RWTH Aachen
Lehrstuhl für Informatik 4
Informatikzentrum
Ahornstraße 55

52074 Aachen

Diplomarbeit

Sicherheit von Voice-over-IP-Systemen

Thomas Skora

Aachen, 15.02.2006

Diplomarbeit

Zur Erlangung des akademischen Grades

Diplom-Informatiker

an der

Rheinisch-Westfälischen Technischen Hochschule Aachen

Lehrstuhl für Informatik 4
Prof. Dr. rer. nat. Otto Spaniol

Thema:	Sicherheit von VoIP-Systemen Security of VoIP-Systems
Diplomand:	Thomas Skora Hardenbergstraße 84 47799 Krefeld Matrikelnummer: 227742
1. Prüfer:	Prof. Dr. rer. nat. Otto Spaniol RWTH Aachen
2. Prüfer:	Prof. Dr.-Ing. Manfred Nagl RWTH Aachen
Abgabedatum:	17. Februar 2006

Danksagung

An dieser Stelle möchte ich mich bei allen Bedanken, die mir diese Arbeit ermöglicht haben, mir während der Bearbeitung hilfreich zur Seite standen und mich durch das Informatikstudium begleitet haben.

An erster Stelle möchte ich meiner Mutter und meinen Großeltern danken, die mir das Studium ermöglicht haben und mich während dieser Zeit moralisch und mit viel Liebe unterstützt haben.

Während des Studiums wurde ich von von vielen Kommilitonen begleitet. Besonders erwähnen möchte ich hier Volker, Daniel und Stefan, mit denen ich viele Aufgaben erfolgreich gelöst habe. Für die angenehme Zeit, die ich mit ihnen verbracht habe möchte ich mich bedanken.

Bei Professor Otto Spaniol möchte ich mich für die Betreuung und bei Professor Manfred Nagl für dessen Arbeit als Zweitprüfer bedanken

Diese Arbeit wurde von Dr. Dogan Kesdogan betreut, der mich stets mit hilfreichen Tipps und Anregungen unterstützte, dafür sei ihm vielmals gedankt. Erwähnen möchte ich an dieser Stelle auch Lexi Pimenidis, der mir mit zahlreichen Ideen und interessanten Diskussionen sehr weitergeholfen hat. Ihm sei dafür gedankt.

Bei Beatrice Haagen möchte ich mich für das Korrekturlesen dieser Arbeit und ihre moralische Unterstützung bedanken.

Ein Dank geht auch an alle Freunde, die mich außerhalb des Studiums begleiteten und immer für mich da waren.

Krefeld, 15. Februar 2006
Thomas Skora

Ehrenwörtliche Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, sind als solche gekennzeichnet.

Krefeld, 15. Februar 2006 Thomas Skora

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einleitung

In den letzten Jahren machte Voice-over-IP die Entwicklung vom technischen Spielzeug zum ernsthaften Konkurrenten der klassischen leitungsverbundenen Telefonie. Die ersten Versuche der Sprachübertragung über das Internet erfolgten im Jahr 1995 durch das Unternehmen Vocaltec, damals noch in mäßiger Qualität und im Halb-Duplex-Betrieb. 1996 erfolgte die erste Normierung der H.323-Protokollfamilie durch die ITU-T, die unter anderem von Microsofts Netmeeting verwendet wurde. Im gleichen Jahr wurde die erste Version des *Real-Time Transport Protocols* (RTP) [SCFJ96], welches zum Transport von Mediendaten insbesondere Sprache verwendet wird, von der Internet Engineering Task Force (IETF) standardisiert und im Jahr 2003 aktualisiert [SCFJ03]. 1999 folgte das *Session Initiation Protocol* (SIP) [HSSR99], das zur Signalisierung verwendet wird, 2002 in der zweiten Version standardisiert wurde [HSSR02] und derzeitig von einem Großteil der Anbieter von VoIP-Diensten verwendet wird.

