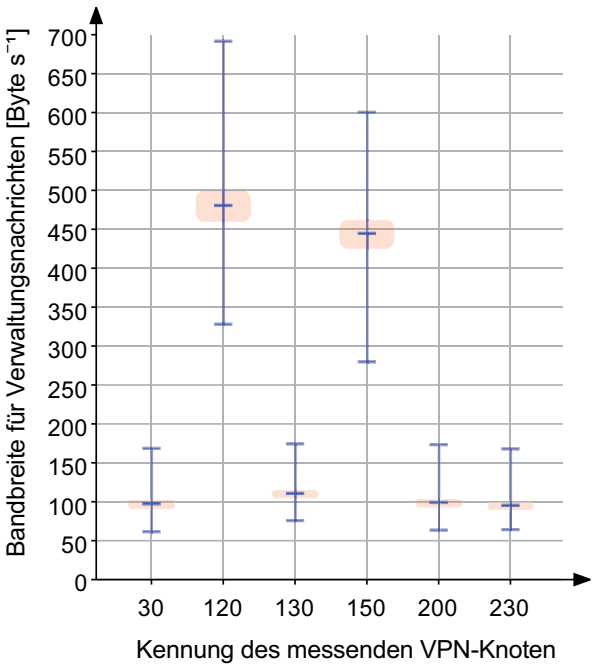


Abbildung 7.5: Durchschnittlich verwendete Bandbreite für Heartbeats und Ringaktualisierungen (Laborszenario)

Der durch proaktive Beziehungen verursachte Nachrichtenaufwand stellt somit für moderne Rechnernetze kein größeres Problem dar, sodass durch diese keine Stausituationen entstehen und die Bandbreiten in den Simulationsszenarien tatsächlich nicht

weiter modelliert werden müssen. Für extrem große VPN oder Knoten mit sehr geringen Bandbreiten mögen diese Werte dennoch ein signifikanter Nachteil sein. Die Werte können zukünftig allerdings optimiert werden, indem beispielsweise bei Heartbeats nicht sämtliche Knoteninformationen, wie beispielsweise alle äußeren IP-Adressen, übertragen werden. Es wäre ausreichend Änderungen mitzuteilen, worauf bisher verzichtet wurde, da diese Pakete dann beispielsweise quittiert werden müssten, sodass die Programmlogik komplexer würde. Ferner ist das Datenformat der Nachrichten bisher nicht optimal: So werden standardmäßig 32-Bit Integer verwendet, und kurze IPv4-Adressen in ihrer IPv6-Repräsentation übertragen. Auch auf diese Optimierungen wurde aufgrund des erhöhten Aufwandes — vor allem bei der Fehlersuche und Analyse des Prototypen — verzichtet. Ferner ist es denkbar die periodisch übertragenen Pakete adaptiv an die Gegebenheiten anzupassen, beispielsweise indem zuverlässige oder selten verwendete Sicherheitsbeziehungen weniger oft auf ihre korrekte Funktion überprüft werden.

7.4 Netzwerksimulationen

Wesentlich komplexere Untersuchungen sind innerhalb des Labornetzes nicht sinnvoll möglich. Daher werden weitere Eigenschaften von SOLID in größeren Szenarien innerhalb des Netzwerksimulators untersucht. Auch hier werden die Extrem- und Mittelwerte angegeben, letztere auch mit einem 99%-Konfidenzintervall. Allerdings wurden, soweit nicht anders angegeben, lediglich 32 Simulationsläufe zur Bildung der statistischen Werte verwendet.

7.4.1 Statische Overlay–Eigenschaften

Einige zentrale Eigenschaften von SOLID-VPN lassen sich durch die Untersuchung der nach dem Einschwingen entstandenen Topologien ableiten. Dazu werden im Folgenden die Verteilung der Knotengrade, die Effizienz der Querverbindungen und die Abweichung von der optimalen Pfadlänge jeweils in unterschiedlichen Szenarien betrachtet.

Verteilung des Knotengrades

Alle SOLID-Knoten eines VPN tragen grundsätzlich gleichmäßig zum Aufbau der Overlay-Struktur bei. Allerdings wird dies beim Aufbau von Querverbindungen durch das Stichprobenkonzept nicht erzwungen, und auch durch die äußeren Transportnetzbedingungen können die Balancierungseigenschaften gestört werden. Daher ist in Simulationen die Verteilung des Knotengrades im direkten und im HOT-Szenario in VPN mit jeweils 250 Knoten ermittelt worden.

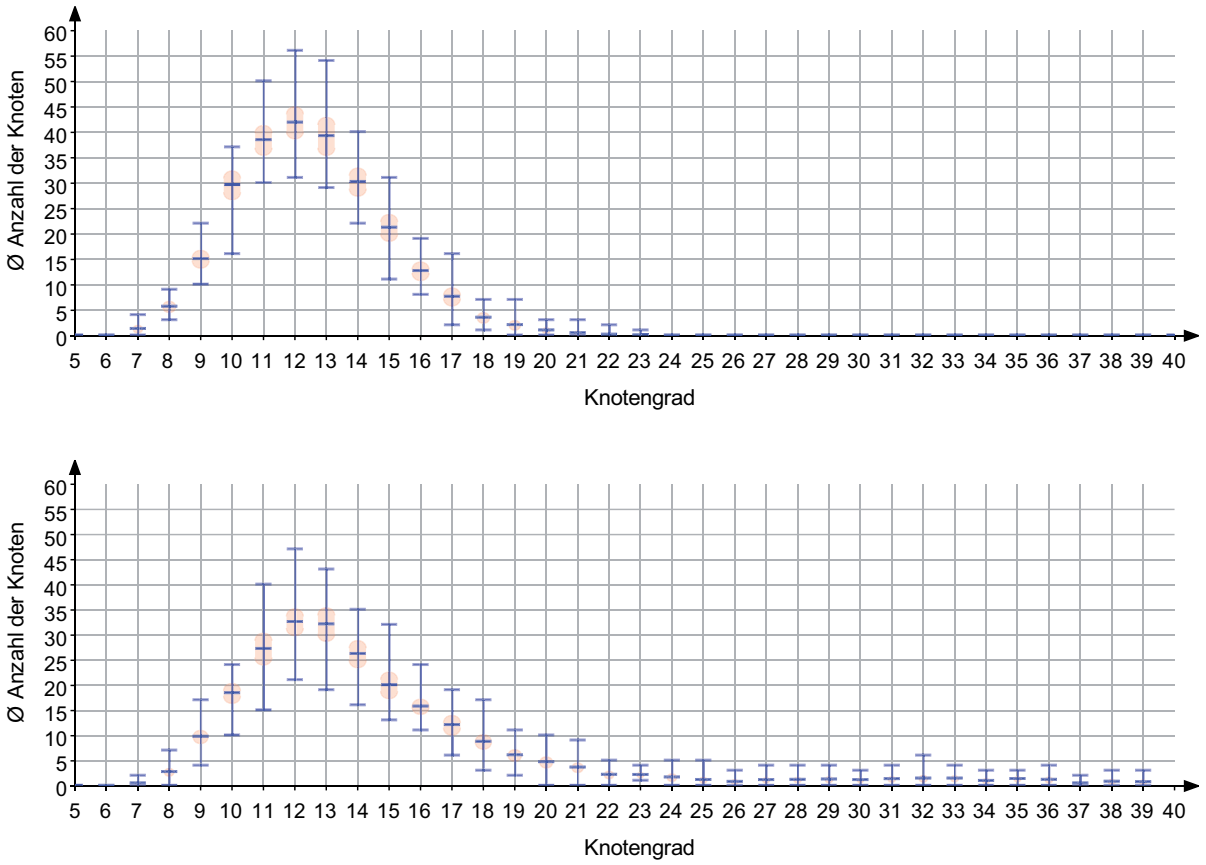


Abbildung 7.6: Verteilung des Knotengrades in direktem Szenario (oben) und HOT-Graph (unten)

Abbildung 7.6 zeigt den Mittelwert, die Extremwerte sowie die Konfidenzintervalle der Verteilungen nach Abschluss der transienten Phase. Die Verteilung im direkten Szenario ähnelt dabei einer Normalverteilung, stimmt mit $\mathcal{N}(12,5;5,77)$ aber nicht statistisch signifikant überein. Die Ursachen sind primär der starke Anstieg der Häufigkeiten bei Knotengraden um zehn, sowie das vereinzelte Auftreten der Knotengrade größer 19. Beide Deformationen sind direkt auf den in Teilabschnitt 4.4.3 vorgestellten stichprobenbasierten Querverbindungsalgorithmus zurückzuführen, da die Knoten immer eine bestimmte Anzahl von Querverbindungen proaktiv aufbauen, wird ein gewisser Mindestknotengrad forciert. Gleichzeitig kann es dabei vorkommen, dass der korrespondierende Knoten bereits eine Verbindung in den entsprechenden Bereich der Stichprobe besitzt und demzufolge mehr Verbindungen aufbaut als nötig. Dies bewirkt die leichte Verschiebung am Ende der Verteilung.

Im HOT-Szenario steigt der Mittelwert des Knotengrades aufgrund der zusätzlichen indirekten Sicherheitsbeziehungen leicht auf 16,12. Als problematisch kann sich die Zunahme der empirischen Varianz auf $s^2 = 85,38$ erweisen: Sie deutet bereits auf das Auftreten von Knoten mit extrem hohen Knotengrad hin. Der maximale Knotengrad, der innerhalb der Messungen auftrat beträgt 111. Zurückzuführen ist dies allerdings allein auf SOLID-Knoten, die eine große Anzahl indirekter Knoten anbinden und zum

optimalen Weiterleiten, in Bezug auf die Anzahl der Hops, direkte Sicherheitsbeziehungen zu allen Nachbarknoten aufbauen müssen.

In der Praxis ist somit durch die leichte Verschiebung in der Knotenverteilungsfunktion kein Einfluss zu erwarten. Die gemessenen hohen Knotengrade müssen in großen, stark fluktuierenden VPN möglicherweise jedoch über leistungsfähigere Systeme abgefangen werden. Andernfalls können zwar auch weiterhin indirekte Sicherheitsbeziehungen über diese Knoten weitergeleitet werden, allerdings werden Topologieoptimierungen, die einen direkten Verbindungsaufbau initiieren, stark verzögert.

Effizienz stochastischer Querverbindungen und des Skip–Graph–Ansatzes

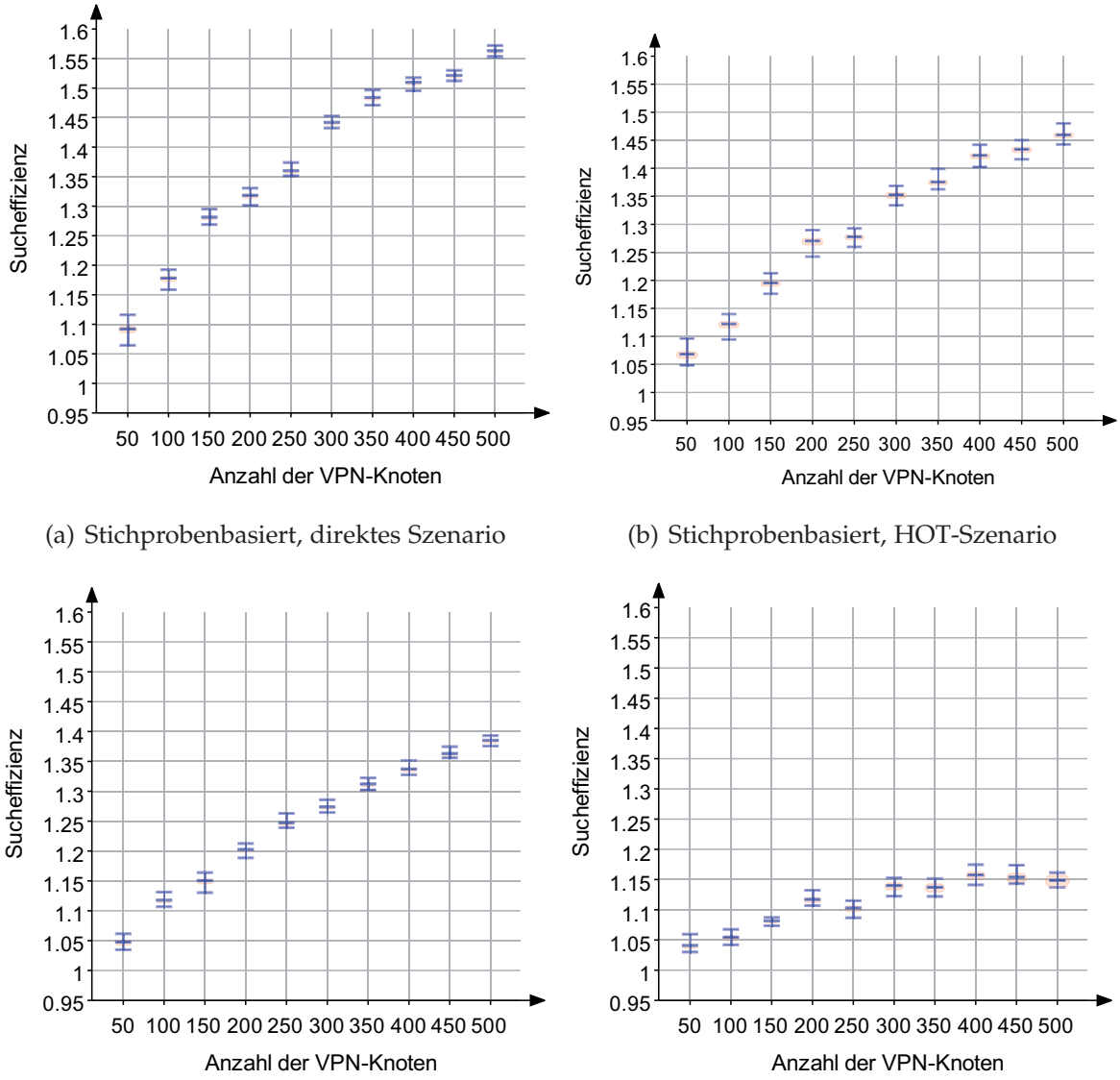
Das Realisieren von Querverbindungen durch das Bilden von Stichproben über den Adressraum wurde in Teilabschnitt 4.4.3 vor allem mit der guten Flexibilität bei der Auswahl der Kommunikationspartner begründet. Die Alternative, der Aufbau von Skip-Graphen, besitzt diese Flexibilität nicht. Zur genaueren Bestimmung des Einflusses dieser Eigenschaft wurde eine vergleichende Implementierung erstellt, und die Unterschiede anschließend in einer experimentellen Untersuchung innerhalb des Netzwerksimulators ermittelt.

Dabei erweist sich die Bewertung der Effizienz als komplex, da beide Ansätze abhängig vom Szenario unterschiedlich viele Querverbindungen aufbauen und somit die durchschnittlich benötigte Anzahl an Overlay-Hops zur Suche nach einem Knoten nicht direkt zu vergleichen ist. Daher wurde eine künstliche Metrik eingeführt: Die Sucheffizienz wird im Folgenden im Vergleich zu einem hypothetischen Overlay-Ansatz gemessen, bei dem die gleiche Anzahl an Knoten über die gleiche Anzahl an zufälligen Querverbindungen gemessen wird:

$$\text{Sucheffizienz} = \frac{\text{Ø Overlay–Hops zur Suche im Zufallsnetz}}{\text{Ø Overlay-Hops zur Suche in untersuchtem Netz}}$$

Nach Verlassen der transienten Phase nach 10.000 Sekunden ergeben sich für die jeweiligen Szenarien stark differierende Sucheffizienzen, wie es in Abbildung 7.7 dargestellt ist. Bei der Wahl auf Basis von Stichproben steigt die Effizienz im direkten Szenario von 1,09 bei 50 Knoten auf 1,56 bei 500 Knoten an. In HOT-Szenarien ist ebenfalls ein streng monotoner Anstieg zu beobachten, allerdings ist die Kurve insgesamt flacher und besitzt ihr Maximum bei 1,46. Der Verlauf der Effizienz im Skip-Graph ist erneut etwas flacher. Das Maximum liegt bei 1,38. Beim Einsatz von Skip-Graphen im HOT-Szenario ist kein streng monotoner Verlauf mehr gegeben: Nach einem langsamen Anstieg der Effizienz auf 1,12 bei 200 Knoten, bildet sich ein Plateau bei einem Verhältnis von etwa 1,15.

Der Anstieg der Effizienz über die Anzahl der Knoten im System ist mit der gewählten Metrik zu begründen, da die zufällige Auswahl von Querverbindungen keine logarithmische Distanz bezüglich der Overlay-Metrik zusichert. Somit fällt mit steigender Anzahl an Overlay-Knoten die Effizienz des Zufallsgraphen. Die leicht schlechtere



(a) Stichprobenbasiert, direktes Szenario

(b) Stichprobenbasiert, HOT-Szenario

(c) Skip-Graph, direktes Szenario

(d) Skip-Graph, HOT-Szenario

Abbildung 7.7: Effizienz bei der Wahl der Querverbindungen

Sucheffizienz des stochastischen Ansatzes in HOT-Graphen ist auf die zusätzlichen Sicherheitsbeziehungen zurückzuführen, welche für die Herstellung des Zusammenhanges und die Weiterleitung benötigt werden. Auch wenn beim Skip-Graph-Ansatz asymptotisch die gleiche Sucheffizienz zu erwarten ist, entwickelt sich der gemessene Werteverlauf bereits im direkten Szenario signifikant schlechter als der auf Stichproben basierende Ansatz. Bedingt wird dies durch die vergleichsweise schlechte Verteilung der Hash-Werte, die zwar asymptotisch den Suchraum in jedem Schritt halbieren, dies aber im Einzelfall nicht unbedingt optimal tun. In verschachtelten Topologien sinkt die Effizienz des Skip-Graph-Ansatzes stark. Zurückzuführen ist dies auf die hohe Anzahl an weiterleitenden Sicherheitsbeziehungen, die dafür benötigt werden, jeden Knoten mit seinen im Skip-Graph benachbarten Knoten zu verbinden. Der stichprobenbasierte Algorithmus kann hingegen wichten, ob eine Querverbindung zu einem weit entfernten Knoten aufgebaut werden muss, oder ob ein bezogen auf

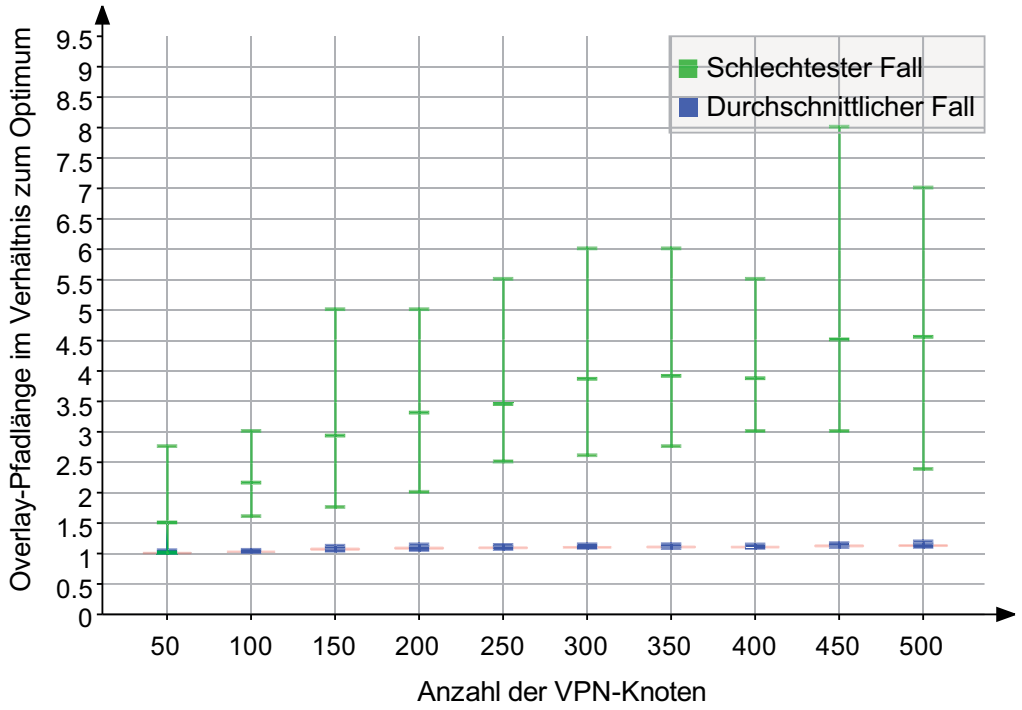


Abbildung 7.8: Durchschnittliches Verhältnis zwischen tatsächlicher Overlay-Pfadlänge und optimaler Länge (vermaschtes HOT-Szenario)

die Overlay-Distanz nicht geringfügig ungünstigerer Knoten in der Nähe verwendet werden kann.

Insgesamt ist der Skip-Graph-Algorithmus somit lediglich für direkte Szenarien zu verwenden, in denen sehr viele, stark fluktuierende Knoten ein VPN bilden. Allerdings sollten dann der Einsatz von Verfügbarkeitszonen und partielle Transportnetzausfälle weitestgehend ausgeschlossen werden, da diese auch in solchen Netzen zum unnötigen Aufbau vieler indirekter Beziehungen führen.

Pfadlängen in nicht–optimalen Transportnetzen

Die Kombination aus SOLID's Routing und Topologiekontrolle garantiert für Transportnetztopologien, in denen die einzelnen Teilnetze ohne Zyklen verbunden sind, eine Weiterleitung auf Pfaden mit optimaler Anzahl an Overlay-Hops (siehe Abschnitt 4.5.3). Wird die Kreisfreiheit nicht gewährleistet, werden unter Umständen lediglich lokale Optima erreicht, da es nicht von Transportnetz- und Overlay-Topologie abhängt, welcher der Pfade gewählt wird. Dies suggeriert, dass SOLID in anderen Topologien möglicherweise lediglich schlechte Wege findet.

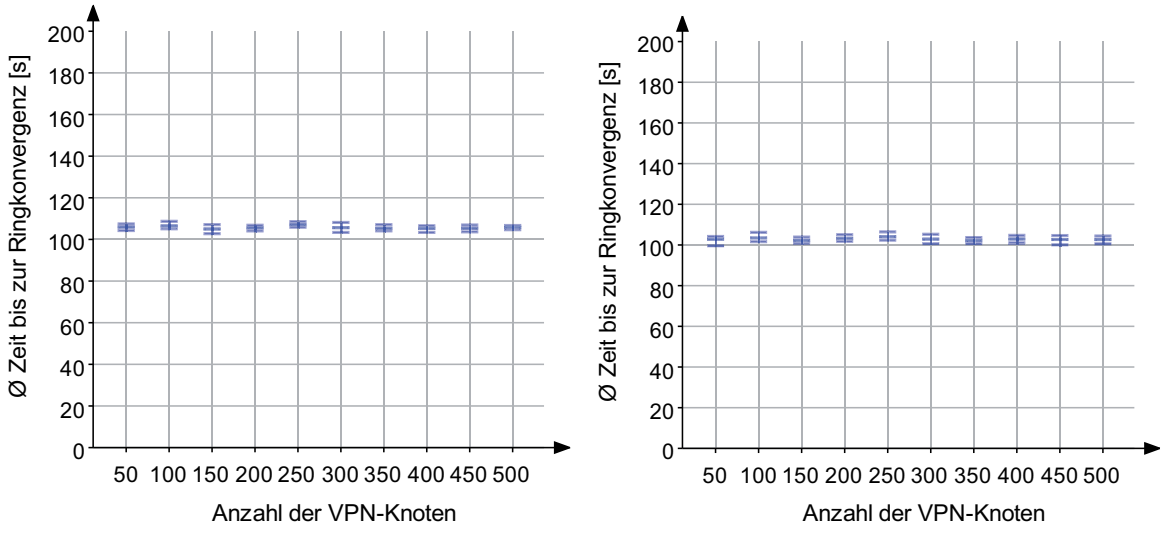
Zur nähren Untersuchung wurden in Simulationen die HOT-Graphen vollständig mit SOLID-Knoten versehen, sodass im Kernbereich ein vermaschtes Netz entsteht. Nach einer Startperiode von 3000 Sekunden wurden alle proaktiv aufgebauten Pfade mit mehr als einem Hop ausgegeben und mit der Länge der jeweils optimalen Pfade ins Verhältnis gesetzt.

Abbildung 7.8 zeigt das Ergebnis des Experiments: Das durchschnittliche Verhältnis zwischen erreichter und optimaler Pfadlänge ist in allen untersuchten Netzgrößen nahezu eins. Maxima, Minima, Mittelwerte und Konfidenzintervalle gehen in der Darstellung fließend ineinander über. Die Länge des ungünstigsten Pfades der einzelnen Simulationen steigt kontinuierlich mit der Größe des VPN von knapp 300% auf 800%. Die einzelnen Werte variieren dabei topologieabhängig relativ stark. Unter anderem weil der zentrale Grenzwertsatz keine Anwendung findet, liegt dadurch auch keine Normalverteilung vor, und es ist auf die Berechnung der Konfidenzintervalle verzichtet worden.

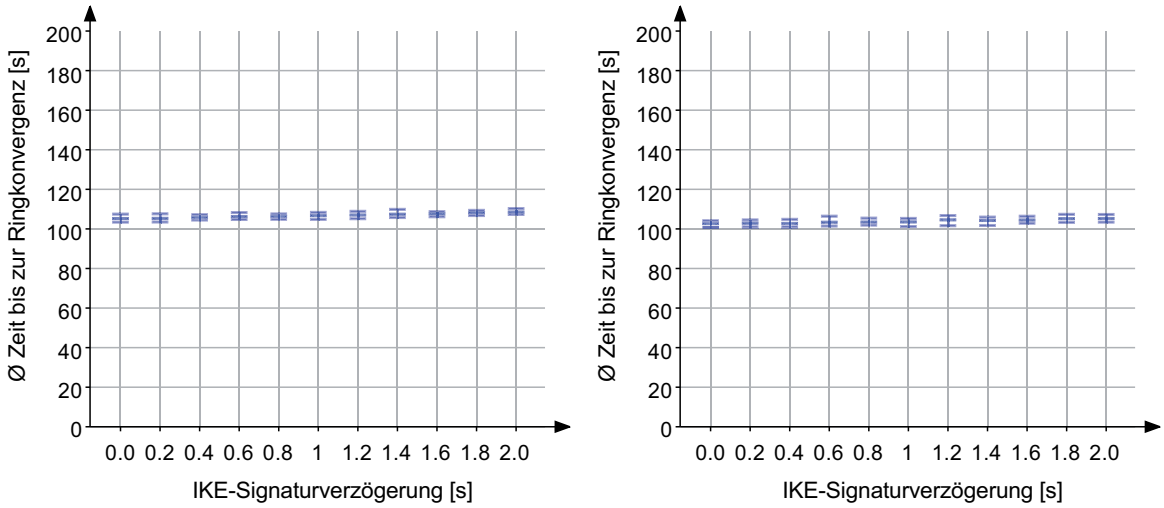
Zumindest im Mittel werden die schlechteren Eigenschaften daher — zumindest in HOT-ähnlichen Netzen — kaum ins Gewicht fallen. Lediglich einige wenige Beziehungen bedürfen unter Umständen zusätzlicher Optimierungsmechanismen. Zukünftig ist hier beispielsweise ein hybrides Verfahren angedacht, bei dem SOLID für schnell verändernde Sicherheitsbeziehungen eingesetzt wird, und ein herkömmliches Routing-Verfahren für langlebige Assoziationen zum Einsatz kommt. Unter Umständen können auch die stochastisch versendeten Ringaktualisierungen durch die Aufzeichnung von Pfadvektoren zur Erkennung kürzerer Wege verwendet werden, sodass Zyklen erkannt werden können ohne dass zusätzliche Pakete generiert werden müssen. Allerdings sind beide Verfahren aus Sicherheitssicht kritisch zu bewerten: Durch die Einführung von Routing-Annoncierungen wird das System potentiell anfälliger gegen Sink- und Wormhole-Angriffe. Wie beispielsweise bereits für das Secure Border Gateway Protocol (S–BGP) [KLS00] vorgeschlagen, müssen die Pfade daher in jedem Schritt signiert werden, um zumindest gegen einzelne kompromittierte Knoten einen guten Schutz zu erhalten. Der Effizienzgewinn rechtfertigt den entstehenden Verwaltungsaufwand daher wahrscheinlich nur in wenigen Szenarien.

7.4.2 Einfügen und Entfernen von Knoten

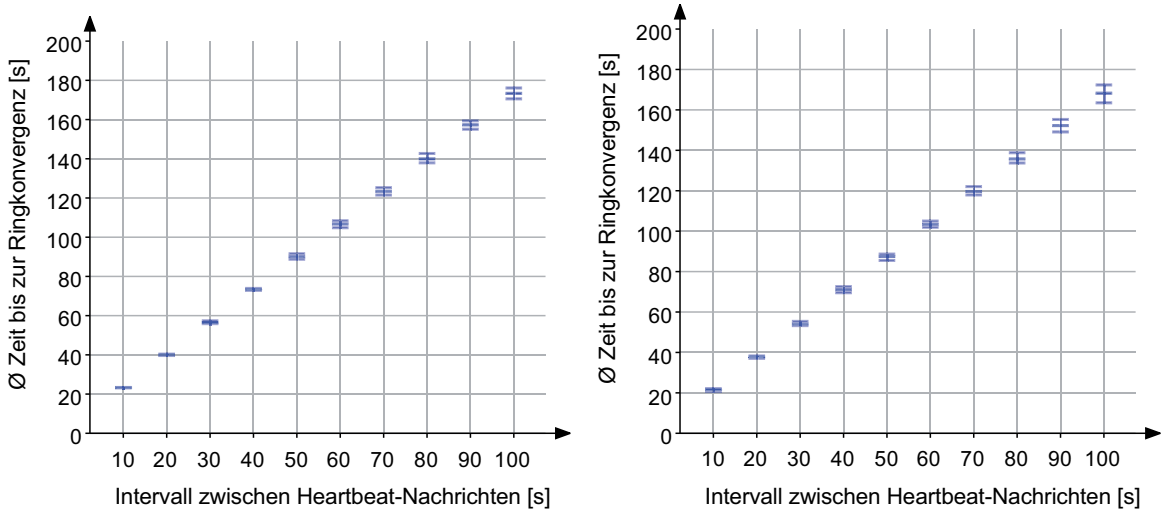
Da der entwickelte Autokonfigurationsmechanismus seine Vorteile vor allem in dynamischen VPN entfalten kann, ist eine Erfassung des dynamischen Verhaltens im regulären Einsatz interessant. Vor allem die verschiedenen Einflussfaktoren wie VPN-Größe, Netzwerkverzögerung und IKE-Verzögerung gilt es dabei zu isolieren und zu vergleichen. In jeweils 32 Experimenten wurden daher einzelne Umgebungsvariablen des SOLID-VPN verändert, und der Einfluss auf die durchschnittliche Dauer des Einfügens und Entfernens zufälliger Knoten gemessen. Jede Simulation dauerte 100.000 simulierte Sekunden, wobei nach 1.000 Sekunden der erste Knoten entfernt wurde. Nach erfolgter Stabilisierung des Ringes und einer Stabilisierungsphase von 100 Sekunden wurde er wieder eingefügt und die Zeit bis zur Ringstabilisierung gemessen. Anschließend wurden nach Stabilisierungsperioden von 100 simulierten Sekunden erneut Knoten entfernt und wiedereingefügt. Im HOT-Szenario sind dabei lediglich SOLID-Knoten entfernt worden, über die keine indirekten Nachbarn eingebunden werden. Andernfalls wäre es durch das Entfernen zu Partitionierungen gekommen, deren Behebung beim erneuten Einfügen des Knotens in einigen Fällen längere Konvergenzzeiten benötigt und somit separat untersucht werden muss. Soweit nicht anders



(a) (b)



(c) (d)



(e) (f)

Abbildung 7.9: Durchschnittliche Dauer des Entfernens eines Knotens in Abhängigkeit von der VPN-Größe, IKE-Verzögerung und der Heartbeat-Periode (direktes Szenario links, HOT rechts)

angegeben, beträgt in den Simulationen die VPN-Größe immer 250 Knoten, die Verzögerung durch IKE-Signaturen in jedem Knoten eine Sekunde und das Heartbeat-Intervall 60 Sekunden.

In Bezug auf die Dauer der Stabilisierung nach dem Entfernen eines Knoten ergeben sich die Einflussfaktoren, wie sie in Abbildung 7.9(a) bis 7.9(f) dargestellt sind: Weder die VPN-Szenarien, noch die VPN-Größen oder die IKE-Signaturverzögerungen haben einen signifikanten Einfluss auf die Dauer zwischen dem Entfernen eines Knotens und dem Zeitpunkt der Ringreparatur an der fragmentierten Stelle. Bei einer Variation der VPN-Größe ist kein Trend feststellbar; der Einfluss der IKE-Signaturverzögerung ist in etwa genauso groß wie die Verzögerung selbst. Es vergehen im direkten Szenario im Mittel jeweils etwa 106 simulierte Sekunden bis zur Konvergenz. Im indirekten Szenario vergehen etwa 103 Sekunden.
Signifikant ist einzig der Einfluss des Intervalls zwischen zwei Heartbeat-Nachrichten: Hier existiert ein nahezu proportionaler Zusammenhang, der auch in beiden Szenarien identisch ist. Im Schnitt dauert es etwa 1,5 Heartbeat-Perioden bis beide Ringnachbarn erkennen, dass ein Fehler vorlag. Hinzu kommen fünf erneute Übertragungen im Abstand von etwa einer Sekunde und die Zeit zum Aufbau einer neuen Sicherheitsbeziehung.
Die Konfidenzintervalle waren bei der Messung in allen Fällen so klein, dass auf eine Darstellung verzichtet worden ist und lediglich Maxima, Minima und Mittelwerte angegeben werden.

Beim Einfügen neuer Knoten sind die Ergebnisse in Abbildung 7.10(a) bis 7.10(f) in den beiden Szenarien ebenfalls vergleichbar. Stärkstes Unterscheidungsmerkmal sind die leicht größeren Schwankungen im HOT-Szenario, die zum einen auf die Topologievariationen und zum anderen die auf unterschiedlichen Verschachtelungstiefen der eingefügten Knoten zurückzuführen ist.

Die Größe des VPN hat einen sehr geringen Einfluss, so steigt die mittlere Dauer des Einfügens eines neuen Knoten im direkten Szenario von 7,7 Sekunden bei 50 Knoten lediglich auf 8,2 Sekunden bei 500 Knoten. Im HOT-Szenario steigt der Mittelwert ebenfalls kaum von 10,2 auf 12,2 Sekunden an. Die etwas sprunghaften Anstiege sind hier auf den Anstieg des maximalen Knotengrades vorgelagerter Knoten zurückzuführen.
Um zu untersuchen, wie sich die Verzögerung des Transportnetzwerks auf das Einfügen auswirkt, wird die in den Szenarien simulierte Ende-zu-Ende-Verzögerung durch einen Faktor von 0,0 bis 2,0 gestaucht beziehungsweise gestreckt. Die Ergebnisgraphiken 7.10(c) und 7.10(d) zeigen, dass ein Einfluss messbar ist: Die Dauer des Einfügens variiert im direkten Szenario zwischen 6,7 und 8,9 Sekunden. Im indirekten Fall steigt die Dauer ebenfalls um knapp 2,5 Sekunden.
Trotz eines gewissen Einflusses ist die Transportnetzverzögerung, wie bereits in Teilabschnitt 7.2.3 vermutet, nicht der signifikanteste Faktor: Die Variation der Verzögerung von IKE-Signaturen zwischen 0,0 und 2,0 Sekunden führt im direkten Szenario zu einem schnellen Anstieg der Einfügeperiode von 2,1 auf 16,4 Sekunden. Der Anstieg

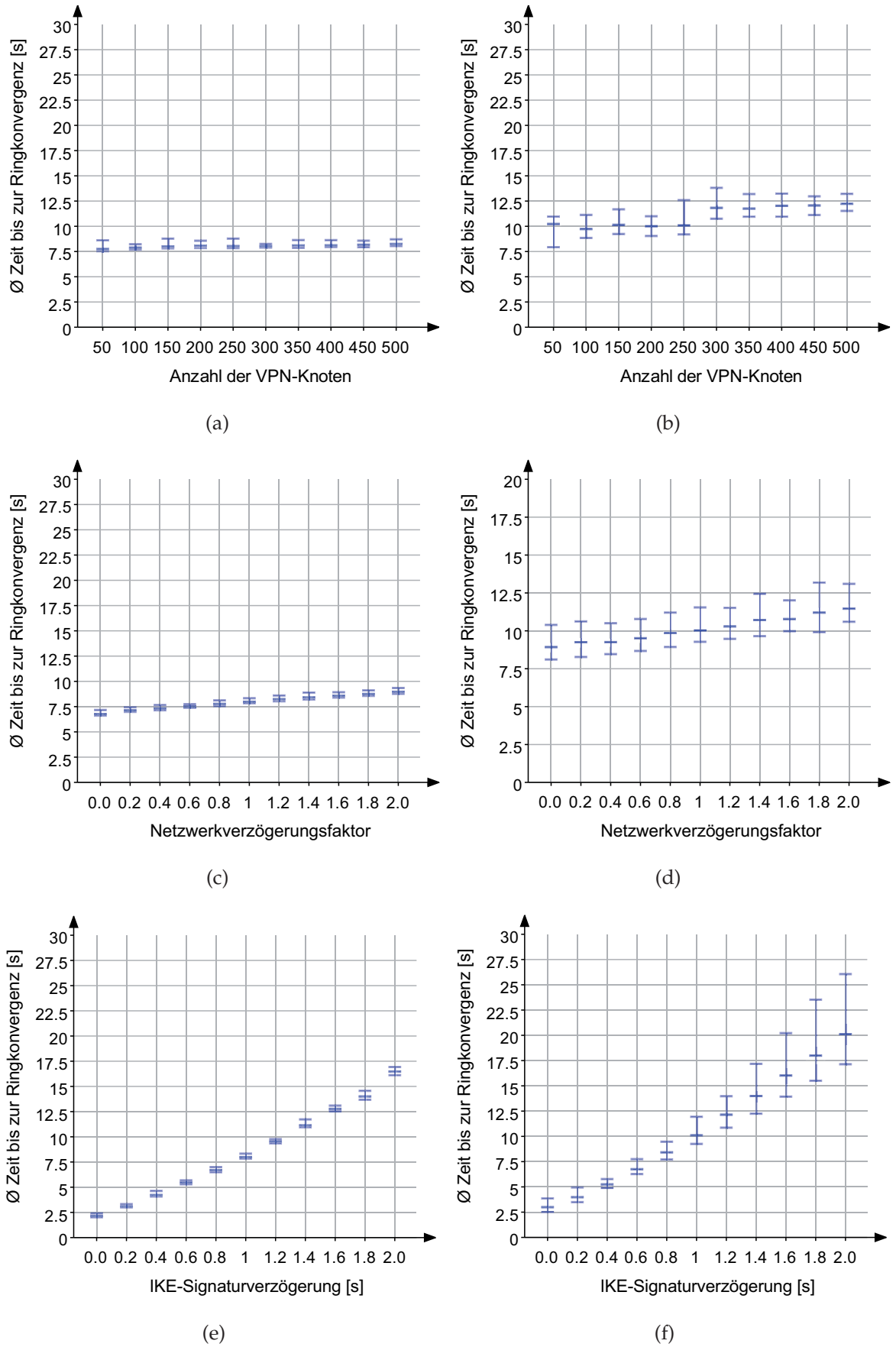


Abbildung 7.10: Durchschnittliche Dauer des Einfügens eines neuen Knotens in Abhängigkeit von der VPN-Größe, Transportnetzverzögerung und IKE-Verzögerung (direktes Szenario links, HOT rechts)

ist dabei sogar mehr als linear, was auf vermehrte Zeitüberschreitungen und Neuübertragungen zurückzuführen ist. Im indirekten Fall steigt der Mittelwert von etwa 2,9 auf 20,1 Sekunden, wobei auch Maximum und Minimum sich trotz der vergleichsweise langen Simulationsdauer stark auffächern.

Insgesamt erweist sich somit beim Entfernen von Knoten lediglich die Heartbeat-Periode als signifikanter Einflussfaktor. Die IKE-Verzögerung spielt durch die langen Perioden bis zur Ringkonvergenz, anteilig eine sehr geringe Vor dem Hintergrund, dass mobile Knoten im VPN verwendet werden können, bietet es sich an, die Heartbeat-Periode bei diesen Knoten dynamisch anzupassen, um schneller auf ihren Austritt reagieren zu können. Beispielsweise indem eine anfängliche Periode von einer Sekunde langsam über den Zeitraum von mehreren Stunden auf den jetzigen Standardwert von 60 Sekunden adaptiert wird. Hinsichtlich des Einfügens von Knoten hat sich im direkten Szenario die IKE-Signaturverzögerung als limitierender Faktor bestätigt. Neben schnelleren Smartcard-Prozessoren lassen sich jedoch auch hier Detailverbesserungen vornehmen: Beispielsweise kann der IKE-Daemon so modifiziert werden, dass der Aufbau von Ringverbindungen mit einer höheren Priorität erfolgt, wodurch eine schnellere Konvergenz erfolgt. In HOT-Szenarien verhält sich der Einfluss der IKE-Signaturen ähnlich, wenn Randknoten betrachtet werden. Beim Entfernen von Knoten, die Gateways für indirekte Knoten sind, können Partitionen auftreten, die aufgrund ihres abweichenden Verhaltens im folgenden Abschnitt gezielt untersucht werden.

7.4.3 Verhalten bei Transportnetz–Partitionierungen

Bei der Wiederherstellung nach Partitionierungen der Overlay-Struktur sind zwei Teilprobleme zu betrachten. Zum einen müssen die vormals getrennten VPN-Teile durch mindestens eine Sicherheitsbeziehung verbunden werden. Zum anderen muss, wie in Teilabschnitt 4.6.3 näher erläutert, die Struktur des Overlays wiederhergestellt werden. Da Partitionierungen bereits durch den Ausfall eines zentralen VPN-Knotens entstehen können, kann die sichere Wiederherstellung eine essentielle Aufgabe sein. Auch hier gilt es, signifikante Einflussfaktoren zu identifizieren. Daher wurden in Simulationsexperimenten im direkten Szenario die Knoten alle 2.500 Sekunden zufällig Partitionen zugeordnet, und jegliche Kommunikation zwischen den Partitionen unterbunden. Aufgrund der Zeitüberschreitungen und der daraufhin eingeleiteten Reparaturen, war nach 500 Sekunden in jeder Simulation ein Zerfall der Ringstruktur in mindestens zwei kleinere Ringe zu beobachten. Daher wurde anschließend die Partitionierung aufgehoben und die Konvergenzzeiten in Bezug auf VPN-Zusammenhang und erneute Ausbildung der Ringstruktur gemessen. Für jeden Einzelversuch wurde der Mittelwert über zehn Messungen bestimmt. Dabei wurden in den einzelnen Experimenten Parameter variiert. Soweit nicht anders angegeben, beträgt das maximale Bootstrap-Intervall allerdings abweichend von anderen Experimenten 25 Sekunden und das VPN wurde in zwei Teile partitioniert.