Waren die ersten Schritte der Internet-Telefonie kommerziell noch unbedeutend, änderte sich das in den vergangenen Jahren grundlegend. Im ersten Quartal des Jahres 2005 wuchs der weltweite Umsatz für VoIP-Produkte um 40% auf 493 Millionen US-Dollar im Verhältnis zum gleichen Quartal des Vorjahres. Für das Jahr 2008 wird sogar ein Jahresumsatzvolumen von 5,8 Milliarden US-Dollar vorausgesagt [KM05].

Als Hauptvorteil für Voice-over-IP geben Unternehmen die Kostenersparnis an. Neben den niedrigeren Verbindungskosten, insbesondere wenn beide Seiten über IP-Netze miteinander kommunizieren, ist es günstiger nur ein Netz, das sowohl Daten als auch Telefonie transportiert, aufzubauen und zu warten. Unternehmen mit mehreren Standorten können so vorhandene IP-Leitungen zum Führen von internen Gesprächen verwenden. Ein weiterer Vorteil ist die Mobilität von Voice-over-IP. Ein Benutzer behält unabhängig vom aktuellen Standort seine Rufnummer und bleibt erreichbar.

Im Vergleich zur traditionellen Telefonie, in der Anrufe über ein separates Netz geführt werden, bietet der Transport von Telefonie und Daten über gemeinsame Leitungen jedoch Angriffspunkte. Das Abhören oder Stören von Telefongesprächen, die über ein PSTN[1] geführt werden, benötigt physischen Zugriff auf Leitungen und Knotenpunkte. Es wird spezielles Know-How benötigt, um die verwendeten Protokolle wie das zur Signalisierung verwendete SS7 und proprietäre Protokolle zu decodieren. Bei VoIP werden dagegen vorwiegend standardisierte und offene Protokolle verwendet, die über IP-Netze transportiert werden. Hier stehen einem Angreifer bereits eine Menge wohlbekannter

[1] Public Switched Telephone Network (öffentliches Telefonnetzwerk), wird auch Plain Old Telephone Service (POTS) genannt.

Programme zur Verfügung, mit denen es möglich ist, Datenverkehr mitzuschneiden und die verwendeten Protokolle zu decodieren. Unsicher implementierte Geräte genießen in einem IP-Netzwerk nicht mehr den Schutz eines vertrauenswürdigen Netzes und können bei vorhandenen Exploits, die eine Sicherheitslücke ausnutzen, sogar von Laien angegriffen werden. Um auf fremde Kosten zu telefonieren, werden aufgrund der Mobilität von VoIP nur Zugangsdaten zum IP-Telefonieanbieter benötigt, die z.B. durch Phishing-Angriffe oder das Entwenden von konfigurierten Endgeräten erlangt werden können. Im Gegensatz dazu muss beim PSTN ein physischer Zugriff zu einem Anschluss erfolgen.

1.1 Problemstellung und Zielsetzung der Arbeit

Bisherigen Arbeiten zum Thema „Sicherheit in Voice-over-IP“ behandeln die Thematik meist aus einem administrativen Standpunkt und abstrahieren von technischen Details der zugrunde liegenden Protokolle. Ein Ziel dieser Arbeit ist es, die technischen Details der Angriffe zu behandeln und sie zu implementieren. Sicherheitsmaßnahmen wurden bisher anhand der Abdeckung der drei Schutzziele Verfügbarkeit, Vertraulichkeit und Integrität untersucht. Aufbauend auf den Kenntnissen über die technischen Details der Angriffe sollen diese Maßnahmen zusätzlich anhand der Absicherung gegenüber spezifischen Angriffen beurteilt werden.

In vorliegender Arbeit wird ein relativ neuer Ansatz zur Sicherheitsbewertung, die Attack Trees [Sch99], auf VoIP angewendet. Dieser Ansatz erlaubt es, systematisch die Kosten für einen erfolgreichen Angriff zu berechnen. Da konkrete monetäre Kosten für jede Umgebung individuell berechnet werden müssen, kann das nicht in einer allgemein gehaltenen Arbeit wie dieser geschehen. Attack Trees können jedoch als Teil einer größeren Analyse, in der VoIP nur eine von mehreren angreifbaren Komponenten ist, wiederverwendet und zur Berechnung mit individuellen Kostenmaßen eingesetzt werden. Um Anhaltspunkte für die Schwierigkeit von Angriffen zu geben, wird in Abschnitt 8.1 ein Kostenmaß definiert, das eine solche Bewertung anhand der Schwierigkeit eines Angriffs erlaubt. Zusätzlich wurde der Formalismus der Attack Trees um eine Komponente erweitert, die eine Kosten-Nutzen-Analyse für die Absicherung des untersuchten Systems erlaubt. Ab Abschnitt 8.2 werden die Attack Trees für die drei Schutzziele Verfügbarkeit, Vertraulichkeit und Integrität ausgearbeitet, beschrieben und in Abschnitt 8.5 bewertet.