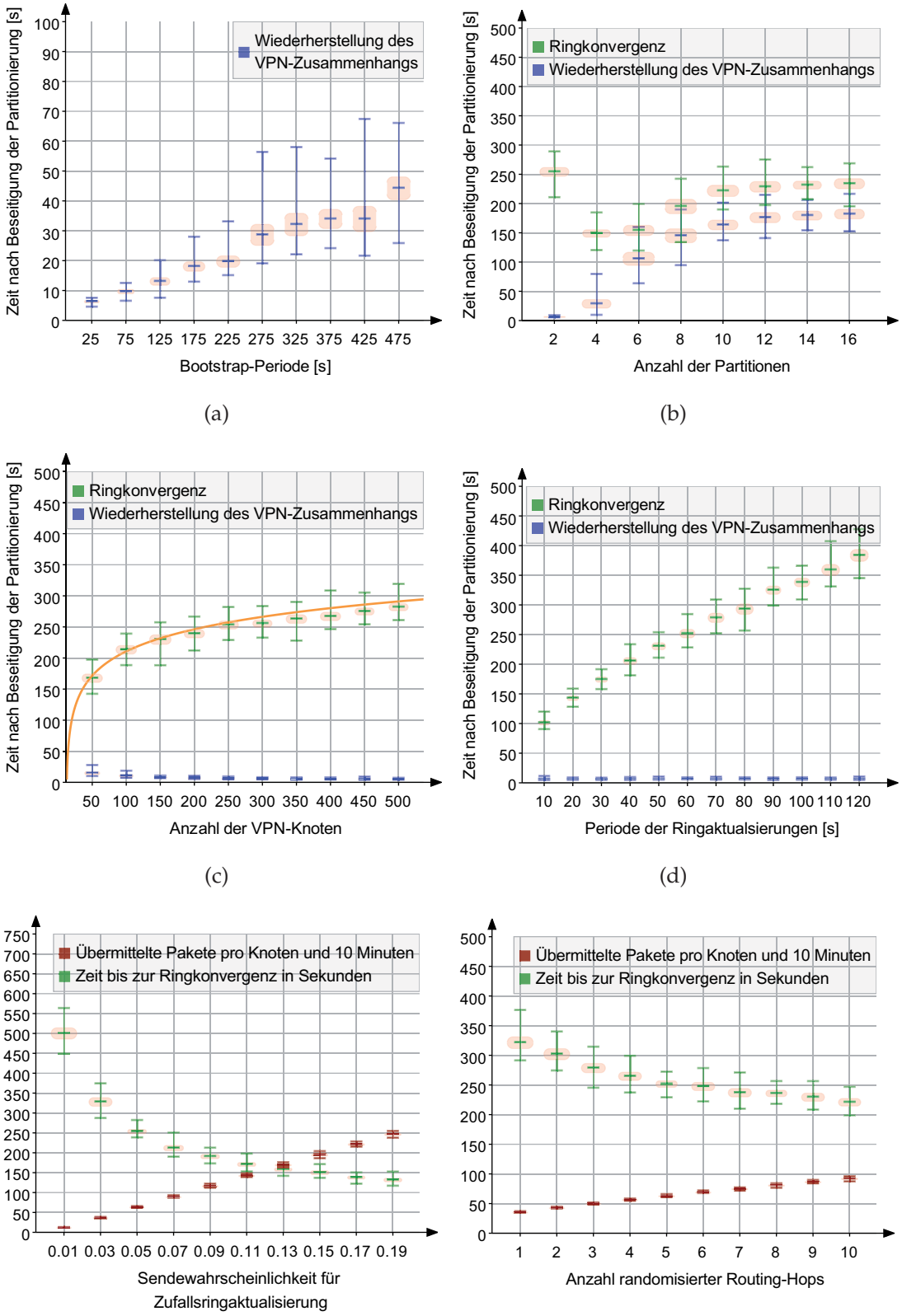


Abbildung 7.11: Durchschnittliche Dauer der Wiederherstellung nach Transportnetzpartitionierungen in Abhängigkeit von der Bootstrap-Periode, der Anzahl der Partitionen, der Größe des VPN, dem Ringaktualisierungsintervall, der Wahrscheinlichkeit von Zufallsaktualisierungen und der Anzahl der Zufallsschritte (direktes Szenario)

Bei der Wiederherstellung des VPN-Zusammenhangs liegt die These nahe, dass der dafür benötigte Zeitraum nahezu allein von der Häufigkeit abhängt, mit der Sicherheitsbeziehungen über das Bootstrapping aufgebaut werden. Die Verzögerungen durch Transportnetzweiterleitungen und IKE-Signaturen sind zu vernachlässigen, da der durch periodisches Bootstrapping ausgelöste, proaktive Aufbau von Sicherheitsbeziehungen aus Effizienzgründen normalerweise nicht sehr häufig geschieht. Die Resultate der Simulationen in Abbildung 7.11(a) bestätigen diese Annahme: Es liegt ein nahezu proportionaler Zusammenhang zwischen Bootstrap-Intervall und Zeit bis zur Wiederherstellung des VPN-Zusammenhangs vor. Bei den untersuchten Intervallen von 25 bis 475 Sekunden steigt die Zeit zur Wiederherstellung von etwa 6 auf 44 Sekunden.

In Bezug auf die Anzahl der Partitionen liegt die Vermutung nahe, dass eine größere Anzahl an Partitionen eine längere Konvergenzperiode bewirkt. Abbildung 7.11(b) zeigt jedoch, dass dies nicht zwangsläufig sein muss. Die Dauer bis zur Wiederherstellung des Zusammenhangs steigt zwar bei etwa acht Partitionen schnell an, stabilisiert sich allerdings anschließend bei etwa 180 Sekunden. Zurückzuführen ist dies auf das vermehrte Auftreten von Partitionen, in denen sich lediglich sehr wenige Knoten befinden, sodass bereits der Aufbau sehr weniger Sicherheitsbeziehungen genügt, mehrere Partitionen zusammenzuführen. Entgegen der Intuition, sinkt die Zeit bis zur Ringkonvergenz zunächst von etwa 250 Sekunden bei zwei Partitionen auf 149 Sekunden bei vier Partitionen. Anschließend steigt sie auf 220 Sekunden bei zehn Partitionen, wo sie sich stabilisiert. Dieser Effekt ist auf ein ähnliches Phänomen zurückzuführen, wie bei der Wiederherstellung des Knotenzusammenhangs. Je mehr Partitionen entstehen, desto mehr Dynamik entsteht im VPN, sodass beispielsweise häufiger Ringaktualisierungen weitergeleitet werden müssen. Sind die dabei ausgetauschten Knoteninformationen für die Topologiekontrolle von Interesse, beispielsweise wegen der Nähe des anfragenden Knotens in der Ringstruktur, initiiert auch der weiterleitende Knoten Ringaktualisierungen. Bei vier Partitionen überwiegt der Effekt sogar deutlich den entstehenden Mehraufwand zur Reparatur.

Der Mehraufwand, der in größeren VPN entsteht, ist durch die Anzahl der aufzubauenden Beziehungen und die Overlay-Pfadlängen in der Regel durch logarithmische Größen beschränkt. Bei der Konvergenz nach Partitionierungen führt sich dieses Verhalten fort, wie es in Abbildung 7.11(c) durch die approximierte Beschränkung $47 \cdot \ln(x)$ gezeigt ist. Die Dauer bis zur Ringkonvergenz erhöht sich von 167 Sekunden bei 50 Knoten lediglich auf 281 Sekunden bei 500 Knoten. Zumindest bei zwei Partitionen erfolgt die Wiederherstellung des Knotenzusammenhangs sogar schneller, wenn das VPN größer ist, da die Wahrscheinlichkeit wächst, dass durch zufällige Verbindungen die VPN-Teile wieder verbunden werden.

Um eine schnellere Konvergenz der Ringverbindungen zu erreichen, kann beispielsweise die Anzahl der versendeten Ringaktualisierungen erhöht werden. Abbildung 7.11(d) zeigt die Auswirkungen dieser Maßnahme, wobei ein nahezu linearer Anstieg von 101 Sekunden bei einem Aktualisierungsintervall von 10 Sekunden auf 383 Sekunden bei einem Intervall von 120 Sekunden zu beobachten ist.

In Teilabschnitt 4.6.3 wurde erläutert, dass das einfache Senden von Ringaktualisierungen nicht ausreicht, wenn Partitionierungen vorliegen. Daher wird in SOLID bei Beibehaltung der Standardeinstellungen mit Wahrscheinlichkeit 0,05 eine Ringaktualisierung über eine zufällige Sicherheitsbeziehung versendet. Werden diese Aktualisierungen häufiger übermittelt, ist davon auszugehen, dass auch die Ringkonvergenz schneller erfolgt. Wie in Abbildung 7.11(e) gezeigt, sinkt die Zeit bis zur erfolgten Ringkonvergenz von 500 Sekunden bei Wahrscheinlichkeit 0,01 zunächst schnell ab. Ab einer Wahrscheinlichkeit von etwa 0,05 verlangsamt sich das Abfallen bis bei einer Wahrscheinlichkeit von 0,19 ein Wert von 131 Sekunden erreicht wird. Die Funktion scheint insgesamt einer Exponentialfunktion mit negativem Exponenten zu folgen. Mit abfallender Konvergenzzeit nimmt die Anzahl der zusätzlich gesendeten und weitergeleiteten Pakete linear zu. Die Gesamtanzahl kann mit maximal 1478 Paketen pro Stunde und Knoten vergleichsweise hoch werden, sodass als Standardwert eine Aktualisierungswahrscheinlichkeit von 0,05 eingestellt ist. Es entsteht ein Aufwand von etwa 375 Paketen pro Stunde und Knoten.

Analog kann die Anzahl der zufällig ausgewählten Overlay-Schritte variiert werden. Abbildung 7.11(f) zeigt jedoch, dass dies im Vergleich zur Aktualisierungswahrscheinlichkeit nur beschränkten Einfluss hat, da bereits durch die Auswahl nur eines Zufallsknotens die Aktualisierung einen vollständig anderen Weg durch das Overlay nehmen kann. Dies spiegelt sich auch im relativ hohen Aufwand wider, den die Übermittlung und Weiterleitung in diesem Basisfall erzeugt (209 Pakete pro Stunde und Knoten). Bei zehn zufälligen Schritten sinkt die Ringkonvergenz um 100 Sekunden, erzeugt aber 550 Pakete pro Stunde und Knoten.

Insgesamt ist es SOLID demzufolge möglich, Partitionierungen auch in großen VPN und bei sehr vielen Knoten sicher zu beseitigen. Je nach Aggressivität, mit der Sicherheitsbeziehungen aufgebaut und Ringaktualisierungen versendet werden, kann sich die Beseitigung von Partitionierungen im Vergleich zur Behandlung normaler Knotendynamik allerdings als langwieriger erweisen. Partitionierungen treten jedoch in realen Netzen vergleichsweise selten ein, sodass gebräuchliche Peer-to-Peer-Systeme sie beispielsweise gar nicht behandeln.

Sollen dennoch schnellere Konvergenzzeiten erreicht werden, bietet es sich an, SOLID um adaptive Mechanismen zu erweitern. Beispielsweise könnten zufällige Ringaktualisierungen bevorzugt über durch Bootstrapping neu aufgebaute Sicherheitsbeziehungen versendet werden, oder die IP-Adressen von Knoten, die vormals Ringnachbarn waren oder zu denen lange Beziehungen existiert haben, häufiger durch Bootstrap-Nachrichten kontaktiert werden.

7.4.4 Verhalten bei partiellen Ausfällen

Neben Ausfällen durch Partitionierungen werden in Abschnitt 4.5.3 auch partielle Ausfälle betrachtet. Im Gegensatz zur vollständigen Separierung ist hier noch eine Kommunikation über dritte Knoten möglich, wobei zu untersuchen ist, wie gut die gefundenen Umleitungen in Bezug auf eine geringe Verzögerung sind, und welchen Einfluss die einzelnen Optimierungsverfahren haben.

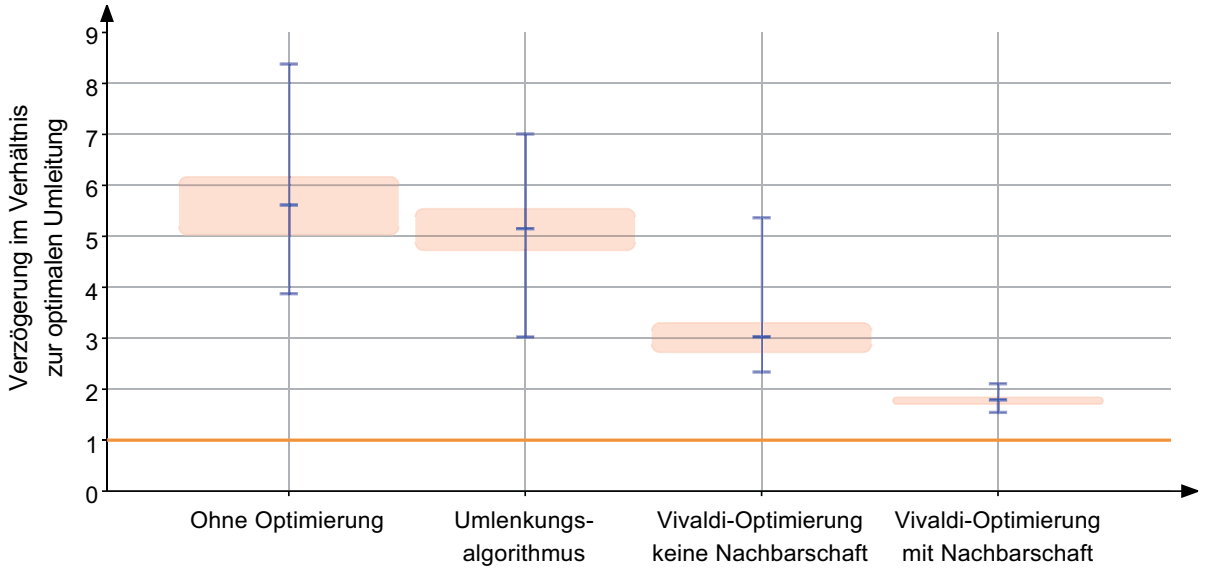


Abbildung 7.12: Güte der Verzögerungszeiten von Umleitungen bei gestörter Kommunikation zwischen einzelnen Knoten in Abhängigkeit der Optimierungsmaßnahmen

Zur Evaluierung sind erneut Netzwerksimulationen im direkten Szenario eingesetzt worden. Zwischen den 250 SOLID-Knoten ist alle 2.000 Sekunden eine zufällige Kommunikationsbeziehung unterbunden worden, und nach der Periode wurde der Pfad der entstandenen indirekten Ersatzbeziehung mit dem optimalen Pfad ins Verhältnis gesetzt. Die Simulationsdauer betrug insgesamt 200.000 Sekunden.

Abbildung 7.12 zeigt das Ergebnis der untersuchten Szenarien: Weiterleitung ohne jegliche Optimierung, Optimierung allein durch den Umlenkungsalgorithmus, Optimierung mithilfe von Vivaldi-Koordinaten und der zusätzliche Einsatz von Nachbarschaftsbeziehungen auf Basis der modifizierten Delaunay-Triangulation. Ohne jegliche Optimierung beträgt die Verzögerung der gefundenen Umleitung im Schnitt das 5,6-fache des Optimums. Wird zusätzlich der Umlenkungsalgorithmus verwendet, sinkt das Verhältnis leicht auf 5,1, da die Pfade nun in Bezug auf die Anzahl an Overlay-Hops optimal gewählt werden. Durch die Einführung von Vivaldi-Koordinaten und die anschließende Verlagerung des Weiterleitungspunktes anhand dieser Koordinaten sinkt das Verhältnis auf etwa 300%, wobei zu berücksichtigen ist, dass das beobachtete Maximum bei 5,3 lag. Dies ist hauptsächlich auf lokale Optima bei der Optimierung zurückzuführen, wie sie in Teilabschnitt 4.6.4 erläutert wurden. Werden zusätzliche Sicherheitsbeziehungen aufgebaut, um eine Delaunay-Triangulation des Koordinatenraumes durchzuführen, sinkt das Verhältnis durch Vermeidung der lokalen Minima im Schnitt auf 1,8. Ferner liegen Maximum und Minimum vergleichsweise eng beieinander, sodass davon auszugehen ist, dass die Optimierung stabiler erfolgt.

Dass nicht immer optimale Weiterleitungspunkte gefunden werden, ist vor allem auf zwei Faktoren zurückzuführen: Der quantitativ größte Fehler ist auf die suboptimale

Einbettung der Meridian-Verzögerungszeiten [WSS05] in das Vivaldi-Koordinatensystem zurückzuführen. Ferner betreffen die Störungen auch die Nachbarschaftsbeziehungen, und in seltenen Fällen wird gerade die ausgefallene Kante für eine weitergehende Optimierung benötigt. In diesen Situationen muss zunächst die Delaunay-Struktur repariert werden, was ohne weitere Optimierungen länger als die evaluierten 2.000 Sekunden dauern kann.

Für eine weitere Verbesserung der erreichten Latenzen ist es demzufolge sinnvoll, hauptsächlich die Einbettung in den Koordinatenraum zu verfeinern. Beispielsweise kann dies mit Filtern oder einer Akkumulation der Differenzen zur Nachbarschaft vor jeder Aktualisierung der eigenen Koordinate erfolgen, wie bereits in [GRS+10a] vorgeschlagen.

7.4.5 Sabotageresistenz

In den folgenden Teilabschnitten wird die Sabotageresistenz von SOLID-Topologien untersucht. Dazu wird zunächst auf die Beobachtung einzelner Knoten eingegangen, wobei sich die dabei zu gewinnenden Strukturinformationen durch die Einführung von Verfügbarkeitszonen stark senken lassen. Daher erfolgt ausschließlich eine quantitative Evaluierung des Einflusses der Verfügbarkeitszonen auf die Effizienz des VPN, im Speziellen auf die Auswirkungen bezüglich der Ende-zu-Ende-Latenz. Im letzten Teilabschnitt werden die Auswirkungen von kooperativen Beobachtungen und Angriffen analysiert und anhand der in Abschnitt 5.2 vorgestellten Anfälligkeit gegen Sabotage bewertet.

Auswirkungen der Beobachtung einzelner Knoten

In Kapitel 5 wurde die Sabotageresistenz über potentiell beobachtete Knoten und den Einfluss von DoS-Angriffen auf deren Nachbarschaft definiert. Wie in Abschnitt 7.4.1 bezüglich des Knotengrades gezeigt wurde, bauen SOLID-VPN lediglich logarithmisch viele Sicherheitsbeziehungen auf. Ohne den Einfluss von Nutzdaten zu betrachten, ist somit davon auszugehen, dass logarithmisch viele Knoten beobachtet werden können. Allerdings werden aufgrund der erzwungenen Dynamik, um potentielle Partitionen aufzulösen, immer wieder Sicherheitsbeziehungen aufgebaut und dadurch beispielsweise auch Querverbindungen leicht variiert. Angreifer, die einen Knoten langfristig beobachten, können daher auch potentiell mehr Nachbarn aufdecken.

In einem Experiment wurde der durchschnittliche Knotengrad über alle Sicherheitsbeziehungen betrachtet, die von jedem Knoten innerhalb der untersuchten Zeitspanne aufgebaut wurden. Die einzelnen Simulationen liefen jeweils in einem direkten Szenario mit 250 Knoten über einen Zeitraum von 300.000 Sekunden, und es wurden Intervalle bis 259.200 Sekunden (72 Stunden) betrachtet. Ein Einfluss der Startphase kann somit ausgeschlossen werden.

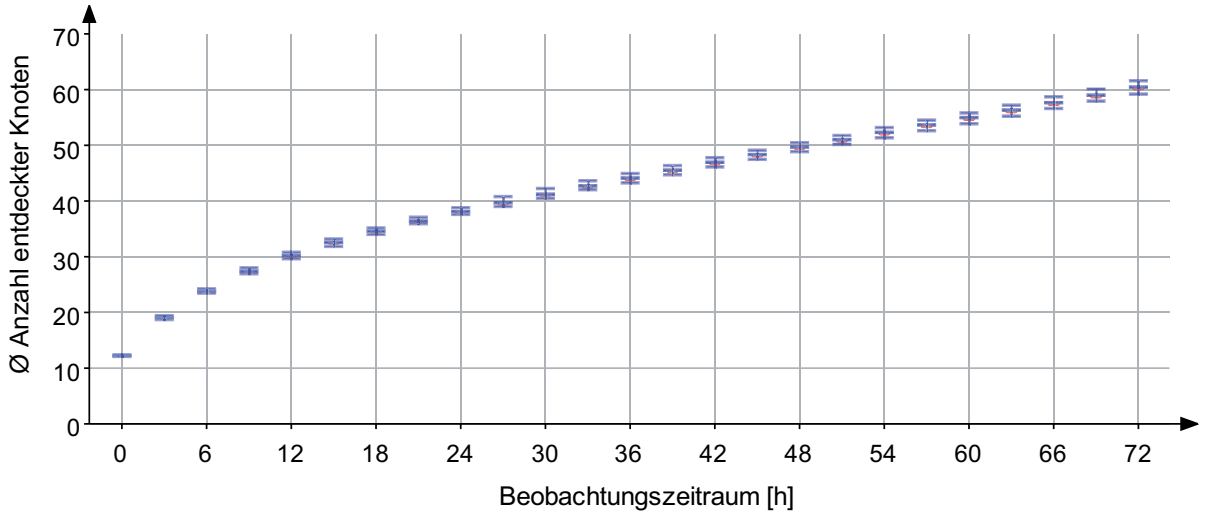


Abbildung 7.13: Anzahl der entdeckten SOLID-Knoten bei Beobachtung eines einzelnen Knotens über die Zeit (durchschnittlicher Fall)

Abbildung 7.13 zeigt die Ergebnisse des Experiments: Im Schnitt sind etwa 12,1 Sicherheitsbeziehungen zu jedem Zeitpunkt aufgebaut. Betrachtet man das Intervall der ersten zwölf Stunden, steigt die Kurve vergleichsweise schnell auf etwa 30 identifizierte Knoten an. Sie geht dann allerdings in einen linearen Verlauf über, sodass nach 72 Stunden im Schnitt lediglich 60,3 der 250 Knoten identifiziert werden.