1.2 Aufbau der Arbeit

In Kapitel 2 werden andere Veröffentlichungen zum Thema VoIP-Sicherheit und verschiedene Richtungen der Forschung auf diesem Gebiet vorgestellt. Außerdem wird eine Abgrenzung und Einordnung dieser Arbeit vorgenommen. Kapitel 3 behandelt die Grundlagen von VoIP, die für das weitere Verständnis der Arbeit notwendig sind. Unter anderem wird der allgemeine Aufbau eines VoIP-Systems mit dessen Komponenten

und die hier behandelten Protokolle SIP, SDP, RTP, SRTP und Skype beschrieben. In Kapitel 4 werden Angriffe auf das Signalisierungsprotokoll SIP vorgestellt, die in dessen ungesicherten Form durchführbar sind. Kapitel 5 widmet sich den Protokollen und Erweiterungen bestehender Protokolle, die der Absicherung von VoIP dienen. In Kapitel 6 folgt eine kurze Einführung in der mit der Definition der Schutzziele und Angreifermodelle die Grundlagen der Sicherheit behandelt werden. In Kapitel 7 wird eine Bewertung der in Kapitel 5 vorgestellten Sicherheitsmaßnahmen vorgenommen, indem die Erfüllung der Schutzziele durch die einzelnen Protokolle bewertet wird. Den zweiten Teil der Sicherheitsanalyse bildet die Ausarbeitung von Attack Trees in Kapitel 8, die eine strukturierte Bewertung der Gesamtsicherheit eines Systems ermöglichen. Neben den Angriffen aus Kapitel 4 werden nicht-VoIP-spezifische Angriffe in die Attack Trees aufgenommen, um ein Gesamtbild der Gefährdung eines VoIP-Systems zu erhalten. Die Attack Trees werden anschließend mit einem Kostenmaß bewertet, das die Schwierigkeit eines Angriffs beurteilt. Kapitel 9 schließt die Arbeit mit der Zusammenfassung der Ergebnisse und einem Ausblick auf die weitere Entwicklung ab.

2 Verwandte Arbeiten

Die Sicherheit von VoIP ist eine umfangreiche Thematik, die neben den VoIP-Protokollen und Komponenten auf die zugrunde liegende IP-Infrastruktur angewiesen ist. Weitere Themen, die in diesem Zusammenhang genannt werden, sind Identitätsdiebstahl, Gebührenbetrug, Lawful Interception[1] und Spam.

Eine Einführung in die Thematik gibt [SL04]. Diese Arbeit beschreibt den Übergang vom PSTN zu IP-basierten Netzen und damit den Wechsel von einem vertrauenswürdigem Netz, das von einer zentralen Instanz kontrolliert wird und das speziell auf die angebotenen Dienste zugeschnitten ist, zu einem Netz, in dem prinzipiell jeder Telefoniedienste anbieten kann und das potentiellen Angreifern einen leichteren Zugang bietet als die traditionellen Telefonnetze. Eine Gegenüberstellung der Angreifbarkeit der PSTN und IP-basierten Telefonie wird in [Kle03] durchgeführt.