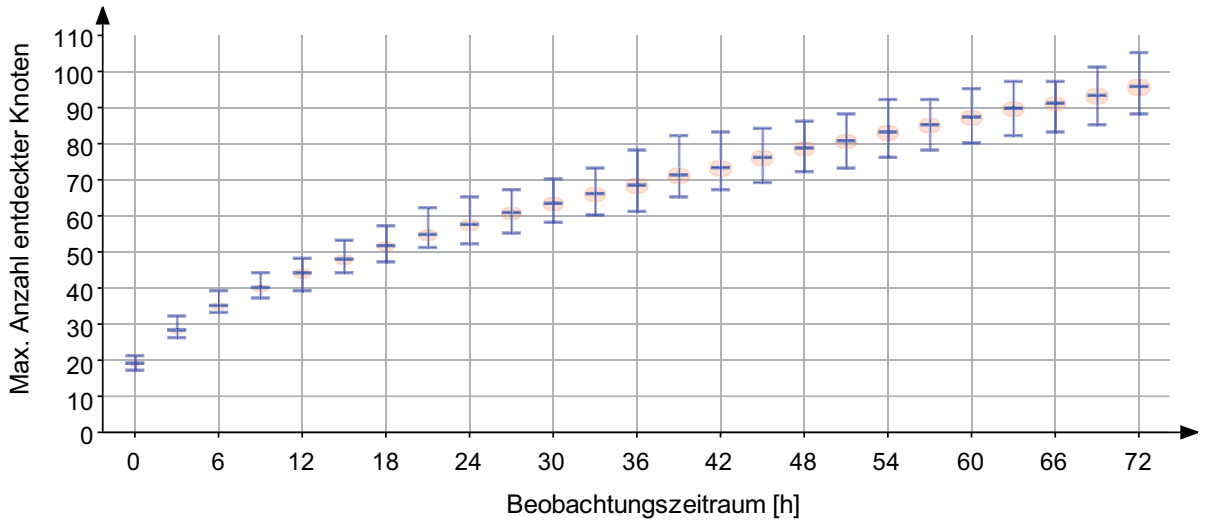
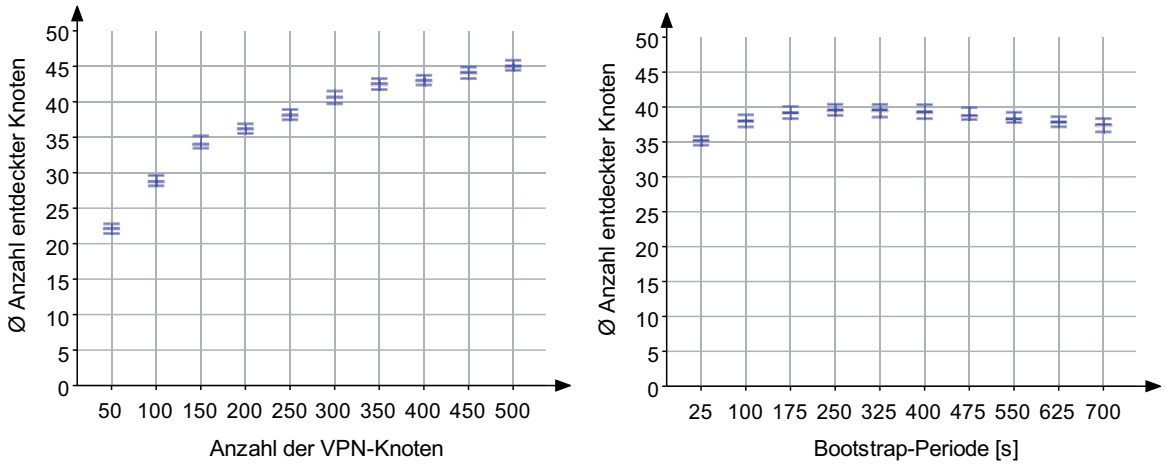


Abbildung 7.14: Anzahl der entdeckten SOLID-Knoten bei Beobachtung eines einzelnen Knotens über die Zeit (ungünstigstet Fall)

Da in Abschnitt 7.4.1 gezeigt wurde, dass der Knotengrad in SOLID-VPN nicht völlig homogen verteilt ist, lässt sich in Bezug auf den ungünstigsten Fall vermuten, dass die Kurve für das Maximum der identifizierten Knoten einen anderen Verlauf besitzt.



(a) (b)

Abbildung 7.15: Anzahl der entdeckten SOLID-Knoten in Abhängigkeit von der VPN-Größe und Bootstrap-Periode (Beobachtungszeitraum 24h)

Die Resultate sind in Abbildung 7.14 dargestellt. Es zeigt sich jedoch, dass im Prinzip lediglich eine Streckung der Funktion stattfindet. Der maximale Knotengrad beträgt im Schnitt zu jedem Zeitpunkt 18,9 und über einen Zeitraum von 72h sind zwischen 88 und 105 Knoten zu identifizieren.

Die ermittelten Werte hängen klar von Einstellungen, Transportnetzbedingungen und Größe des VPN ab, daher wurden zwei weitere Experimente im direkten Szenario durchgeführt, um zum einen der Einfluss der Größe des VPN und zum anderen den Einfluss des maximalen Bootstrap-Intervalls zu bestimmen.

Ersteres führt zu den Resultaten in Abbildung 7.15(a) und zeigt, dass die logarithmische Anzahl der Querverbindungen auch bei einer Betrachtung über einen Zeitraum von 24 Stunden eine signifikante Rolle spielt. Die Anzahl, der im Schnitt aufgedeckten Knoten, folgt in etwa einer logarithmischen Funktion, die bei einer VPN-Größe von 50 Knoten 22 beträgt und bei der zehnfachen Größe auf 45 aufgedeckte Knoten steigt. Der prozentuale Anteil sinkt also mit steigender VPN-Größe.

Die Werte in Bezug auf das maximale Bootstrapping-Intervall in Abbildung 7.15(b) entsprechen nicht völlig den Erwartungen. Während längere Zeiten zum Aufbau von weniger Sicherheitsbeziehungen führen sollten, ist dieser Effekt lediglich ab einer maximalen Periode von 325 Sekunden zu beobachten. Sehr kurze Perioden führen hingegen ebenfalls zu einer geringeren Anzahl an aufgedeckten Knoten. Zurückzuführen ist dieser scheinbare Widerspruch auf den Querverbindungsmechanismus, welcher bei kurzen Bootstrap-Intervallen die offenbar vom Prozess benötigten Sicherheitsbeziehungen verwendet und demzufolge weniger eigene Beziehungen aufbaut.

Insgesamt sind SOLID-Topologien damit durch ihre Struktur vor Beobachtern relativ gut geschützt, da durch passives Beobachten über einen kurzen Zeitraum lediglich eine logarithmische Anzahl von Knoten identifiziert werden kann. Länger dauernde

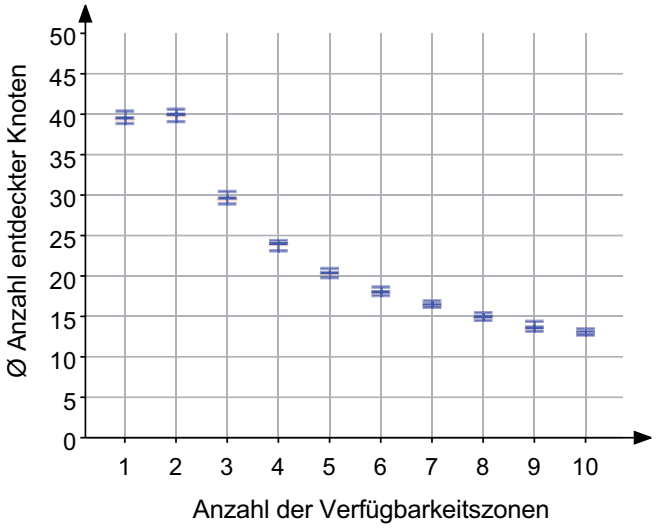


Abbildung 7.16: Anzahl der entdeckten SOLID-Knoten in Abhängigkeit von der Anzahl an Verfügbarkeitszonen (Beobachtungszeitraum 24h)

Beobachtungen erkennen zumindest in den ersten Tagen lediglich eine lineare Anzahl zusätzlicher Teilnehmer.

Soll der Schutz vor Beobachtung weiter verbessert werden, oder müssen auch bei vielen reaktiv aufgebauten Sicherheitsbeziehungen weiterhin Beschränkungen zugesichert werden, ist es möglich die in Kapitel 5 vorgestellten Verfügbarkeitszonen einzusetzen. Allein durch deren Existenz ist zu erwarten, dass die Konnektivität in Bezug auf den direkten VPN-Graphen abnimmt und so die Beobachtbarkeit sinkt. Wie in Abbildung 7.16 gezeigt, lässt sich dieses Verhalten auch experimentell bestätigen, wobei 250 Knoten im direkten Szenario gleichmäßig auf die linear in einer Dimension angeordneten Zonen verteilt wurden. Bei zwei Zonen ist erwartungsgemäß noch keine Verbesserung zu messen, da die Knoten weiterhin alle miteinander kommunizieren können. Aber bereits bei drei Zonen sinkt die Beobachtbarkeit schnell ab, bis bei zehn Zonen lediglich durchschnittlich 12,9 Knoten innerhalb von 24 Stunden entdeckt werden können.

Auswirkungen von Verfügbarkeitszonen auf die Ende–zu–Ende–Latenz

Gleichzeitig erhöht sich mit der Anzahl an Verfügbarkeitszonen jedoch die Länge der Pfade. Die Untersuchung des Verhältnisses zwischen der Verzögerung zwischen zwei Knoten bei Overlay-Beziehung und direkter Konnektivität führt zu den Resultaten in Abbildung 7.17. Nach 100.000 simulierten Sekunden im direkten Szenario wurden die aufgebauten indirekten Pfade ausgegeben und ins Verhältnis zur direkten Verzögerung gesetzt. Ferner wurden mit globalem Wissen Knoten gesucht, die möglicherweise einen besseren Pfad ergeben hätten, und die entstehende Verzögerung als zu erreichender Idealwert ermittelt.

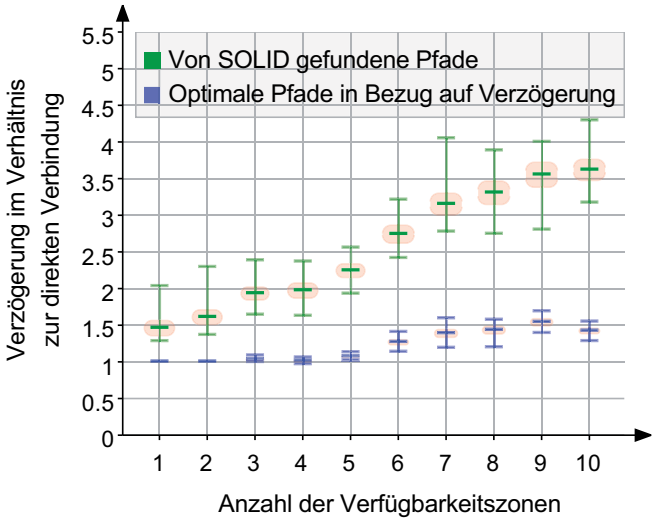


Abbildung 7.17: Verzögerungszeiten idealer und real entstehender Pfade indirekter Sicherheitsbeziehungen in Bezug auf die Anzahl an Verfügbarkeitszonen

Während im Idealfall die durchschnittlichen Verzögerungszeiten bei zehn Verfügbarkeitszonen lediglich auf das 1,4-fache steigen, wachsen im realen System bei der gleichen Zonenanzahl die Verzögerungen auf das 3,6-fache. Dies ist auf mehrere Effekte zurückzuführen:

- Die Ergebnisse der gemessenen Pfade sind teilweise nicht vollständig optimiert, da durch die Fluktuationen aufgrund des periodischen Bootstrappings und der beschriebenen Interaktion mit dem Querverbindungsmechanismus neue Pfade aufgebaut werden. Ferner kann auch die Dynamik im Vivaldi-Koordinatensystem dazu führen, dass Nachbarschaftsbeziehungen auf lange Sicht auf- und abgebaut werden. So sind beispielsweise die ersten zwei Werte der erfassten Topologien zu erklären, da aufgrund des direkten Szenarios in beiden Fällen zu erwarten gewesen wäre, dass gar keine indirekten Sicherheitsbeziehungen existieren. Beide Werte sind allerdings auf die wenigen neu aufgebauten Sicherheitsbeziehungen zurückzuführen.
- Im optimalen Fall können Knoten zur Weiterleitung verwendet werden, welche die Dreiecksungleichung in Bezug auf die Transportnetzverzögerung verletzen (siehe auch Abschnitt A.1). Auf diese Weise können durch Weiterleitungen tatsächlich leicht schnellere Wege gefunden werden, die beispielsweise auf der Umgehung von Policies des Border Gateway Protocol (BGP) beruhen. Die Auswirkungen dieses Effektes sind bei vier Verfügbarkeitszonen zu erkennen; das Minimum der gemessenen Werte liegt hierbei sogar unter der Verzögerung einer direkten Verbindung. Wie in Teilabschnitt 4.5.1 argumentiert, kann SOLID aufgrund des vergleichsweise niedrigen Gewinns bisher keine Optimierungen vornehmen, welche die verletzten Dreiecksungleichungen ausnutzen.
- Die verletzten Dreiecksungleichungen können darüber hinaus nicht mit gängigen Netzwerkkoordinatensystemen, wie Vivaldi, erfasst werden. Dies ist ein Grund dafür, dass keine optimale Einbettung der realen Verzögerungszeiten in den

Koordinatenraum erfolgen kann, und dass die Abbildung demzufolge nicht ideal ist. Vivaldi bildet allerdings die Grundlage für die Auswahl der Weiterleitungspunkte, sodass durch Fehleinschätzungen akkumulierte Fehler auftreten.

- Wie in Teilabschnitt 5.5.2 beschrieben ist, findet die Optimierung aufgrund des lokalen Wissens nicht in allen Fällen das globale Minimum. Gerade bei sehr langen Pfaden ist mit diesem Effekt häufiger zu rechnen.

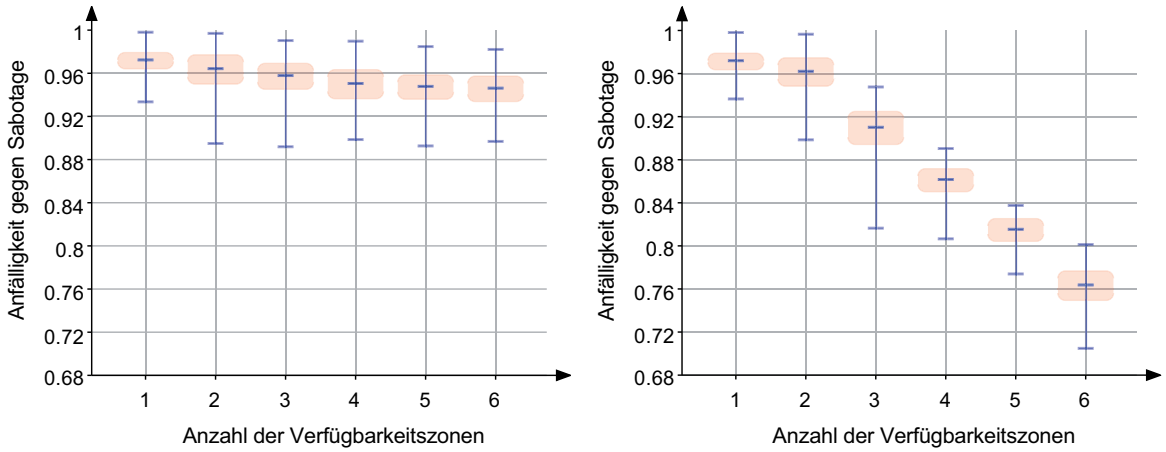
Die Anzahl der Verfügbarkeitszonen wirkt sich somit zwar auf die Ende-zu-Ende-Verzögerung aus, wobei ein Szenario mit bis zu neun Zonenübergängen in der Praxis wahrscheinlich als relativ extrem angesehen wird. Tatsächlich zu erreichende Verzögerungszeiten sind darüber hinaus sehr stark abhängig von den Gegebenheiten des VPN. Stellt die Ende-zu-Ende-Verzögerung dennoch ein praxisrelevantes Problem dar, könnten weitergehende Optimierungsmaßnahmen entwickelt werden. Beispielsweise kann durch eine monotone Klassenverteilung sichergestellt werden, dass nur wichtige Knoten in den oberen Schichten des VPN existieren. In diesem Fall könnten die Vivaldi-Koordinaten der Knoten unterer Schichten in die höheren Schichten weitergeleitet werden, sodass mit dem globaleren Wissen von dort Optimierungen initiiert werden. Darüber hinaus ist es möglich, Verfügbarkeitszonen auch netztopologisch korrekt zu vergeben, zum Beispiel indem Knoten vermittelnder Zonen in der Nähe von Internet-Knotenpunkten betrieben werden.

Anfälligkeit gegen Sabotage

Das Ziel der Einführung der Verfügbarkeitszonen, und den damit verbundenen erhöhten Ende-zu-Ende-Latenzen, war allerdings nicht nur die Senkung der Anzahl der durch einen Knoten beobachteten Nachbarn. Durch die Akkumulation dieser Wissensbeschränkung ist vielmehr zu erwarten, dass die in Abschnitt 5.2 vorgestellte Anfälligkeit gegen Sabotage bei steigender Anzahl der Verfügbarkeitszonen abfällt. Zur näheren Untersuchung des Effektes wurden weitere Experimente durchgeführt; aufgrund der hohen Berechnungskomplexität der Anfälligkeit gegen Sabotage aber lediglich mit 50 Knoten. Die Knoten wurden jeweils mit zufälligen Beobachtungswahrscheinlichkeiten aus dem Intervall $]0, 1]$ initialisiert und auf bis zu sechs Verfügbarkeitszonen aufgeteilt. Um die Werte gut vergleichen zu können, erfolgte die Anordnung der Zonen erneut linear in einer Dimension.

In einer ersten Messreihe wurde von einem passiven externen Beobachter ausgegangen, der über einen Zeitraum von 24 Stunden eine Reihe von Knoten des SOLID-VPN beobachten kann. Anschließend erfolgt ein Angriff auf alle identifizierten Knoten, sodass in Abhängigkeit von der Beobachtungswahrscheinlichkeit die Ende-zu-Ende-Konnektivität innerhalb des VPN sinkt. Die Anfälligkeit gegen Sabotage bestimmt sich nun analog zu Abschnitt 5.2 als Fläche unter der Kurve aus Eintrittshäufigkeit und dem Anteil ausgefallener Ende-zu-Ende-Verbindungen.

Abbildung 7.18(a) zeigt, dass auch ohne Wissen der tatsächlichen Beobachtungswahrscheinlichkeiten eine leichte Verbesserung der Anfälligkeit zu erreichen ist, da hier



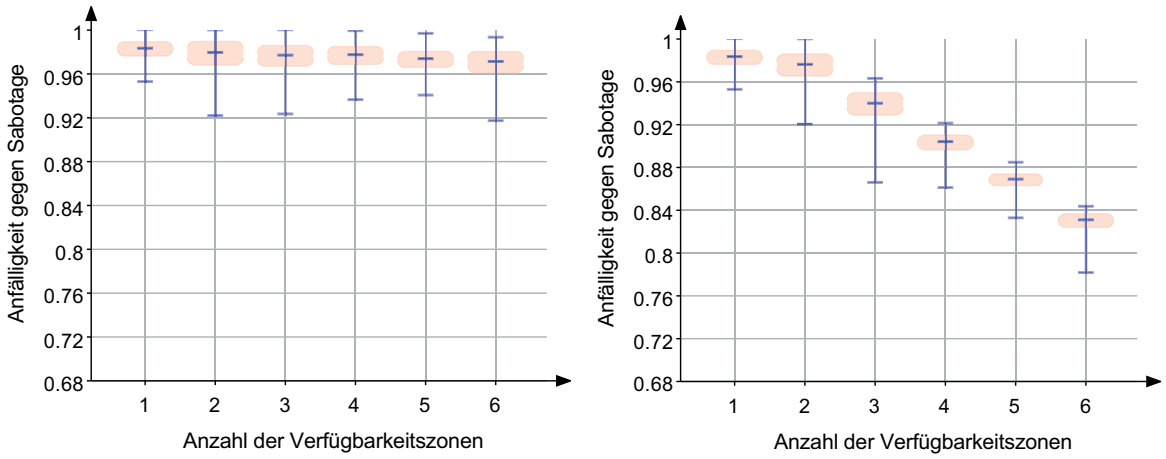
(a) (b)

Abbildung 7.18: Anfälligkeit gegen Sabotage in Abhängigkeit von der Anzahl an Verfügbarkeitszonen (Beobachtungszeitraum 24h, zufällige Vergabe links, monotone Vergabe rechts)

einfach eine zufällige Zuordnung von Knoten auf Verfügbarkeitszonen erfolgt ist. Die Anfälligkeit sinkt von im Schnitt 0,97 bei einer Zone auf 0,95 bei sechs Zonen. Die eingezeichneten Konfidenzintervalle haben in diesem Fall allerdings lediglich einen informativen Charakter, da die berechneten Werte aufgrund der Nähe zur theoretischen Obergrenze von nicht völlig normalverteilt sind. Eine Besonderheit ist in den Experimenten mit zwei Verfügbarkeitszonen zu erkennen: Die Anfälligkeit sinkt hier bereits leicht ab, obwohl beide Verfügbarkeitszonen direkt miteinander kommunizieren können. Zurückzuführen ist dies auf den Querverbindungsalgorithmus, der Knoten gleicher Zone leicht bevorzugt, und somit durch die proaktiv aufgebauten Sicherheitsbeziehungen eine gewisse Separierung hervorruft. Solange ein gewisser Mindestzusammenhang gewährleistet werden kann, sind demzufolge durch den Einsatz von Verfügbarkeitszonen zumindest keine negativen Auswirkungen zu erwarten.

Allerdings hilft eine korrekte Einschätzung der Beobachtungswahrscheinlichkeiten die Resistenz zu erhöhen, wie Abbildung 7.18(b) zeigt. Hier wurden die Knoten je nach ihrer Beobachtungswahrscheinlichkeit in monotoner Reihenfolge in Zonen eingeteilt. Einzig die beiden höchsten Verfügbarkeitszonen wurden vertauscht, sodass die resistentesten Knoten die höchste Zone vom Rest des VPN abschirmen. Durch die bessere Zuordnung sinkt die Anfälligkeit gegen Sabotage von ebenfalls von 0,97 bei einer Zone auf 0,76 im besten Fall.

Das bisher betrachtete Angreifermodell ist jedoch vergleichsweise schwach. Interne Angreifer oder Angreifer, die gleichzeitig DoS-Angriffe durchführen und Knoten beobachten können, sind in der Lage, die Adressen aller Knoten kompatibler Verfügbarkeitszonen zu identifizieren. Letztere müssen dazu lediglich die Nachbarn der beobachteten Knoten nacheinander sabotieren und können anschließend erkennen, zu welchen anderen Knoten sich die observierten Knoten verbinden. Das gleiche Szenario



(a) (b)

Abbildung 7.19: Anfälligkeit gegen Sabotage in Abhängigkeit von der Anzahl an Verfügbarkeitszonen (Angriff bei gleichzeitiger Beobachtung, zufällige Vergabe links, monotone Vergabe rechts)

tritt ein, wenn ein beobachteter Knoten beispielsweise mit allen anderen Knoten des VPN Nutzdaten austauscht.