In [ASRS01, Aki02a, Aki02c, Col05, See04] werden Gefahren für den VoIP-Betrieb zusammengefasst. Ein Angriff, bei dem der Sprachdatenstrom mit einem speziellen Packet-Sniffer aufgezeichnet wird, wird in [Lon02] beschrieben. Das gleiche Ziel verfolgt das Projekt Vomit [vom], das die Medienströme aus einem Tcpdump-Mitschnitt in Audiodateien speichert. In [Iac02] geht der Autor des Artikels einen Schritt weiter und leitet die RTP-Pakete an einen beliebigen Host weiter, der sie aufzeichnen kann. Ein weiterer Angriff dieser Art, der einen Zusammenhang zwischen Angriffen auf die IP-Infrastruktur und dem darauf aufbauenden VoIP-System herstellt, wird in [ACB+05] präsentiert. Hier wird der Medienstrom mit Hilfe eines Angriffs auf das ARP-Protokoll an den Angreifer umgeleitet und kann so von ihm abgehört werden. Auch Angriffe auf die IP-Stacks der Endgeräte werden in Betracht gezogen [GS]. Protokolle, wie das Trivial File Transfer Protocol (TFTP), das aufgrund seines einfachen Aufbaus in vielen Embedded Devices wie VoIP-Telefonen zum Download von Firmware-Images und Konfigurationen verwendet wird, sorgen für zusätzliche Schwachstellen [Aki02b]. Weitere Arbeiten, die sich mit Gefahren auseinandersetzen und Lösungen präsentieren, sind [Cis02, Tuc04, PS05, Wei01]. Eine strukturierte Übersicht, welche Angriffe welche Schutzziele verletzen, findet sich in [Osw05, NQBS]. In vielen Arbeiten werden Schwächen der Protokollparser als Angriffspunkt genannt, was durch Versuche, die im Rahmen dieser Arbeit durchgeführt wurden, bestätigt werden konnte. Um solche Sicherheitslücken automatisiert finden zu können, wurden mehrere Programme entwickelt, die in [AHM+05] verglichen werden.

[1] Wird oft auch mit der Abkürzung CALEA (Communications Assistance for Law Enforcement Act) bezeichnet

Umfassend wird VoIP-Sicherheit in [KWF05, AAGea05, RR05] behandelt. Diese Veröffentlichungen behandeln zunächst die Grundlagen von VoIP und den dazugehörigen Kommunikationsprotokollen, deren Sicherheitsprobleme und Sicherheitsmaßnahmen, die diese Probleme lösen. In [AAGea05] wird außerdem auf den rechtlichen Aspekt von Telefoniediensten eingegangen und hervorgehoben, dass das Fernmeldegeheimnis und der Datenschutz auch für VoIP gelten. Des Weiteren wird bewertet, welche Sicherheitsmaßnahmen welche Schutzziele abdecken. Andere Arbeiten, die sich jedoch nur mit der Absicherung von VoIP beschäftigen und die ausführliche Benennung der Gefahren auslassen, sind [Def04, SKS04, MCA99]. [VoI05, Rob04] behandeln dagegen nur Gefahren für den VoIP-Betrieb.