Dieses sehr starke Angreifermodell führt dazu, dass bei einer zufälligen Verteilung von Knoten auf die Verfügbarkeitszonen (siehe Abbildung 7.19(a)) die Anfälligkeit kaum abnimmt. Sie sinkt lediglich auf 0,97. Wie in Abbildung 7.19(b) dargestellt ist, erreicht die Anfälligkeit allerdings bei einer monotonen Verteilung der Knoten — trotz des starken Angreifermodells — einen Wert von 0,83 bei sechs Zonen.

Insgesamt kann durch die Einführung von Verfügbarkeitszonen die Resistenz eines VPN gegenüber DoS-Angriffen zum Teil stark gesteigert werden. Ferner können diese Werte zukünftig durch eine detaillierte Modellierung realer Szenarien und eine darauf abgestimmte Zonenverteilung zu potentiell verbesserten Ergebnissen führen. Gegenüber dem passiven Beobachter bietet SOLID bereits jetzt, aufgrund der wenigen proaktiv aufgebauten Sicherheitsbeziehungen, eine gewisse Resistenz.

7.5 Fazit

In diesem Kapitel wurden die in Abschnitt 3.1 formulierten Anforderungen mit den von SOLID erreichten Eigenschaften verglichen. Dabei wurden alle Anforderungen zunächst qualitativ diskutiert, und anschließend das Verhalten des Systems in Prototyp und innerhalb des OMNeT++-Netzwerksimulators unter verschiedenen Umweltbedingungen und Fragestellungen erfasst. Insbesondere wurden die folgenden nichtfunktionalen Eigenschaften untersucht:

- **Skalierbarkeit:** Alle Experimente, bei denen die VPN-Größe betrachtet wurde, zeigen, dass das entworfene System hochgradig skalierbar in Bezug auf die Anzahl der Teilnehmer ist. Sowohl das Einfügen und Entfernen von Knoten als auch

das Zusammenführen von VPN-Partitionen erfolgt in maximal logarithmischer Zeit (siehe Teilabschnitt 7.4.2 und 7.4.3). Entsprechen die Transportnetze nicht den Annahmen aus Abschnitt 4.5.3 und besitzen Zyklen, die nicht lokal optimiert werden, wächst die maximale Pfadlänge ebenfalls maximal logarithmisch (Abschnitt 7.4.1). Der gemessene Knotengrad wächst in direkten Szenarien ebenfalls logarithmisch (Abschnitt 7.4.1 und 7.4.5). In HOT-Szenarien wird der maximale Knotengrad durch den Grad der Knoten im Transportnetz bestimmt, sodass es Aufgabe eines Administrators ist, in diesen Szenarien eine der Exponiertheit entsprechende leistungsfähige Hardware einzusetzen.

- **Robustheit:** In Bezug auf die Robustheit wurden Netzpartitionierungen, und insbesondere solche, die nicht durch den Ausfall von SOLID-Gateways entstehen, als kritischste Situation identifiziert. Dennoch ist es mit den vorgeschlagenen Algorithmen möglich, auch aus solchen Konstellationen sicher in einen konsistenten Zustand zu gelangen, wie in Teilabschnitt 7.4.3 gezeigt wurde. Der in den Anforderungen erwähnte gleichzeitige Ausfall und Start von Knoten wird im Rahmen dieser Ausarbeitung nicht näher vorgestellt, da die Ergebnisse der dazu durchgeführten Simulationen keine besonderen Erkenntnisse einschließen. In beiden Situationen treten im ungünstigsten Fall unabhängige Partitionen auf, die in ähnlicher Zeit wie die untersuchten Transportnetzpartitionen behoben werden können. Genauere Resultate zum Verhalten bei gleichzeitigem Start beziehungsweise Ausfall finden sich unter anderem in [Sch10].

- **Agilität:** Vor allem durch die wenigen proaktiv aufzubauenden Sicherheitsbeziehungen erlaubt SOLID das schnelle Einfügen von VPN-Knoten. Wie in Teilabschnitt 7.4.2 gezeigt wurde, ist die Verzögerung von wenigen Sekunden primär auf die IKE-Signaturen zurückzuführen. Der damit verbundene Rechenaufwand ist jedoch nicht Teil des Autokonfigurationsmechanismus, sodass in Zukunft lediglich durch das Einführen von MOBIKE für mobile Knoten eine Verbesserung zu erreichen ist. Die Dauer bis zur Ringreparatur nach dem Entfernen von Knoten wird nahezu ausschließlich vom Heartbeat-Intervall bestimmt. Optimierungen sind hier nur durch häufigeres Senden der Pakete oder die zusätzliche Detektion des Ausbleibens von Nutzdatenverkehr möglich.

- **Effizienz:** Der in Teilabschnitt 7.3.2 gemessene Verwaltungsaufwand bewegt sich in einem akzeptablen Bereich, und kann in Zukunft durch Optimierung der von SOLID ausgetauschten Nachrichten weiter verringert werden. Auch hier ist der primäre Grund für die bisher erreichten Ergebnisse die niedrige Anzahl an proaktiv aufgebauten Sicherheitsbeziehungen. Durch den eingeführten stichprobenbasierten Querverbindungsmechanismus ist es dabei auch bei der Einbettung in verschachtelte Transportnetze möglich, sehr effektiv Sicherheitsbeziehungen zu initiieren (siehe Abschnitt 7.4.1). Die aufgebauten Sicherheitsbeziehungen können darüber hinaus in sehr vielen Situationen optimal abgekürzt werden, und in Transportnetzen, in denen die Bedingungen für den Umlenkungsalgorithmus nicht ideal sind, ist ein signifikantes Ansteigen der mittleren Pfadlänge ebenfalls nicht zu erwarten (Abschnitt 7.4.1). Wie in Teilabschnitt 7.4.4 gemessen wurde, können darüber hinaus auch bei partiellen Ausfällen vergleichsweise gute

Weiterleitungspunkte gefunden werden. Lediglich bei der Verwendung vieler Verfügbarkeitszonen ist in Teilabschnitt 7.4.5 gezeigt worden, dass im Verhältnis zum Optimum Raum für Verbesserungen vorhanden ist.

Ferner konnte in Teilabschnitt 7.4.5 mithilfe von Simulationen die Resistenz von SOLID-Netzen in Bezug auf die Knotenbeobachtung belegt werden. In diesem Zusammenhang konnte auch die Wirksamkeit von Verfügbarkeitszonen am Beispiel von Topologien gezeigt werden, welche das entwickelte System erzeugt.

Einzig die quantitative Bewertung des Multicast-Teilsystems erfolgte an dieser Stelle nicht, da dessen Feinabstimmung nicht im primären Fokus der Arbeit gesehen wird. Zentrale quantitative Eigenschaften können jedoch [Hei09, HRS10] entnommen werden.

Insgesamt erfüllt SOLID die aufgestellten qualitativen und quantitativen Anforderungen — insbesondere im Vergleich zum bisherigen Stand von Wissenschaft und Technik — sehr gut. Es ist für den Nutzer einfach zu konfigurieren, verhält sich transparent, erlaubt unter anderem die Nutzung privater IP-Adressbereiche, die Unterstützung beliebiger Schachtelungstiefen, skaliert über die Anzahl der Teilnehmer, und ist außerdem effizient und robust. Die Sicherheitseigenschaften werden durch den Einsatz des komplexen Konfigurationsmechanismus nicht geschwächt. Im Gegenteil: die Flexibilität von SOLID-VPN erlaubt es, dynamisch auf Sabotageangriffe zu reagieren und durch die Einführung von Verfügbarkeitszonen kann die Beobachtbarkeit von VPN zum Teil erheblich gesenkt werden.

The outcome of any serious research can only be to make two questions grow where only one grew before.

(Thorstein Veblen)

Kapitel 8

Resümee

Das schließende Kapitel fasst zunächst den Ausgangspunkt und die zentralen Ideen dieser Arbeit zusammen und versucht im zweiten Teil einen Ausblick auf zukünftige wissenschaftliche und technische Entwicklungen zu geben.

8.1 Zusammenfassung

Über die letzten 20 Jahre hat sich das Internet zu einem derart kostengünstigen Medium entwickelt, dass es zum einen für einen globalen Informationsfluss unabdingbar geworden ist. Zum anderen hat sich die Geschwindigkeit, des darüber realisierten Austausches, zu einem Hauptkriterium für den Erfolg von Firmen, Behörden und Organisationen herauskristallisiert. Um diese Vorzüge auch für den internen Datenaustausch zu nutzen, müssen jedoch Vertraulichkeit, Authentizität und Sabotageresistenz gewährleistet werden. Da das Internet Protokoll diese Eigenschaften nicht inhärent bietet, werden in der Regel VPN über das Internet hinweg betrieben.

Diese normalerweise manuell konfigurierten Overlay-Netze dienen sowohl der statischen Vernetzung von Firmenstandorten als auch der flexiblen Anbindung einzelner Teilnehmer, die ad-hoc in das Netz integriert werden. Für letzteres Szenario werden häufig Hub-and-Spoke-Architekturen eingesetzt, bei der ein oder mehrere zentrale VPN-Konzentratoren die Verbindungen von Außenstandorten oder mobilen Teilnehmern entgegennehmen und Daten zwischen diesen weiterleiten. Bei der Vernetzung von Firmenstandorten werden auch gleichberechtigte Sicherheits-Gateways eingesetzt, mit denen die Standorte vollvermascht werden. Im Gegensatz zu den Hub-and-Spoke-Topologien sind diese jedoch aufwändig zu konfigurieren und zu warten. Dadurch leidet zum einen die Skalierbarkeit und zum andern können mobile Teilnehmer nicht berücksichtigt werden. Im Gegenzug sind vollvermaschte VPN effizienter bezüglich der Pfadlängen und besitzen keinen dedizierten Ausfallpunkt oder Flaschenhals. Die Sabotageresistenz, Robustheit und Skalierbarkeit über die Anzahl der Teilnehmer bereitet somit in beiden Szenarien Probleme: in Hub-and-Spoke-Netzen aufgrund des zentralen Konzentrators und in vollvermaschten Umgebungen durch die hohe Anzahl aufzubauender Sicherheitsbeziehungen.

Ausgehend von dieser unbefriedigenden Situation, wurden in den letzten Jahren eine Vielzahl von VPN-Autokonfigurationsansätzen entwickelt und teilweise auch

implementiert. Wie in Abschnitt 3.2 und [RS11] dargestellt ist, adressieren sie die Verfügbarkeit und Robustheit der aufgebauten VPN-Topologien allerdings nicht in ausreichendem Maße. Die wenigen skalierbaren Systeme erfüllen darüber hinaus selbst einfache funktionale Anforderungen, wie den Einsatz privater IP-Adressen innerhalb des VPN, nicht. Zwei Ansätzen konnten im Rahmen der Dissertation sogar eklatante Sicherheitsmängel nachgewiesen werden [RS09, Ros10].

Aus diesem Grund ist im Rahmen einer Diplomarbeit [Ros07] ein erstes Konzept für einen neuartigen VPN-Autokonfigurationsansatz auf Basis von IPsec entstanden. Dieses Konzept ist im Rahmen der Dissertation stark ausgebaut worden und hat zur Entwicklung von SOLID [RSM09, RSS10] geführt — einem in Kapitel 4 vorgestellten Autokonfigurationsansatz, der es erlaubt, schnell und einfach dynamische IPsec-Infrastrukturen einzurichten.

Durch den Einsatz von Peer-to-Peer-Prinzipien und dem damit verbundenen Verzicht auf exponierte Systeme stellt der Ansatz eine ideale Grundlage für ein skalierbares und hochgradig verfügbares System dar, da zumindest aus Sicht des VPN Knoten nahezu beliebig entfernt und hinzugefügt werden können. Ein grundlegender Punkt, der für eine Verwendung von SOLID spricht, ist die Unterstützung indirekt angebundener VPN-Knoten, also solchen Knoten, die nur über andere erreichbar sind. Dabei wird die Vertraulichkeit, die Integrität und die Authentizität der übertragenen Daten immer Ende-zu-Ende garantiert, und in Bezug auf Verfügbarkeit beispielsweise eine Umleitung von Verkehr im Falle partieller Kommunikationsstörungen über dritte Standorte erlaubt.

Durch den Autokonfigurationsmechanismus ist es ferner möglich, auf Sabotageangriffe durch Veränderung der Infrastruktur schnell zu reagieren, indem beispielsweise die externen IP-Adressen betroffener Systeme gewechselt werden. Beim Entwurf des Systems konnte unter anderem gezeigt werden, dass auch ohne den Broadcast von Routing-Informationen in gebräuchlichen zyklenfreien VPN-Topologien optimale Pfade gefunden werden, und wie bei der Einbettung strukturierter Overlay-Netze in beliebige Transportnetztopologien Inkonsistenzen und Partitionierungen vermieden werden können.

Ein zusammenhängendes Overlay vorausgesetzt, können darüber hinaus direkte Sicherheitsbeziehungen nahezu beliebig aufgebaut oder administrativ verboten werden. VPN-Knoten ohne direkte Kommunikationsmöglichkeit können in diesem Fall Pakete über automatisch konfigurierte indirekte Sicherheitsbeziehungen austauschen. Durch den Verzicht direkter Sicherheitsbeziehungen kann SOLID die nach außen sichtbaren IP-Adressen bestimmter VPN-Gateways verbergen und sie so proaktiv vor DoS-Angriffen schützen. Die analytische Konstruktion von optimal sabotageresistenten Topologien erwies sich im allgemeinen Fall als nicht in angemessener Zeit berechenbar, daher ist im Rahmen der Arbeit ein System [BRS09] entwickelt worden, welches unter Zuhilfenahme verschiedener Heuristiken dennoch Topologien mit garantierten Sicherheitseigenschaften konstruieren kann. Dabei wird unter anderem jedem VPN-Knoten eine Verfügbarkeitszone zugeordnet und lediglich zwischen bestimmten Zonen eine direkte Kommunikation erlaubt. Durch die Anordnung der Verfügbarkeitszo-

nen und die Einteilung der Knoten können Administratoren zwischen Effizienz und Sabotageresistenz abwägen.

Das Gesamtkonzept wurde mithilfe einer umfangreichen Implementierung evaluiert, welche unter anderem in [RSSM09, Ste08b] und Kapitel 6 näher vorgestellt wird. Dazu sind plattformunabhänige Teile gekapselt, sodass eine parallele Untersuchung in OMNeT++/INET-Simulationen und Realumgebung auf Basis von strongSwan/Linux erfolgen kann. Beispielsweise durch die Einführung von IP-Adressattestationen zur Sicherung gegen interne Angreifer und die flexible Nutzung von IPIP-Tunneln in BEET-Beziehungen geht auch die Implementierung über den Stand der Technik hinaus. Der Prototyp ist dabei bereits so entworfen, dass eine spätere Integration in Produktivsysteme erfolgen kann.

Insgesamt hat sich somit im Rahmen der Arbeit sukzessive aus einer ersten Idee ein komplexes System zur Konfiguration von VPN auf IPsec-Basis entwickelt. Für die Grundfunktionalität existiert eine zusammen mit der secunet Security Networks AG eingereichte europaweite Patentanmeldung [RSM09]. Im Rahmen der in diesem Kontext aufgebauten Partnerschaft wird zurzeit in einem Industrieprojekt ein gemeinsamer Prototyp auf Basis von SOLID und SINA entwickelt, und auch in Zukunft wird es gemeinsame Entwicklung geben. Ein auf der Arbeit aufbauendes BMBF-Projekt zur Nutzung von SOLID für den Einsatz in hochgradig mobilen Umgebungen ist im Oktober 2010 gestartet und ist mit einem Gesamtaufwand von insgesamt acht Personenjahren an der TU Ilmenau und in der secunet Security Networks AG veranschlagt.

8.2 Ausblick

Neben den Mobilitätseigenschaften werden derzeit auch weitere Aspekte aus wissenschaftlicher und technischer Sicht untersucht. Dazu zählen unter anderem eine weitere analytische Stabilitätsbewertung gegen Sabotageangriffe und eine Dezentralisierung der Netzwerkdienste höherer Schichten.

Auch wenn SOLID durch den Verzicht auf Routing-Annoncierungen eine vergleichsweise hohe Resistenz gegen Sinkhole-Angriffe aufweist, können kompromittierte Knoten dennoch Verkehr auf sich ziehen: Vor allem durch Manipulationen des Vivaldi-Koordinatensystems und durch Unterdrücken der Routen-Optimierungen. Wie bereits in [GRS+10a] angedacht, könnte dies durch Verbesserungen am Koordinatensystem und unter Ausnutzung VPN-spezifischer Gegebenheiten geschehen, wie etwa der Möglichkeit Attestationen auszustellen. Unterdrückte Optimierungen können möglicherweise durch stochastisches Routen über andere Pfade geschehen, wie dies bereits bei den Ringaktualisierungen geschieht. Wird SOLID, wie unter anderem in Abschnitt 7.4.1 angedacht, mit herkömmlichen Routing-Verfahren kombiniert, müssen zusätzliche Maßnahmen, wie Pfadattestierungen, ergriffen werden, um Sinkhole-Angriffe einzuschränken. Ferner ist es unter Umständen möglich, Angreifer durch Knotenstatistiken zu entdecken, wenn diese sehr viel mehr Verbindungen weiterleiten als zeitlich und

räumlich üblich. Es ist geplant die genaueren Eigenschaften dieser Gegenmaßnahmen durch eine Implementierung und weitere simulative Experimente zu untersuchen.

Eine Analyse des übertragenen Datenverkehrs wird in herkömmlichen VPN in der Regel durch die Verwendung von Traffic Shaping, das Generieren von Cover Traffic und ähnlich aufwendige Maßnahmen erschwert. Innerhalb von SOLID-Netzen können diese Ansätze in Zukunft durch dynamische Maßnahmen ergänzt werden, beispielsweise indem Pfade asymmetrisch weitergeleitet werden und Verkehrsflüsse mit einem geringen Volumen nicht optimiert werden. Ebenso wie Multipath-Routing-Strategien können diese Mechanismen es einem externen Angreifer erschweren, einzelne Verkehrsflüsse zu identifizieren.

Als Alternative zur statischen Einteilung in Verfügbarkeitszonen ist es denkbar, SOLID-VPN durch dynamische Topologieadaption stabile Strukturen einnehmen zu lassen. Wie in Abschnitt 5.6 erwähnt, könnten dann beispielsweise mobile VPN-Knoten, mit häufig wechselnden Standorten, lediglich zu Bereichen des VPN direkt in Kontakt treten, die nicht kritisch für die Kernfunktionen sind. Darüber hinaus können Gruppierungen auf Basis gleicher geographischer Lage oder Anbindungen über den gleichen ISP vorgenommen werden.

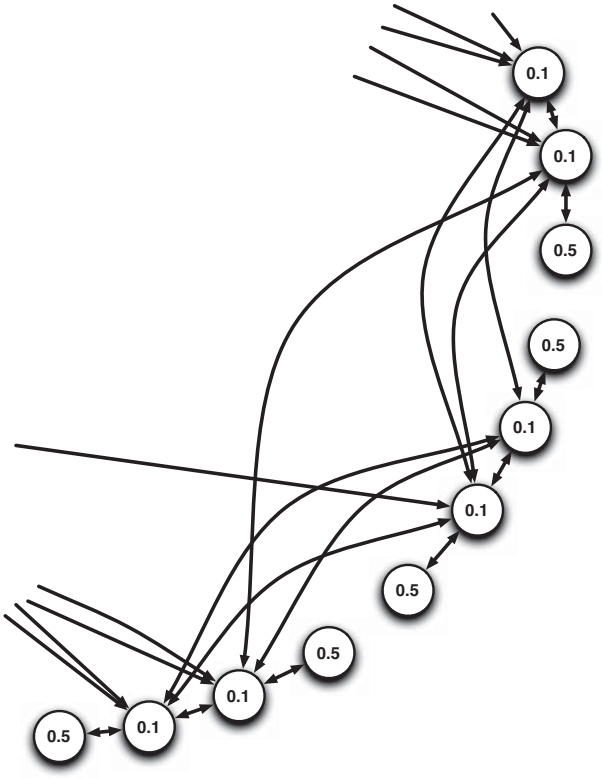


Abbildung 8.1: Beispiel eines vergleichsweise angriffsstabilen Topologiemusters

Interessant könnte sich auch die Untersuchung unterschiedlicher lokaler Strukturen auf die Resistenz des Gesamtnetzes erweisen. Startpunkte hierzu könnten eine Analyse auf Basis verschiedener Network Motifs [MSI+02] oder Protection Cycles [SG00] sein. Es ist dabei zu erwarten, dass die entstehenden Muster sich aufgrund der anderen Voraussetzungen stark von denen herkömmlicher, auf Robustheit ausgelegter Netze

unterscheiden. Abbildung 8.1 zeigt beispielsweise einen Ausschnitt, bei dem zwei beliebige VPN-Knoten mit Beobachtungswahrscheinlichkeit 0,5 ausfallen können, ohne dass es zu einer Trennung des Ringes kommt. Die möglicherweise zu identifizierenden Strukturmuster würden dann sowohl in der statischen Zonenzuweisung, als auch in dynamischen Systemen zum Einsatz kommen.

Für einen verteilten Konfigurationsansatz kann SOLID vergleichsweise schnell Knoten in ein VPN reintegrieren. Zur Anbindung von stark mobilen Nutzern sind jedoch zentrale VPN-Konzentratoren oder Home Agents auf Mobile-IP-Basis zurzeit noch besser geeignet, da lediglich ein Verbindungskontext aktualisiert werden muss und weniger Fluktuationen im VPN entstehen. Diesem Umstand soll unter anderem im Rahmen des BMBF-Projektes durch drei Maßnahmen begegnet werden:

- Auf technischer Seite kann durch die Unterstützung des IKEv2 Mobility and Multihoming Protocol (MOBIKE) [Ero06] der in Teilabschnitt 7.4.2 identifizierte Haupteinflussfaktor, die asymmetrische Kryptographie, reduziert werden.
- Gleichzeitig müssen schnelle Handover für indirekte Sicherheitsbeziehungen implementiert werden, sodass im verschachtelten Fall auch bei einem Wechsel des vorgelagerten VPN-Knotens Pfade schnell aktualisiert werden.
- Knoten, die über einen hohen Grad an Mobilität verfügen, sollen ferner über Statistiken erkannt werden, sodass zum einen häufiger Heartbeat-Nachrichten übertragen werden und sie zum anderen auch bei der Weiterleitung von Suchanfragen und der Wahl von Querverbindungen weniger stark berücksichtigt werden. Auf diese Weise ist es möglich, den Einfluss der mobilen Knoten auf das restliche VPN zu minimieren.

Für einen sinnvollen Einsatz in MANET-Umgebungen genügen diese Verbesserungen noch nicht: Hier müssen vermehrt Zyklen im Transportnetz erkannt und so beispielsweise koordiniert eine von mehreren Verbindungen ins Internet gewählt werden. Hierzu laufen erste Untersuchungen, die erkennen sollen, inwiefern beispielsweise durch passives Protokollieren der Pfade von Suchanfragen Zyklen zeitnah erfasst und bewertet werden können.

Für den Einsatz in ad-hoc Netzen, aber auch zur Realisierung sabotageresistenter Kommunikationsinfrastrukturen, ist die verteilte Konfiguration des VPN nicht ausreichend, solange die benötigten unterstützenden Dienste, wie Namensauflösung und Zeitsynchronisation innerhalb des VPN weiterhin zentral zur Verfügung gestellt werden. Der SOLID-Prototyp besitzt aus diesem Grund erste verteilte Realisierungen von DNS und NTP [RSM10a, GRS10b, Gol10, Sch10], wobei ein Fokus der Entwicklungen auf der Resistenz gegen DoS-Attacken und internen Angreifer lag. Für einen Produktivbetrieb sind jedoch weitere Tests und Implementierungen notwendig. Insbesondere müssen Möglichkeiten geschaffen werden, die VPN-Software sicher zu aktualisieren, Zertifikate zu verwalten und Zugriffssteuerungslisten zu verteilen. Letztere dienen der Realisierung interner Kommunikationsbarrieren, wie sie in manuell konfigurierten VPN häufig durch Hinzufügen von oder Verzicht auf IPsec-Policies realisiert werden. Darüber hinaus müssen Mechanismen für ein verteiltes Management geschaffen werden, wie beispielsweise Logging und Visualisierung.

Aus technischer Sicht müssen vor einem Produktiveinsatz vor allem Optimierungen durchgeführt werden, welche die Belastung des Netzwerks möglichst verringern. Ferner ist der Prototyp zurzeit an die Bedürfnisse einer forschungsnahen Evaluierung und einfachen Erweiterbarkeit angepasst; infolgedessen ist noch keine Härtung der Sicherheit des Prototypen erfolgt. Dieser müsste beispielsweise in mehrere Teilprogramme mit unterschiedlichen Rechten zergliedert werden, und gegebenenfalls einem systematischen Audit auf Implementierungslücken wie Buffer-Overflows unterzogen werden.

Neben diesen auf eine bessere Qualität abzielenden Maßnahmen können verschiedene funktionale Erweiterungen vorgenommen werden, um neue Nutzungsszenarien zu erschließen. Beispielsweise kann durch eine automatische Erkennung von weiteren SOLID-Gateways im roten Netz eine Load-Balancing-Funktionalität bereits auf VPN-Ebene realisiert werden, sodass pro VPN-Gateway weniger Sicherheitsbeziehungen aufgebaut werden müssen. Von solchen ausfallsicheren Netzen würde unter Umständen auch die Sabotageresistenz profitieren.

Der SOLID-Prototyp ist größtenteils auf IPv4 ausgelegt. Durch die Resultate einer aktuell laufenden Arbeit [Sta10] werden zumindest Teile der SOLID-Funktionalität auch innerhalb von IPv6-Netzen einzusetzen sein, und auch die inneren Netze können von dem Protokoll profitieren.

Anhang A

Weiterführende Messungen

Im Folgenden werden Umwelteinflüsse beschrieben die signifikante Entwurfsentscheidungen von SOLID begründen. Zunächst wird dazu auf die Verzögerungszeit eingegangen, die durch kryptographische Operationen bei der Aushandlung von Sicherheitsbeziehungen entstehen. Im zweiten Teilabschnitt werden Verletzungen der Dreiecksungleichung bei Verzögerungszeiten zwischen Internet-Endsystemen untersucht.

A.1 Verzögerung der IPsec–Schlüsselaushandlung

Der Einsatz von asymmetrischer Kryptographie ist durch die verwendeten großen Zahlen für seine Ressourcenintensität bekannt. Um die genauen Auswirkungen der einzelnen kryptographischen Verfahren auf IPsec-VPN zu evaluieren, wurden Messungen auf zwei Embedded Computern mit jeweils einem AMD-Geode-LX800-Prozessor durchgeführt. Softwareseitig wurde der IKEv2-Daemon aus dem strongSwan-Projekt [Ste08a] unter Linux eingesetzt; als Smartcards wurden zwei Aladdin eToken PRO verwendet.

Dabei ergaben sich nach jeweils 32 Messungen die Verzögerungen, wie sie in Abbildung A.1 dargestellt sind. Der Abstand der gemessenen Minima und Maxima, sowie das berechnete 99%-Konfidenzintervall (rot) sind dabei sehr gering, sodass von sehr exakten Werten ausgegangen werden muss.

Beim Einsatz kryptographischer Signaturen auf Basis der elliptischen Kurven `secp256r1` und `secp521r1` sind — verglichen mit dem erreichten Sicherheitsniveau — klar die geringsten Verzögerungen möglich. Allerdings dauert auch in diesen Fall eine Aushandlung im Schnitt mindestens 0,05 Sekunden. Sehr viel kritischer ist der Einsatz von RSA auf Smartcards zu bewerten, die Verzögerungen betragen hierbei 1,5 Sekunden bei 1024-Bit-Schlüsseln beziehungsweise 2,2 Sekunden bei 2048 Bit. Beim Aufbau vollvermaschter VPN ist somit für die Eingliederung von VPN-Knoten eine signifikante Zeit zu erwarten.

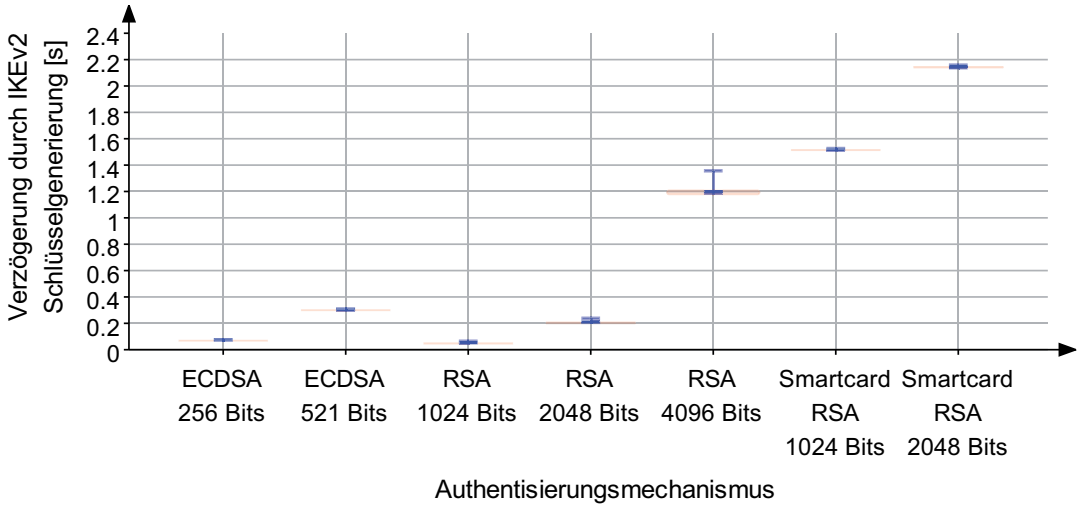


Abbildung A.1: Verzögerung der Schlüsselaushandlung beim Einsatz verschiedener Authentisierungsverfahren in IKEv2 mit min-max-Werten (blau) und 99,9%-Konfidenzintervallen (rot)

A.2 Häufigkeit der Verletzung der Dreiecksungleichung bei Verzögerungszeiten

Um zu untersuchen wie häufig es sich in einem Overlay in Bezug auf die Verzögerungszeit lohnt über einen dritten Knoten trotz direkter Verbindung weiterzuleiten, wurde die Häufigkeit der Verletzung der Dreiecksungleichung des Meridian-Datensatzes [WSS05] untersucht. Dabei ergab sich die kumulative Aufstellung aus Abbildung A.2, aus der hervor geht, dass nur in 7,8% der Fälle eine Optimierung möglich ist, und diese sogar in unter 0,37% der Fälle mehr als 10*ms* beträgt.

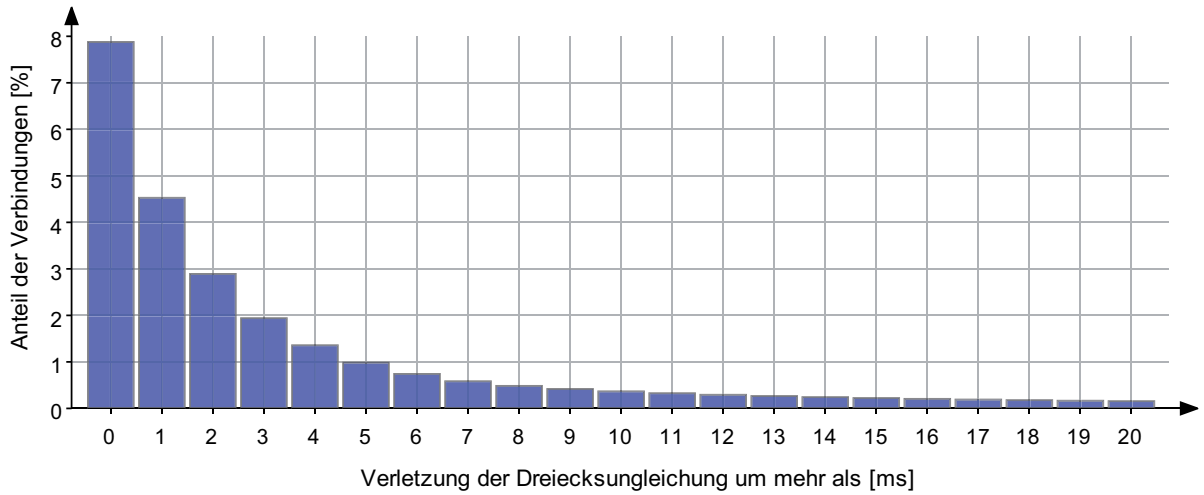


Abbildung A.2: Kumulative Häufigkeit von Verbindungen bei denen die Dreiecksungleichung nicht erfüllt wird

Literaturverzeichnis

[ABKM01] Andersen, David ; Balakrishnan, Hari ; Kaashoek, Frans ; Morris, Robert: Resilient Overlay Networks. In: *SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles*, 2001. – ISBN 1–58113–389–8, S. 131–145

[AD04] Aboba, Bernard ; Dixon, William: *IPsec-Network Address Translation (NAT) Compatibility Requirements*. IETF Request for Comments 3715 (Proposed standard). `http://www.ietf.org/rfc/rfc3715.txt`. Version: 2004

[And03] Andersen, David G.: Mayday: Distributed Filtering for Internet Services. In: *Proceedings of the 4th Symposium on Internet Technologies and Systems (USITS)*, 2003

[ANL01] Aura, Tuomas ; Nikander, Pekka ; Leiwo, Jussipekka: DOS-Resistant Authentication with Client Puzzles. In: *Security Protocols*. Springer Berlin / Heidelberg, 2001, S. 170–177

[AS07] Aspnes, James ; Shah, Gauri: Skip graphs. In: *ACM Transactions on Algorithms* 3 (2007), Nr. 4, S. 37

[Aur03] Aura, Tuomas: Cryptographically Generated Addresses (CGA). In: *Information Security*. Springer Berlin / Heidelberg, 2003, S. 29–43

[Aur05] Aura, Tuomas: *Cryptographically Generated Addresses (CGA)*. IETF Request for Comments 3972 (Proposed standard). `http://www.ietf.org/rfc/rfc3972.txt`. Version: 2005

[Bar64] Baran, Paul: *On Distributed Communications*. RAND Corperation, 1964

[Bar99] Bard, Jonathan F.: *Practical Bilevel Optimization: Algorithms and Applications*. Kluwer Academic Publishers, 1999. – ISBN 978–0792354581

[BBF+08] Braun, David ; Buford, John F. ; Fish, Robert S. ; Gelman, Alexander D. ; Kaplan, Alan ; Khandelwal, Rajesh ; Narayanan, Sathya R. ; Shim, Eunsoo ; Yu, Heather: UP2P: A Peer-to-Peer Overlay Architecture for Ubiquitous Communications and Networking. In: *IEEE Communications Magazine* 46 (2008), Nr. 12, S. 32–39

[BD04] Burmester, Mike ; Desmedt, Yvo: Is hierarchical public-key certification the next target for hackers? In: *Communications of the ACM* 47 (2004), Nr. 8, S. 68–74

[Beh06] Behringer, Michael H.: *Analysis of the Security of BGP/MPLS IP Virtual Private Networks (VPNs)*. IETF Request for Comments 4381 (Proposed standard). http://www.ietf.org/rfc/rfc4381.txt. Version: 2006

[BFG+09] Brinkmeier, Michael ; Fischer, Mathias ; Grau, Sascha ; Schaefer, Guenter ; Strufe, Thorsten: Methods for Improving Resilience in Communication Networks and P2P Overlays. In: *PIK – Praxis der Informationsverarbeitung und Kommunikation* 32 (2009), S. 64–78

[Bha08] *Kapitel* Part III: Data Privacy. In: Bhaiji, Yusuf: *Network Security Technologies and Solutions*. 1st. Cisco Press, 2008. – ISBN 978–1–58705–246–0

[BHHW03] Baugher, Mark ; Hardjono, Thomas ; Harney, Hugh ; Weis, Brian: *The Group Domain of Interpretation*. IETF Request for Comments 3547 (Proposed standard). http://www.ietf.org/rfc/rfc3547.txt. Version: 2003

[Bib75] Biba, Kenneth J.: Integrity Considerations for Secure Computer Systems. 1975 (MTR–3153). – Forschungsbericht. – MITRE

[BKW05] Bollapragada, Vijay ; Khalid, Mohamed ; Wainner, Scott: *IPSec VPN Design*. Cisco Press, 2005. – ISBN 1–58705–111–7

[BL73] Bell, David E. ; LaPadula, Leonard J.: Secure Computer Systems: Mathematical Foundations. 1973 (MTR–2547). – Forschungsbericht. – MITRE

[BLLM01] Baltatu, Madalina ; Lioy, Antonio ; Lombardo, Dario ; Mazzocchi, Daniele: Towards a policy system for IPsec: issues and an experimental implementation. In: *Proceedings of 9th IEEE International Conference on Networks (ICON)*, 2001, S. 146–151

[BNW05] Bu, Tian ; Norden, Samphel ; Woo, Thomas: A survivable DoS-resistant overlay network. In: *Computer Networks* 50 (2005), S. 1281–1301

[BOH09] Bos, Joppe W. ; Özen, Onur ; Hubaux, Jean-Pierre: Analysis and Optimization of Cryptographically Generated Addresses. In: *Information Security*. Springer Berlin / Heidelberg, 2009, S. 17–32

[BP01] Beaulieu, Stephane ; Pereira, Roy: *Extended Authentication within IKE (XAUTH)*. Expired Internet Draft. http://tools.ietf.org/html/draft-ietf-ipsec-isakmp-xauth-06. Version: 2001

[Bra10] Braun, Michael: *Methoden zur Stabilitätsbewertung und -optimierung virtueller privater Netztopologien*. Bachelorarbeit, Fachgebiet Telematik/Rechnernetze, Betreuung: Michael Roßberg, 2010

[Bro01] Broderick, J. S.: Implementing Virtual Private Networks in Today's Organization. In: *Information Security Technical Report* 6 (2001), Nr. 1, S. 23–30

[Bro10] Broersma, Matthew: *Botnet price for hourly hire on par with cost of two pints*. http://www.zdnet.co.uk/news/security-threats/2010/05/25/botnet-price-for-hourly-hire-on-par-with-cost-of-two-\pints-40089028/. Version: 25. Mai 2010

[BRS09] Brinkmeier, Michael ; Rossberg, Michael ; Schaefer, Guenter: Towards a Denial-of-Service Resilient Design of Complex IPsec Overlays. In: *Proceedings of International Conference on Communications (ICC)*, 2009

[Bru08] Brunner, Tobias: *IKEv2 Mediation Extension*. Expired Internet Draft. http://tools.ietf.org/html/draft-brunner-ikev2-mediation-00. Version: 2008

[BS04] Bernstein, Jeff ; Spets, Tim: CPE WAN Management Protocol. Version: 2004. http://www.broadband-forum.org/technical/download/TR-069.pdf. 2004 (TR-069). – Forschungsbericht. – DSL Forum

[BSBK09] Bilge, Leyla ; Strufe, Thorsten ; Balzarotti, Davide ; Kirda, Engin: All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks. In: *Proceedings of the 18th International World Wide Web Conference*, 2009, S. 551–560

[Bun10] Bundesgerichtshof: *Haftung für unzureichend gesicherten WLAN-Anschluss*. BGH, Urteil vom 12. Mai 2010 - I ZR 121/08 - OLG Frankfurt/Main. http://juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?Gericht=bgh&Art=pm&Datum=2010&Sort=3&anz=113&pos=12&nr=52202&linked=urt&Blank=1&file=dokument.pdf. Version: 2010

[BYL08] Buford, John ; Yu, Heather ; Lua, Eng K.: *P2P Networking and Applications*. Morgan Kaufmann, 2008. – ISBN 978–0–123–74214–8

[CCN+06] Caesar, Matthew ; Castro, Miguel ; Nightingale, Edmund B. ; Arbor, Ann ; O, Gerg ; Rowstron, Antony: Virtual Ring Routing: Network Routing Inspired by DHTs. In: *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, 2006, S. 363–374

[CCRK03] Costa, Manuel ; Castro, Miguel ; Rowstron, Antony ; Key, Peter: PIC: Practical Internet Coordinates for Distance Estimation. In: *In International Conference on Distributed Systems*, 2003, S. 178–187

[CDK+03] Cox, Russ ; Dabek, Frank ; Kaashoek, Frans ; Li, Jinyang ; Morris, Robert: Practical, Distributed Network Coordinates. In: *Second Workshop on Hot Topics in Networks*, 2003

[CDKR02] Castro, Miguel ; Druschel, Peter ; Kermarrec, Anne-Marie ; Rowstron, Antony: Scribe: a large-scale and decentralized application-level multicast infrastructure. In: *IEEE Journal on Selected Areas in Communications (JSAC)* 20 (2002), Nr. 8, S. 1489–1499

[Cis04] Cisco Systems Inc.: *Easy VPN Configuration Example*. White Paper. http://www.cisco.com/en/US/docs/routers/access/1800/1841/software/configuration/guide/ezvpn.pdf. Version: 2004

[Cis06a] Cisco Systems Inc.: *Cisco IPsec VPN Implementation Group Password Usage Vulnerability*. Cisco Security Notice. http://www.cisco.com/warp/public/707/cisco-sn-20040415-grppass.pdf. Version: 2006

[Cis06b] Cisco Systems, Inc.: *Dynamic Multipoint VPN (DMVPN)*. http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ftgreips.pdf. Version: 2006

[Cis07] Cisco Systems Inc.: *Cisco Group Encrypted Transport VPN*. Cisco Feature Guide. http://www.cisco.com/en/US/docs/ios/12_4t/12_4t11/htgetvpn.pdf. Version: 2007

[CK10] Cheshire, Stuart ; Krochmal, Marc: *DNS-Based Service Discovery*. Internet Draft. http://tools.ietf.org/html/draft-cheshire-dnsext-dns-sd-07. Version: 2010

[CM02] Cheung, K. H. ; Mišic, J.: On virtual private networks security design issues. In: *Computer Networks* 38 (2002), February, S. 165–179

[Com03] Committee on the Internet Under Crisis Conditions: *The Internet Under Crisis Conditions: Learning from September 11*. National Research Council, 2003. – ISBN 0–309–50404–X

[CSWH01] Clarke, Ian ; Sandberg, Oskar ; Wiley, Brandon ; Hong, Theodore W.: Freenet: A Distributed Anonymous Information Storage and Retrieval System. In: *Designing Privacy Enhancing Technologies*. Springer Berlin / Heidelberg, 2001, S. 46–66

[DA08] Deri, Luca ; Andrews, Richard: N2N: A Layer Two Peer-to-Peer VPN. In: *Resilient Networks and Services*. Springer Berlin / Heidelberg, 2008, S. 53–64

[DA10] Deri, Luca ; Andrews, Richard: *n2n: a Layer Two Peer-to-Peer VPN*. http://www.ntop.org/n2n/. Version: 2010

[DCKM04] Dabek, Frank ; Cox, Russ ; Kaashoek, Frans ; Morris, Robert: Vivaldi: A Decentralized Network Coordinate System. In: *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, 2004, S. 15–26

[Del09] Deloitte: *Losing Ground – 2009 TMT Global Security Survey Full report*. http://www.deloitte.com/assets/Dcom-Germany/Local%20Assets/Documents/dtt_TMT-Security-Survey09-full(1).pdf. Version: July 2009

[Dem02] Dempe, Stephan: *Foundations of Bilevel Programming*. Kluwer Academic Publishers, 2002. – ISBN 978–1–4020–0631–9

[Den76] Denning, Dorothy E.: A Lattice Model of Secure Information Flow. In: *Communications of the ACM* 19 (1976), Nr. 5, S. 236–243

[DJWC08] Dinger, Jochen ; Jünemann, Konrad ; Waldhorst, Oliver ; Conrad, Michael: Autonome Kommunikationsinfrastrukturen – Eine praxisnahe Betrachtung. In: *PIK – Praxis der Informationsverarbeitung und Kommunikation* 31 (2008), S. 69–75

[DLL+00] Diot, Christophe ; Levine, Brian N. ; Lyles, Bryan ; Kassem, Hassan ; Balensiefen, Doug: Deployment Issues for the IP Multicast Service and Architecture. In: *IEEE Network* 14 (2000), Nr. 1, S. 78–88