In minisip[2] wurden erstmals Sicherheitsfeatures wie SRTP [Cab03] und MIKEY in ein freies SIP-Softphone implementiert, dessen Entwicklung weitgehend im Rahmen von Diplomarbeiten und Dissertationen stattfand. In [Bil03] werden dafür allgemeine und VoIP-spezifische Anforderungen an ein Schüsseltauschprotokoll untersucht und MIKEY in minisip implementiert. Ein kryptographisches Protokoll beinhaltet neben aufwändigen Berechnungen zusätzliche Daten, die übertragen werden müssen. Beim Rufaufbau können bereits relativ kurze Wartezeiten störend wirken, insbesondere wenn das sogenannte Clipping auftritt. Clipping bedeutet, dass der Medienstrom zwar schon empfangen wird, vom Empfänger aber noch nicht decodiert werden kann, weil notwendige Berechnungen ausstehen. Aus diesem Grund sollten die Verzögerungen, die durch den Einsatz zusätzlicher Protokolle zustande kommen, so niedrig wie möglich sein. In [BEOV05] wurden die durch MIKEY verursachten Verzögerungen für SRTP dargestellt. Im Ergebnis waren die Verzögerungen bis zum Klingeln des Telefons und der Antwortphase beim Diffie-Hellman-Schlüsseltausch deutlich länger, jedoch im vertretbaren Bereich. In [Orr05] wurde der Fokus auf die Verwendung von IPsec zur Übertragung der Medienströme gelegt. In der Antwortphase sind die Verzögerungen bei IPsec um ein Vielfaches höher als bei SRTP, was zu der Folgerung führt, dass SRTP IPsec vorzuziehen ist. Das wird in [AAGea05] ebenfalls empfohlen. Weiter werden Optionen aufgezeigt, die das Clipping durch den Transport von MIKEY-Nachrichten in anderen SIP-Paketen als der initialen Invite-Anfrage des Anrufers und dem ersten zuverlässig übertragenen Antwortpaket des Angerufenen mindern können. Mit dem mobilen Einsatz von VoIP in WLANs beschäftigt sich [Vat05]. Der mobile Einsatz unterscheidet sich dadurch, dass ein mobiles Telefon das Netz und damit die IP-Adresse wechseln kann. In einem WLAN führt ein solcher Handover zu einer kurzen Unterbrechung des Paketflusses, weil Sicherheitsparameter zwischen Mobilteil und neuem Access-Point ausgehandelt werden müssen und auf Anwendungsebene die neue Netzwerkadresse signalisiert werden muss. Eine Arbeit, die sich auch mit der Implementierung eines Sicherheitsfeatures in ein bestehendes Produkt beschäftigt ist [Tha01]. Es wird eine Authentifizierung zwischen zwei Gatekeepern in einem H.323-System entwickelt. Des Weiteren enthält die Arbeit eine Analyse der Si-

[2] `http://www.minisip.org`

cherheitsprobleme von H.323. In [Kul05] beschäftigt sich der Autor mit der Absicherung von Textübertragungen, die durch SIP signalisiert werden und behandelt dabei unter anderem auch die Protokolle SRTP, MIKEY, TLS und IPsec.

Eine andere Arbeit, die sich mit dem Einfluss der Verschlüsselung auf die Dienstgüte von VoIP beschäftigt, behandelt IPsec [BBR02]. Als Nachteil wurde festgestellt, dass der Header eines RTP-Pakets mit 40 Bytes Payload und UDP- und IP-Header bei einer Verschlüsselung mit DES und ohne Absicherung der Integrität mittels eines Hashwertes auf 122 Byte anwachsen lässt, was 52,5% entspricht. Daraus folgt, dass die Verzögerung eines verschlüsselten Pakets bei steigender Sättigung der Netzwerkleitung bereits deutlich früher ansteigt und insgesamt weniger IPsec-Pakete zu einem Zeitpunkt gesendet werden können. Der Vergrößerung der IPsec-Pakete kann durch die ebenfalls in [BBR02] beschriebene IPsec-Headerkompression, die auf der RTP-Headerkompression[CJ99] basiert, entgegengewirkt werden. Bei der Headerkompression wird die Tatsache genutzt, dass sich die im IPsec-Paket gekapselten Protokollheader für einzelne Verbindungen nur selten ändern. Die Kompressionsroutine verwaltet diese Header intern, identifiziert einzelne Verbindungen mit einem Session-Identifier und überträgt nur einen Teil der Header. Bei einem Paketverlust kann unter Umständen die Synchronisation zwischen dem Zustand der Kompressionroutinen und den realen Gegebenheiten verloren gehen. In so einem Fall muss zur Resynchronisation ein unkomprimiertes Paket mit etwas Overhead gesendet werden. Messungen ergeben, dass auch bei einer Fehlerrate von 10% kaum mehr Bandbreite beansprucht wird als bei einer fehlerfreien Übertragung. Als Engpass bei IPsec wurden die kryptographischen Berechnungen ausgemacht. Insbesondere deswegen, weil sie keine Priorisierung von Paketen anbieten. Ein weiterer Faktor bei der Leistungsfähigfähigkeit ist die Größe der verschlüsselten Pakete. Werden, wie bei VoIP üblich, viele kleine Pakete übertragen, erreicht die IPsec-Implementierung nur einen Teil des Durchsatzes von Übertragungen mit großen Paketen.