[DMP+04] DaSilva, Luiz A. ; Midkiff, Scott F. ; Park, Jahng S. ; Hadjichristofi, George C. ; Davis, Nathaniel J. ; Phanse, Kaustubh S. ; Lin, Tao: Network mobility and protocol interoperability in ad hoc networks. In: *IEEE Communications Magazine* 42 (2004), Nr. 11, S. 88–96

[DMS04] Dingledine, Roger ; Mathewson, Nick ; Syverson, Paul: Tor: The Second-Generation Onion Router. In: *Proceedings of the 13th conference on USENIX Security Symposium (SSYM'04)*, 2004, S. 303–320

[Dou02] Douceur, John R.: The Sybil Attack. In: *Peer-to-Peer Systems* Bd. 2429. Springer Berlin / Heidelberg, 2002, S. 251–260

[DOW92] Diffie, Whitfield ; Oorschot, Paul C. ; Wiener, Michael J.: Authentication and Authenticated Key Exchanges. In: *Designs, Codes and Cryptography* 2 (1992), S. 107–125

[DR08] Dierks, Tim ; Rescorla, Eric: *The Transport Layer Security (TLS) Protocol*. IETF Request for Comments 5246 (Proposed standard). `http://www.ietf.org/rfc/rfc5246.txt`. Version: 2008

[DW09] Dinger, Jochen ; Waldhorst, Oliver P.: Decentralized Bootstrapping of P2P Systems: A Practical View. In: *In Proceedings of NETWORKING 2009*, Springer Berlin / Heidelberg, 2009 (Lecture Notes in Computer Science), S. 703–715

[Eck09] Eckert, Claudia: *IT-Sicherheit: Konzepte - Verfahren - Protokolle*. 6. überarbeitete und erweiterte Auflage. Oldenbourg, 2009. – ISBN 978–3486589993

[Edd07] Eddy, Wesley M.: *TCP SYN Flooding Attacks and Common Mitigations*. IETF Request for Comments 4987 (Proposed standard). `http://www.ietf.org/rfc/4987.txt`. Version: 2007

[EHL05] Ek, Mattias ; Hultin, Fredrik ; Lindblom, Jan: WASTE Peer-to-Peer Protocol. Version: 2005. `http://sourceforge.net/projects/j-waste/files/WASTE%20Documentation/WASTE%20Documentation%201.1/waste_documentation-1.1.pdf/download`. 2005. – Forschungsbericht. – Luleå tekniska universitet

[Ero06] Eronen, Pasi: *IKEv2 Mobility and Multihoming Protocol (MOBIKE)*. IETF Request for Comments 4555 (Proposed standard). http://www.ietf.org/rfc/rfc4555.txt. Version: 2006

[FBJW08] Figueiredo, Renato ; Boykin, P. O. ; Juste, Pierre S. ; Wolinsky, David: Social VPNs: Integrating Overlay and Social Networks for Seamless P2P Networking. In: *17th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE/COPS)*, 2008, S. 93–98

[FC06] Fu, Xiaoming ; Crowcroft, Jon: GONE: an Infrastructure Overlay for Resilient, DoS-Limiting Networking. In: *Proceedings of ACM Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 2006

[FDKC06] Fuhrmann, Thomas ; Di, Pengfei ; Kutzner, Kendy ; Cramer, Curt: Pushing Chord into the Underlay: Scalable Routing for Hybrid MANETs / Fakultät für Informatik, Universität Karlsruhe. Version: 2006. http://i30www.ira.uka.de/research/publications/p2p/. 2006 (2006-12). – Technischer Bericht

[Fei06] Feilner, Markus: *OpenVPN: Building and Integrating Virtual Private Networks*. Packt Publishing, 2006. – ISBN 1–904–81185–X

[Flu00] Fluhrer, Scott: *Tunnel Endpoint Discovery*. Expired Internet Draft. http://tools.ietf.org/html/draft-fluhrer-ted-00. Version: 2000

[Flu07a] Fluhrer, Scott: *Method and apparatus for determining secure endpoints of tunnels in a network that uses internet security protocol*. United States Patent US 2007/7207063 B1, 2007

[Flu07b] Fluhrer, Scott: *SYSTEM AND METHOD FOR PROTECTED SPOKE TO SPOKE COMMUNICATION USING AN UNPROTECTED COMPUTER NETWORK*. United States Patent US 2007/0271451 A1, 2007

[For94] Ford, Warwick: *Computer Communications Security: Principles, Standard Protocols and Techniques*. Prentice Hall, 1994. – ISBN 978–013–799453–3

[For03] Ford, Bryan: Unmanaged Internet Protocol: Taming the Edge Network Management Crisis. In: *Second Workshop on Hot Topics in Networks*, 2003

[For08] Ford, Bryan A.: *UIA: A Global Connectivity Architecture for Mobile Personal Devices*, Massachusetts Institute of Technology, Diss., 2008

[For09] Forrester Consulting: *The Trends And Changing Landscape Of DDoS Threats And Protection*. https://www.verisign.com/ddos-protection/resources/whitepaper-ddos-threats-protection-forrester.pdf. Version: July 2009

[Fra01] Frankel, Sheila: *Demystifying the IPsec Puzzle*. Artech House Publishers, 2001

[FS00] Ferguson, Niels ; Schneier, Bruce: A Cryptographic Evaluation of IPsec / Counterpane Internet Security, Inc. Version: 2000. http://www.schneier.com/paper-ipsec.pdf. 2000. – Forschungsbericht

[FSK05] Ford, Bryan ; Srisuresh, Pyda ; Kegel, Dan: Peer-to-Peer Communication Across Network Address Translators. In: *Proceedings of the USENIX 2005 Annual Technical Conference*, 2005, S. 179–192

[Fuh05] Fuhrmann, Thomas: The Use of Scalable Source Routing for Networked Sensors. In: *Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors*, 2005, S. 163–165

[GABF06] Ganguly, Arijit ; Agrawal, Abhishek ; Boykin, P. O. ; Figueiredo, Renato: IP over P2P: enabling self-configuring virtual IP networks for grid computing. In: *20th International Parallel and Distributed Processing Symposium (IPDPS)*, 2006

[Gar00] Garber, Lee: Denial-of-Service Attacks Rip the Internet. In: *Computer* 33 (2000), Nr. 4, S. 12–17

[gem09] gematik – Gesellschaft für Telematikanwendungen in der Gesundheitskarte mbH: *Konnektorspezifikation Version 3.0.0*. `http://www.gematik.de/cms/media/dokumente/release_4_0_0/DezentraleKomponenten.zip`. Version: September 2009

[Gen01] Gentry, Perry B.: What is a VPN? In: *Information Security Technical Report* 6 (2001), Nr. 1, S. 15–22

[GHJV94] Gamma, Erich ; Helm, Richard ; Johnson, Ralph E. ; Vlissides, John: *Design Patterns. Elements of Reusable Object-Oriented Software.* Addison-Wesley Longman, 1994. – ISBN 0–201–63361–2

[Gin10] Ginolas, Wolfgang: *P2PVPN – Documentation*. `http://p2pvpn.org/documentation.html`. Version: 2010

[Gol10] Golembewski, René: *Sichere, verteilte Zeitsynchronisation in virtuellen privaten Netzwerken*. Diplomarbeit, Fachgebiet Telematik/Rechnernetze, Betreuung: Michael Roßberg, 2010

[GRS+10a] Girlich, Franz ; Rossberg, Michael ; Schaefer, Guenter ; Boehme, Thomas ; Schreyer, Jens: *Scrubbing the Vivaldi network coordinate system*. Systems and Network Security Doctoral Workshop (IWK 2010), September 2010

[GRS10b] Golembewski, René ; Rossberg, Michael ; Schaefer, Guenter: *DoS-resistant Distributed Time Synchronization for Virtual Private Networks*. Systems and Network Security Doctoral Workshop (IWK 2010), September 2010

[GT03] Grusho, Alexander ; Timonina, Elena: Construction of the Covert Channels. In: Gorodetsky, Vladimir (Hrsg.) ; Popyack, Leonard (Hrsg.) ; Skormin, Victor (Hrsg.): *Computer Network Security* Bd. 2776. Springer Berlin / Heidelberg, 2003, S. 428–431

[GTEB05] Ghosh, Abhrajit ; Talpade, Rajesh ; Elaoud, Moncef ; Bereschinsky, Michael: Securing ad-hoc networks using IPsec. In: *Proceedings of the IEEE Military Communications Conference (MILCOM)*, 2005, S. 2948–2953

[Ham09] Hamblen, Matt: *Verizon expands anti-DoS protection*. `http://news.techworld.com/security/110661/verizon-expands-anti-dos-protection/`. Version: February 2009

[HASG07] Hosseini, Mojtaba ; Ahmed, Dewan T. ; Shirmohammadi, Shervin ; Georganas, Nicolas D.: A Survey of Application-Layer Multicast Protocols. In: *IEEE Communications Surveys Tutorials* 9 (2007), Nr. 3, S. 58–74

[HD03] Hardjono, Thomas ; Dondeti, Lakshminath R.: *Multicast and Group Security*. Artech House Publishers, 2003. – ISBN 978–158–053342–3

[Hei09] Heinlein, Alexander: *Konzept und Implementierung einer Multicast-Architektur für virtuelle private Netze*. Bachelorarbeit, Fachgebiet Telematik/Rechnernetze, Betreuung: Michael Roßberg, 2009

[HJS+03] Harvey, Nicholas J. A. ; Jones, Michael B. ; Saroiu, Stefan ; Theimer, Marvin ; Wolman, Alec: SkipNet: a scalable overlay network with practical locality properties. In: *Proceedings of the 4th Symposium on Internet Technologies and Systems (USITS)*, 2003

[HKE09] Hanigk, Sebastian ; Kretzschmar, Michael ; Eyermann, Frank: A Distributed Routing Architecture for Secure Communication over Highly Dynamic Radio Networks. In: *Communication Systems and Networks and Workshops*, 2009

[HLL+07] Hei, Xiaojun ; Liang, Chao ; Liang, Jian ; Liu, Yong ; Ross, Keith W.: A Measurement Study of a Large-Scale P2P IPTV System. In: *IEEE Transactions on Multimedia* 9 (2007), Nr. 8, S. 1672–1687

[HOI+04] Honda, Osamu ; Ohsaki, Hiroyuki ; Imase, Makoto ; Ishizuka, Mika ; Murayama, Junichi: Understanding TCP over TCP: Effects of TCP Tunneling on End-to-End Throughput and Latency. 2004. – Forschungsbericht

[HPV+99] Hamzeh, Kory ; Pall, Gurdeep S. ; Verthein, William ; Taarud, Jeff ; Little, W. A. ; Zorn, Glen: *Point-to-Point Tunneling Protocol (PPTP)*. IETF Request for Comments 2637 (Proposed standard). `http://www.ietf.org/rfc/2637.txt`. Version: 1999

[HRH01] Hoang, Dang-Hai ; Reschke, Dietrich ; Horn, Werner: Adaptive Quality of Service Management Using QoS Proxy and User Feedback for Wireless Links. In: *Proceedings of International Workshop on Innovative Internet Computing Systems (IICS)*, 2001, S. 31–40

[HRS10] Heinlein, Alexander ; Rossberg, Michael ; Schaefer, Guenter: Secure Multicast in Internet-wide VPN. In: *Proceedings of Systems and Network Security Doctoral Workshop (IWK 2010)*, 2010

[HS04] Hutter, Dieter ; Schairer, Axel: Possibilistic Information Flow Control in the Presence of Encrypted Communication. In: *Proceedigs of ESORICS*, 2004, S. 209–224

[HSV+05] Huttunen, Ari ; Swander, Brian ; Volpe, Victor ; DiBurro, Larry ; Stenberg, Markus: *UDP Encapsulation of IPsec ESP Packets*. IETF Request for Comments 3948 (Proposed standard). `http://www.ietf.org/rfc/3948.txt`. Version: 2005

[Hu03] Hu, Jim: *AOL pulls Nullsoft file-sharing software*. CNET News. `http://news.cnet.com/2100-1032_3-1011585.html`. Version: 2003

[JB99] Juels, Ari ; Brainard, John: Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks. In: *Proceedings of Networks and Distributed Security Systems (NDSS '99)*, 1999, S. 151–165

[Jos06] Josefsson, Simon: *Storing Certificates in the Domain Name System (DNS)*. IETF Request for Comments 4398 (Proposed standard). `http://www.ietf.org/rfc/rfc4398.txt`. Version: 2006

[JRS+09] Jacob, Riko ; Richa, Andrea ; Scheideler, Christian ; Schmid, Stefan ; Täubig, Hanjo: A Distributed Polylogarithmic Time Algorithm for Self-Stabilizing Skip Graphs. In: *Proceedings of the 28th ACM symposium on Principles of distributed computing (PODC 09)*, 2009, S. 131–140

[Kam08] Kaminsky, Dan: Catching up with Kaminsky. In: *Network Security* 2008 (2008), Nr. 9, S. 4–7

[Kar94] Karn, Phil: *The Photuris Key Management Protocol*. Expired Internet Draft. `http://tools.ietf.org/html/draft-karn-photuris-00`. Version: 1994

[Kau05] Kaufman, Charlie: *Internet Key Exchange (IKEv2) Protocol*. IETF Request for Comments 4306 (Proposed standard). `http://www.ietf.org/rfc/rfc4306.txt`. Version: 2005

[KHW+09] Knoll, Mirko ; Helling, Matthias ; Wacker, Arno ; Holzapfel, Sebastian ; Weis, Torben: Bootstrapping Peer-to-Peer Systems using IRC. In: *Proceeedings of the 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE/COPS)*, 2009, S. 122–127

[KK03] Kaashoek, M. F. ; Karger, David R.: Koorde: A simple degree-optimal distributed hash table. In: *Proceedings of 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, 2003

[KK04] Khanvilkar, Shashank ; Khokhar, Ashfaq: Virtual Private Networks: An Overview with Performance Evaluation. In: *IEEE Communications Magazine* 42 (2004), Nr. 10, S. 146–154

[KKVS09] Kurian, Jinu ; Kulkarni, Ajay ; Vu, Hai T. ; Sarac, Kamil: ODON: An On-Demand Security Overlay for Mission-Critical Applications. In: *Proceedings of 18th International Conference on Computer Communications and Networks (ICCCN)*, 2009

[KLS00] Kent, Stephen ; Lynn, Charles ; Seo, Karen: Secure Border Gateway Protocol (S-BGP). In: *IEEE Journal on Selected Areas in Communications (JSAC)* 18 (2000), Nr. 4, S. 582–592

[KMR02] Keromytis, Angelos D. ; Misra, Vishal ; Rubenstein, Dan: SOS: Secure Overlay Services. In: *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, 2002, S. 61–72

[Knu84] Knuth, Donald E.: The Complexity of Songs. In: *Communications of the ACM* 27 (1984), Nr. 4, S. 344–346

[Kra09a] Kratochvil, Miroslav: *CloudVPN how it works*. `http://e-x-a.org/stuff/cloudvpn-poster.jpg`. Version: 2009

[Kra09b] Kravets, Wired.com, David: *Pirate Bay Launches VPN Service*. `http://www.wired.com/threatlevel/2009/06/ipredator`. Version: 2009

[KRAV03] Kostić, Dejan ; Rodriguez, Adolfo ; Albrecht, Jeannie ; Vahdat, Amin: Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh. In: *ACM SIGOPS Operating Systems Review* 37 (2003), Nr. 5, S. 282–297

[KRS11] Koellmer, Thomas ; Rossberg, Michael ; Schaefer, Guenter: Attributbasierter Schlüsselaustausch in virtuellen privaten Netzen. In: *Eingereicht zum 12. Deutscher IT-Sicherheitskongress*, 2011, S. 334–342

[KS05] Kent, Stephen ; Seo, Karen: *Security Architecture for the Internet Protocol*. IETF Request for Comments 4301 (Proposed standard). `http://www.ietf.org/rfc/rfc4301.txt`. Version: 2005

[KS06] Kurian, Jinu ; Sarac, Kamil: FONet: A Federated Overlay Network for DoS Defense in the Internet: A Position Paper. In: *IEEE INFOCOM Global Internet Workshop*, 2006

[Kö10] Köllmer, Thomas: *A secure authentication scheme for multi-level network architectures*. Studienarbeit, Fachgebiet Telematik/Rechnernetze, Betreuung: Michael Roßberg, 2010

[LAPS03] Lakshminarayanan, Karthik ; Adkins, Daniel ; Perrig, Adrian ; Stoica, Ion: Taming IP Packet Flooding Attacks. In: *Proceedings of Second Workshop on Hot Topics in Networks*, 2003

[LAWD04] Li, Lun ; Alderson, David ; Willinger, Walter ; Doyle, John: A First-Principles Approach to Understanding the Internet's Router-level Topology. In: *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, 2004, S. 3–14

[Lev98] Levy, Elias: The Crumbling Tunnel. In: *Phrack* 53 (1998). http://www.phrack.com/issues.html?issue=53&id=12&mode=txt

[LG02] Le, Van ; Guyennet, Herve: IPSec and DNSSEC to Support GRID Application Security. In: *IEEE International Symposium on Cluster Computing and the Grid*, 2002, S. 458–459

[Lip07] Lipp, Manfred: *VPN - Virtuelle Private Netzwerke: Aufbau und Sicherheit*. Addison-Wesley, 2007. – ISBN 978–382–732647–8

[LKS04] Lynn, Charles ; Kent, Stephen ; Seo, Karen: *X.509 Extensions for IP Addresses and AS Identifiers*. IETF Request for Comments 3779 (Proposed standard). http://www.ietf.org/rfc/3779.txt. Version: 2004

[LMK07] Laganier, Julien ; Montenegro, Gabriel ; Kukec, Ana: *Using IKE with IPv6 Cryptographically Generated Addresses*. Expired Internet Draft. http://tools.ietf.org/html/draft-laganier-ike-ipv6-cga-02. Version: 2007

[Log09] LogMeIn: *LogMeIn Hamachi^2 Security*. https://secure.logmein.com/US/products/hamachi2/security.aspx. Version: 2009

[Lö10] Löfberg, Johan: *Tutorial/Bilevel Programming*. http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Commands.Solvebilevel. Version: 2010

[MAHS10] Müller, Günter ; Accorsi, Rafael ; Höhn, Sebastian ; Sackmann, Stefan: Sichere Nutzungskontrolle für mehr Transparenz in Finanzmärkten. In: *Informatik Spektrum* 33 (2010), Nr. 1, S. 3–13

[Man07] Manral, Vishwas: *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*. IETF Request for Comments 4835 (Proposed standard). http://www.ietf.org/rfc/4835.txt. Version: 2007

[Mar07] Marcovitz, Alan: *Introduction to Logic and Computer Design*. McGraw-Hill Science/Engineering/Math, 2007. – ISBN 0–073–31417–4

[MBR03] Manku, Gurmeet S. ; Bawa, Mayank ; Raghavan, Prabhakar: Symphony: Distributed Hashing in a Small World. In: *Proceedings of the 4th Symposium on Internet Technologies and Systems (USITS)*, 2003

[MGMOK06] Merino, Pedro J. M. ; García-Martínez, Alberto ; Organero, Mario M. ; Kloos, Carlos D.: Enabling Practical IPsec Authentication for the Internet. In: *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, 2006, S. 392–403

[MHK+07] Mahadevan, Priya ; Hubble, Calvin ; Krioukov, Dmitri ; Huffaker, Bradley ; Vahdat, Amin: Orbis: Rescaling Degree Correlations to Generate Annotated Internet Topologies. In: *SIGCOMM Computer Communications Review* 37 (2007), Nr. 4, S. 325–336

[Mil92] Millen, Jonathan K.: A Resource Allocation Model for Denial of Service. In: *Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy*, 1992, S. 137–147

[Mit03] Mitchell, Chris J.: Cryptanalysis of Two Variants of PCBC Mode When Used for Message Integrity. In: *Information Security and Privacy*. Springer Berlin / Heidelberg, 2003, S. 560–571

[MJB+06] McCune, Jonathan M. ; Jaeger, Trent ; Berger, Stefan ; Caceres, Ramon ; Sailer, Reiner: Shamon: A System for Distributed Mandatory Access Control. In: *22nd Annual Computer Security Applications Conference*, 2006, S. 23–32

[MM00] Mcauley, A. J. ; Manousakis, K.: Self-configuring networks. In: *Proceedings of the IEEE Military Communications Conference (MILCOM)*, 2000, S. 315–319

[MM02] Maymounkov, Petar ; Mazieres, David: Kademlia: A peer-to-peer information system based on the XOR metric. In: *Proceedings of 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002

[MNR02] Malkhi, Dahlia ; Naor, Moni ; Ratajczak, David: Viceroy: A Scalable and Dynamic Emulation of the Butterfly. In: *Proceedings of the 21th Annual Symposium on Principles of Distributed Computing (PODC)*, 2002, S. 183–192

[MSI+02] Milo, Ron ; Shai, Shen-Orr ; Itzkovitz, Shalev ; Kashtan, Nadav ; Chklovskii, David ; Alon, Uri: Network motifs: simple building blocks of complex networks. In: *Science* 298 (2002), October, Nr. 5594, 824–827. http://dx.doi.org/10.1126/science.298.5594.824. – DOI 10.1126/science.298.5594.824. – ISSN 1095–9203

[MST+09] Malkhi, Dahlia ; Sen, Siddhartha ; Talwar, Kunal ; Werneck, Renato ; Wieder, Udi: Virtual Ring Routing Trends. In: *Proceedings of the 23rd International Symposium on Distributed Computing (DISC)*, 2009, S. 392–406

[Nir06] Nir, Yoav: *Repeated Authentication in Internet Key Exchange (IKEv2) Protocol*. IETF Request for Comments 4478 (Proposed standard). http://www.ietf.org/rfc/rfc4478.txt. Version: 2006

[NM08] Nikander, Pekka ; Melen, Jan: *A Bound End-to-End Tunnel (BEET) mode for ESP*. Expired Internet Draft. http://tools.ietf.org/html/draft-nikander-esp-beet-mode-09. Version: 2008

[NZ02] Ng, T. S. E. ; Zhang, Hui: Predicting Internet Network Distance with Coordinates-Based Approaches. In: *Proceeedings of the 21st Conference of the IEEE Computer and Communications (INFOCOM)* Bd. 1, 2002, S. 170–179

[Oll09] Ollmann, Gunter: *Want to rent an 80-120k DDoS Botnet?* http://blog.damballa.com/?p=330. Version: 28. August 2009

[Pas06] Pasko, Doug: *Security Expansion Layers Beyond the Firewall*. Verizon White Paper. `http://www.verizonbusiness.com/us/resources/security/managed/secureexpanlay.pdf`. Version: February 2006

[Pet10] Petric, Kresimir: *Wippien - Open source p2p VPN software*. `http://www.wippien.com/`. Version: 2010

[PSDY00] Patton, Samuel ; Smith, Bryan ; Doss, David ; Yurcik, William: A Layered Framework Strategy for Deploying High Assurance VPNs. In: *Proceedings of Fifth IEEE International Symposim on High Assurance Systems Engineering*, 2000, S. 199–202

[RD01] Rowstron, Antony ; Druschel, Peter: Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In: *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, 2001, S. 329–350

[Rei07] Reinhardt, BWI Informationstechnik GmbH, Jochen: *2.400 Bundeswehr-Mitarbeiter kommen jetzt zur BWI*. Statement to the Press (German). `http://www.bwi-it.de/presse_koeln_20-04-07.html`. Version: 2007

[Rel09] Relakks – Trygghetsbolaget i Lund AB: *Relakks – Questions and Answers*. `https://www.relakks.com/faq/qna/`. Version: 2009

[RFH+01] Ratnasamy, Sylvia ; Francis, Paul ; Handley, Mark ; Karp, Richard ; Shenker, Scott: A Scalable Content-Addressable Network. In: *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, 2001, S. 161–172

[RHKS01] Ratnasamy, Sylvia ; Handley, Mark ; Karp, Richard ; Shenker, Scott: Application-Level Multicast Using Content-Addressable Networks. In: Crowcroft, Jon (Hrsg.) ; Hofmann, Markus (Hrsg.): *Networked Group Communication* Bd. 2233. Springer Berlin / Heidelberg, 2001, S. 14–29

[Ric05] Richardson, Michael C.: *A Method for Storing IPsec Keying Material in DNS*. IETF Request for Comments 4025 (Proposed standard). `http://www.ietf.org/rfc/rfc4025.txt`. Version: 2005

[RM04] Rescorla, Eric ; Modadugu, Nagendra: The Design and Implementation of Datagram TLS. In: *ISOC Network and Distributed System Security (NDSS) Symposium*, 2004

[RM06] Rescorla, Eric ; Modadugu, Nagendra: *Datagram Transport Layer Security*. IETF Request for Comments 4347 (Proposed standard). `http://www.ietf.org/rfc/rfc4347.txt`. Version: 2006

[Roe10] Roethlisberger, Daniel: *FakeIKEd*. `http://www.roe.ch/FakeIKEd`. Version: 2010

[Ros07] Rossberg, Michael: *Peer–to–Peer–basierte Autokonfiguration komplexer virtueller privater IPsec–Netze*. Diplomarbeit, Fachgebiet Telematik/Rechnernetze, Betreuung: Guenter Schaefer und Thorsten Strufe, 2007

[Ros10] Rossberg, Michael: Multiple Security Issues in Wippien. Version: 2010. `http://www.tu-ilmenau.de/fileadmin/media/telematik/rossberg/wippien-advisory.txt`. 2010. – Forschungsbericht. – Ilmenau University of Technology

[RR05] Richardson, Michael C. ; Redelmeier, D. H.: *Opportunistic Encryption using the Internet Key Exchange (IKE)*. IETF Request for Comments 4322 (Proposed standard). `http://www.ietf.org/rfc/rfc4322.txt`. Version: 2005

[RRDO10] Rescorla, Eric ; Ray, Marsh ; Dispensa, Steve ; Oskov, Nasko: *Transport Layer Security (TLS) Renegotiation Indication Extension*. Internet Draft. `http://tools.ietf.org/id/draft-ietf-tls-renegotiation-03.txt`. Version: 2010

[RRKC04] Raghunath, Satish ; Ramakrishnan, K. K. ; Kalyanaraman, Shivkumar ; Chase, Chris: Measurement based characterization and provisioning of IP VPNs. In: *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, 2004, S. 342–355

[RS09] Rossberg, Michael ; Schaefer, Guenter: Ciscos Group Encrypted Transport VPN – A sceptical analysis. In: *Proceedings of D-A-CH security, German*, 2009, S. 351–360

[RS11] Rossberg, Michael ; Schaefer, Guenter: A Survey on Automatic Configuration of Virtual Private Networks. In: *Erscheint im Computer Networks Journal* 55 (2011), 1684–1699. `http://dx.doi.org/10.1016/j.comnet.2011.01.003`

[RSM09] Rossberg, Michael ; Schaefer, Guenter ; Martius, Kai: *Method for configuring the secure transmission of data between secure networks*. Europa-Patent EP20080010510, angemeldet am 10. Juni 2008, veröffentlicht am 16.12.2009, 2009

[RSM10a] Rossberg, Michael ; Schaefer, Guenter ; Martius, Kai: Automatic configuration of complex IPsec-VPNs and implications to higher layer network management. In: *Proceedings of Information Security Solutions Europe (iSSE)*, 2010

[RSM10b] Rossberg, Michael ; Schaefer, Guenter ; Martius, Kai: Secure Overlay-Based Auto-configuration of Complex IPsec VPNs. In: *Proceedings of Future Security*, 2010

[RSS07] Rossberg, Michael ; Schaefer, Guenter ; Strufe, Thorsten: Using Reccuring Costs for Reputation Management in Peer-to-Peer Streaming

Systems. In: *Proceedings of 3rd International Conference on Security and Privacy in Communications (SecureComm)*, 2007, S. 283–292

[RSS10] Rossberg, Michael ; Schaefer, Guenter ; Strufe, Thorsten: Distributed Automatic Configuration of Complex IPsec-Infrastructures. In: *Journal of Network and Systems Management* 18 (2010), S. 300–326

[RSSM09] Rossberg, Michael ; Steudel, Wolfgang ; Schaefer, Guenter ; Martius, Kai: Eine Software-Architektur zur Konstruktion flexibler IPsec-Infrastrukturen. In: *11. Deutscher IT-Sicherheitskongress*, 2009, S. 297–308

[SAA+99] Savage, Stefan ; Anderson, Tom ; Aggarawl, Amit ; Anderson, Tom ; Aggarwal, Amit ; Becker, David ; Cardwell, Neal ; Collins, Andy ; Hoffman, Eric ; Snell, John ; Vahdat, Amin ; Voelker, Geoff ; Zahorjan, John: Detour: a Case for Informed Internet Routing and Transport. In: *IEEE Micro* 19 (1999), S. 50–59

[SAS+02] Stoica, Ion ; Adkins, Daniel ; Shenker, Scott ; Surana, Sonesh ; Zhuang, Shelley: Internet Indirection Infrastructure. In: *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, 2002, S. 73–86

[SC02] Sanchez, Luis A. ; Condell, Matthew N.: *Security Policy Protocol*. Expired Internet Draft. `http://tools.ietf.org/html/draft-ietf-ipsp-spp-01`. Version: 2002

[Sch03] Schäfer, Günter: *Netzsicherheit: Algorithmische Grundlagen und Protokolle*. dpunkt.verlag GmbH, 2003. – ISBN 978–3898642125

[Sch10] Schüttler, Florian: *Sichere dezentrale Namensauflösung in IPsec-Infrastrukturen*. Bachelorarbeit, Fachgebiet Telematik/Rechnernetze, Betreuung: Michael Roßberg, 2010

[SG00] Stamatelakis, Demetrios ; Grover, Wayne D.: IP layer restoration and network planning based on virtual protection cycles. In: *IEEE Journal on Selected Areas in Communications (JSAC)* 18 (2000), Nr. 10, S. 1938–1949

[SKN+05] Stavrou, Angelos ; Keromytis, Angelos D. ; Nieh, Jason ; Misra, Vishal ; Rubenstein, Dan: MOVE: An End-to-End Solution To Network Denial of Service. In: *Proceedings of the ISOC Symposium on Network and Distributed System Security (SNDSS)*, 2005

[Sli10] Sliepen, Guus: *The difficulties of a peer-to-peer VPN on the hostile Internet*. Slides of Talk at FOSDEM. `http://www.tinc-vpn.org/presentations/fosdem-2010/tinc_fosdem2010_slides.pdf`. Version: 2010

[SMK+01] Stoica, Ion ; Morris, Robert ; Karger, David ; Kaashoek, M. F. ; Balakrishnan, Hari: Chord: A scalable peer-to-peer lookup service for internet applications. In: *ACM SIGCOMM Computer Communication Review* 31 (2001), Nr. 4, S. 149–160. – ISSN 0146–4833

[SNDW06] Singh, Atul ; Ngan, Tsuen-Wan J. ; Druschel, Peter ; Wallach, Dan S.: Eclipse Attacks on Overlay Networks: Threats and Defenses. In: *Proceedings of the 25th Conference of the IEEE Computer and Communications (INFOCOM)*, 2006

[SS99] Srisuresh, Pyda ; Sanchez, Luis A.: *Policy Framework for IP Security*. Expired Internet Draft. `http://tools.ietf.org/html/draft-ietf-ipsec-policy-framework-00`. Version: 1999

[SSAP06] Shi, Elaine ; Stoica, Ion ; Andersen, David ; Perrig, Adrian: OverDoSe: A Generic DDoS Protection Service Using an Overlay Network. 2006 (CMU-CS-06-114). – Forschungsbericht. – Carnegie Mellon University

[Sta10] Stadie, Daniel: *Automatische Konfiguration virtueller privater Netze in IPv6-Umgebungen*. Studienarbeit, Fachgebiet Telematik/Rechnernetze, Betreuung: Michael Roßberg, 2010

[Ste08a] Steffen, Andreas: strongSwan VPNs scalable and modularized! In: *Proceedings of LinuxTag*, 2008

[Ste08b] Steudel, Wolfgang: *Implementierung und Analyse eines Peer-to-Peer-basierten IPsec-Autokonfigurations-Ansatzes*. Diplomarbeit, Fachgebiet Telematik/Rechnernetze, Betreuung: Michael Roßberg, 2008

[Str07] Strufe, Thorsten: *Ein Peer–to–Peer-basierter Ansatz zur Live-Übertragung multimedialer Datenströme*. Dissertation, Fachgebiet Telematik/Rechnernetze, Betreuung: Guenter Schaefer, 2007

[SW05] Steinmetz, Ralf ; Wehrle, Klaus: *Lecture Notes in Computer Science*. Bd. 3485: *Peer-to-Peer Systems and Applications*. Springer, 2005. – ISBN 3–540–29192–X

[SZP+09] Sontag, David ; Zhang, Yang ; Phanishayee, Amar ; Andersen, David G. ; Karger, David: Scaling All-Pairs Overlay Routing. In: *Proceedings of the 5th international conference on Emerging networking experiments and technologies (CoNEXT)*, 2009, S. 145–156

[TEW04] Touch, Joe ; Eggert, Lars ; Wang, Yu-Shun: *Use of IPsec Transport Mode for Dynamic Routing*. IETF Request for Comments 3884 (Proposed Standard). `http://www.ietf.org/rfc/rfc3884.txt`. Version: 2004

[Tra06] Tran, Trung: Proactive Multicast-Based IPSEC Discovery Protocol and Multicast Extension. In: *Proceedings of the IEEE Military Communications Conference (MILCOM)*, 2006

[TVR+99] Townsley, W. M. ; Valencia, Andrew J. ; Rubens, Allan ; Pall, Gurdeep S. ; Zorn, Glen ; Palter, Bill: *Layer Two Tunneling Protocol „L2TP“*. IETF Request for Comments 2661 (Proposed standard). `http://www.ietf.org/rfc/2661.txt`. Version: 1999

[TWP08] Tews, Erik ; Weinmann, Ralf-Philipp ; Pyshkin, Andrei: Breaking 104 Bit WEP in Less Than 60 Seconds. In: *Information Security Applications*. Springer Berlin / Heidelberg, 2008, S. 188–202

[UPS09] Urdaneta, Guido ; Pierre, Guillaume ; Steen, Maarten van: A Survey of DHT Security Techniques. In: *ACM Computing Surveys* (2009). – `http://www.globule.org/publi/SDST_acmcs2009.html`, to appear

[Ver10] VeriSign: *DDoS Mitigation - Best Practices for a Rapidly Changing Threat Landscape*. `http://www.verisign.com/forms/ddosbestpracticeswp.html`. Version: April 2010

[VH08] Varga, András ; Hornig, Rudolf: An overview of the OMNeT++ simulation environment. In: *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops (Simutools)*, 2008, S. 1–10

[VLO09] Vu, Quang H. ; Lupu, Mihai ; Ooi, Beng C.: *Peer-to-Peer Computing: Principles and Applications*. Springer, 2009. – ISBN 978–3–642–03513–5

[VPN09] *VPN (The Easy Way) V24+*. `http://www.dd-wrt.com/wiki/index.php/VPN_%28the_easy_way%29_v24%2B`. Version: 2009

[WJBF10] Wolinsky, David I. ; Juste, Pierre S. ; Boykin, P. O. ; Figueiredo, Renato: Addressing the P2P Bootstrap Problem for Small Networks. In: *Proceedings of the 10th IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2010

[WLC03] Wang, Ju ; Lu, Linyuan ; Chien, Andrew A.: Tolerating denial-of-service attacks using overlay networks: impact of topology. In: *SSRS '03: Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems*, 2003, S. 43–52

[WR08] Williams, Nicolas ; Richardson, Michael C.: *Better-Than-Nothing Security: An Unauthenticated Mode of IPsec*. IETF Request for Comments 5386 (Proposed standard). `http://www.ietf.org/rfc/rfc5386.txt`. Version: 2008

[WR09] Weis, Brian ; Rowles, Sheela: *Updates to the Group Domain of Interpretation (GDOI)*. MSEC Working Group Internet Draft. `http://tools.ietf.org/html/draft-ietf-msec-gdoi-update-04`. Version: 2009

[WSS05] Wong, Bernard ; Slivkins, Aleksandrs ; Sirer, Emin G.: Meridian: A Lightweight Network Location Service without Virtual Coordinates. In: *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, 2005, S. 85–96

[ZF10] Zuquete, Andre ; Frade, Carlos: Fast VPN mobility across Wi-Fi hotspots. In: *2nd International Workshop on Security and Communication Networks (IWSCN)*, 2010

[ZLLY05] Zhang, Xinyan ; Liu, Jiangchuan ; Li, Bo ; Yum, Y.-S.P.: CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming. In: *Proceeedings of the 24st Conference of the IEEE Computer and Communications (INFOCOM)* Bd. 3, 2005, S. 2102–2111

[ZLS+05] Zhuang, Shelley ; Lai, Kevin ; Stoica, Ion ; Katz, Randy ; Shenker, Scott: Host Mobility Using an Internet Indirection Infrastructure. In: *Wireless Networks* 11 (2005), Nr. 6, S. 741–756

Abkürzungsverzeichnis

AES	Advanced Encryption Standard
ACL	Access Control List
AH	Authentication Header
ALM	Application Layer Multicast
BEET	Bound End-to-End Tunnel
BTNS	Better-Than-Nothing Security
CA	Certificate Authority
CAN	Content Addressable Network
CBC	Cipher-Block-Chaining
CGA	Cryptographically Generated Addresses
DHCP	Dynamic Host Configuration Protocol
DHT	Distributed Hash Table
DMVPN	Dynamic Multipoint VPN
DNS	Domain Name System
DoS	Denial-of-Service
DSCP	Differential Service Code Point
DTLS	Datagram Transport Layer Security
EIGRP	Enhanced Interior Gateway Routing Protocol
ESP	Encapsulating Security Payload
GDOI	Group Domain of Interpretation
GET	Group Encrypted Transport
HOT	Heuristically Optimal Topology
HMAC	Hash-based Message Authentication Code
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Management Protocol
IETF	Internet Engineering Task Force

IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPsec	IP security
IKE	Internet Key Exchange
ISP	Internet Service Provider
IV	Initialisierungsvektor
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
MANET	Mobile Ad–hoc–Netze
MPLS	Multiprotocol Label Switching
MOBIKE	IKEv2 Mobility and Multihoming Protocol
MTU	Maximum Transmission Unit
NAT	Network Address Translation
NTP	Network Time Protocol
OSPF	Open Shortest Path First
PFS	Perfect Forward Secrecy
PIM	Protocol Independent Multicast
PPTP	Point-to-Point Tunneling Protocol
QoS	Quality of Service
RADIUS	Remote Authentication Dial In User Service
RTT	Round Trip Time
SLA	Service Level Agreement
SOLID	Secure OverLay for IPsec Discovery
SPI	Security Parameter Index
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TED	Tunnel Endpoint Discovery
TLS	Transport Layer Security
UDP	User Datagram Protocol
VPN	virtuelles privates Netzwerk
WLAN	Wireless Local Area Network

Stichwortverzeichnis

Adresse
  äußere, 21
  innere, 21
Adresszuweisung, 71, 75–76, 134
Agilität, 37, 77, 147, 187
Anfälligkeit gegen Sabotage, 111, 184–186
Anforderungen, 35–41, 127
Angreifer
  -modell, 72–73
  -strategien, 20–21, 32–33
  externe, 72, 106–112, 148–150
  interne, 72, 120–123, 150–154
Angriffsresistenz, 106–112
Anonymität, 38
Application Layer Multicast, 31, 100
Authentizität, 19, 71, 155
Autokonfiguration, 35, 143

Beobachtungswahrscheinlichkeit, 106
Better–Than–Nothing Security, 54
Blackhole–Angriff, 20, 33, 149, 152
Bootstrapping, 71, 73–75, 123, 132–133, 149, 150, 153, 174

Chord, 28, 62, 78
CloudVPN, 60
Cryptographically Generated Addresses, 54–55

DHT, 28–29, 75, 78
Discovery, 72, 76–84, 134–135
DNS, 47–48, 74, 193
DoS–Angriff, *siehe* Sabotage
DoS–Recovery, 41, 157
DTLS, 25, 128
Dynamic Multipoint VPN, 22, 51–53

Easy VPN, 42–43
Eclipse–Angriff, 33, 154
Effizienz, 37, 77, 98, 147, 187
Ende–zu–Ende–Sicherung, 37, 92, 151–155, 157

Forwarding, *siehe* Paketweiterleitung

Graceful Degradation, 41, 157
Grayhole–Angriff, 20, 33, 97, 98, 122, 149, 152
Group Encrypted Transport VPN, 43–46

Hamachi^2, 46
Hole Punching, 25, 140

IKE, 25, 123, 134, 137, 138, 141, 160, 195–196
  –Mediation, 140, 145, 154
Infrastructure Hiding, 38, 155
Integrität, 19, 38, 71, 155
IPsec, 25–27, 128–129, 132–133
  BEET–Modus, 26, 136, 140
  Transportmodus, 25–26, 132, 134, 136
  Tunnelmodus, 26, 141
IPv6, 194

Konfigurationsdienste, 193
Kontrollierter Zugang, 39, 155

L2TP, 27
Latenz, *siehe* Verzögerungszeit
LDAP, 74
Linux, 129

Man–in–the–Middle–Angriff, 20, 124, 150, 155
MOBIKE, 193
Mobilität, 37, 77, 147, 193
Monotones Routing, 119–120
MTU, 91, 145

Multi–Level–Sicherheitssystem, 113
Multicast, 36, 74, 100–103, 139, 144, 145
Multipath–Routing, 122, 141, 192

N2N, 49–51
Namensauflösung, 193
NAT, 25, 26, 36, 132–133, 140, 144
Netz
  rotes, 21
  schwarzes, 21, 132
Netzwerkkoordinatensystem, *siehe* Vivaldi
Netzwerkpartitionierung, 37, 71, 96–98, 146, 174–177
Nichtabstreitbarkeit, 19
NTP, 20, 193

OMNeT++, 128, 186
Opportunistic Encryption, 54

P2PVPN, 58–59
Paketverzögerung, 20, 149, 152
Paketweiterleitung, 72, 90–92
Paketwiedereinspielung, 20, 132, 149, 152
Pastry, 28, 78, 163
Peer–to–Peer–Systeme, 27–31
PFS, 38, 154
PMIDP, 56–57
PPTP, 27

QoS, 45, 141

Remote–Zugriffsszenario, 21
Road Warrior, 21
Robustheit, 37, 146, 186
Routing, 72, 85–90

Sabotage, 13, 20, 192
  –resistenz, 40–41, 156–157, 179–186
Security Policy Protocol, 56
Selbstkonfiguration, *siehe* Autokonfiguration
Sicherheitsbeziehung
  indirekte, 23
Sicherheitsziele, 19–20, 37–41
Single Point of Failure, 22, 40
Sinkhole–Angriff, 33, 73, 153, 154, 169, 191–192
Skalierbarkeit, 37, 145, 186
Skip–Graphen, 30, 79, 166
SkipNet, *siehe* Skip–Graphen
Smartcard, 40, 73, 155, 195
Social VPN, 48–49
Stackelbergspiel, 112
Sybil–Angriff, 33, 154
Symphony, 28, 163

tinc, 59
TLS, 24–25, 123
Topologiekontrolle, 72, 93–100, 124–126, 135
Tunnel Endpoint Discovery, 55–56

Verfügbarkeit, 19, 39, 156–157
Verfügbarkeitszone, 113–114, 123–126
Verkehrsflussanalyse, 20, 72, 73, 92, 148, 151, 154, 192
Vertex–Cover, 110
Vertraulichkeit, 19, 37, 38, 71, 154
Verzögerungszeit, 85, 196
Vivaldi, 32, 93, 99, 124, 153, 177–178, 183, 191
VPN
  –Gateway, 21
  –Knoten, 21
  –Konzentrator, 21
  –Laborszenario, 158
  –Protokolle, 24–27
  –Struktur, 69
  –Szenarien, 21–24
  –Szenario, direktes, 158
  dezentrale, 22, 47–53
  Graphenmodell, 105
  HOT–Szenario, 159
  Hub–and–Spoke, 22, 42, 111, 189
  Schichtenmodell, 69
  Site–to–Site, 21–22, 25
  taktische, 23
  verschachtelte, 23
  verteilte, 53–60
  vollvermaschte, 22, 111, 189

WASTE, 57–58
Wippien, 48

Zeitsynchronisation, 193

Zero–Knowledge–Authentisierung, 124
Zurechenbarkeit, 19