Neben der Verschlüsselung des Medienstroms bietet SRTP die Absicherung der Integrität der RTP-Pakete an. Der Hashwert, der an ein SRTP-Paket angehangen wird, erzeugt allerdings einen Overhead, der in Relation zur Paketgröße groß ist und dadurch weitere Verzögerungen beim Transport verursacht. Durch digitale Wasserzeichen in Audiodaten [YH04] kann die Integrität ohne zusätzlichen Overhead in der Paketgröße gesichert werden. Nützlich für die Absicherung ist das sogenannte Fragile Watermarking [Cve04], bei dem der eingebettete Watermark bewusst keine Robustheit gegen Manipulationen aufweist. In [MK06] wird die Integritätsabsicherung auf die Signalisierung ausgeweitet, indem die Signalisierungspakete in die Berechnung des Watermarks für RTP-Pakete mit einfließen. Ein Nachteil dieses Verfahrens ist jedoch, dass die Signalisierungspakete erst im Nachhinein geprüft werden und deswegen kein Schutz vor DoS-Angriffen geboten wird, bei denen kein Medienstrom zustande kommt. Ein weiteres Einsatzgebiet von Watermarks ist die Intrusion Detection [SSN+02]. Im Gegensatz zu den vorherigen Einsatzgebieten werden robuste Watermarks eingesetzt, anhand deren Inhalt und Existenz Medienströme an Firewalls oder Session Border Controllers erlaubt

oder geblockt werden können.

Auf dem Gebiet der Lawful Interception [MSR+03b, MSR+03a, li1, li2] kann Watermarking ebenfalls verwendet werden, um bei Direct-IP-Calls, deren Ziel durch Anonymisierungsnetzwerke verschleiert wird, beide Endpunkte zu identifizieren. In [WCJ05] wird die Tatsache ausgenutzt, dass der zeitliche Abstand der Pakete des Medienstroms bis auf kleine durch das Netzwerk oder Systemauslastung bedingte Variationen gleich ist. Die Verzögerungen der Pakete des Medienstroms werden so variiert, dass sie am anderen Endpunkt wiedererkannt werden. Insgesamt kann Lawful Interception nicht mehr in dem Umfang wie in einem PSTN durchgeführt werden, weil bei VoIP eine Ende-zu-Ende-Absicherung eines Telefonats möglich ist.

Mit der weiteren Verbreitung von VoIP kam Bedarf auf, auch Notrufdienste ordnungsgemäß in den VoIP-Betrieb einzubinden. Anforderungen an Vertraulichkeit, Verfügbarkeit, Integrität und Authentizität werden in [CM05a, CM05b] betrachtet und für VoIP bewertet. Der Aufbau der Infrastruktur stellt weitere Anforderungen [MHRSW05] wie die Möglichkeit zur geographischen Lokalisierung des Anrufers und dem anschließenden Routing zur zuständigen Dienststelle. Das Problem der Lokalisierung soll durch zusätzliche DHCP-Informationen, einen GPS-Empfänger oder bei mobilem Einsatz durch Anpeilen des Senders ermittelt werden.

Eine der wichtigsten Anforderungen für Notrufe ist die Verfügbarkeit, die in [JS03] mit

$$\text{Verfügbarkeit} = \frac{\text{Anzahl der erfolgreichen Anrufe}}{\text{Anzahl der Erstanrufversuche}}$$

definiert wird. Laut [JS03] garantieren Betreiber von PSTNs eine Verfügbarkeit von über 99,9%. Im konkreten Fall gibt das US-Telekommunikationsunternehmen AT&T für 1997 eine Verfügbarkeit von 99,98% an, was bedeutet, dass das Festnetz in diesem Jahr etwa eine Stunde und 45 Minuten nicht verfügbar war. Messungen für VoIP über einen Zeitraum von ca. 2 Monaten ergaben eine Verfügbarkeit von 99,53%, was auf ein Jahr hochgerechnet einer Gesamtausfallzeit von einem Tag und 17 Stunden entspricht. Da in PSTNs nach einem erfolgten Rufaufbau im Normalfall eine qualitativ gute Verbindung besteht, wird als Maß für die Verfügbarkeit das Verhältnis der erfolgreich hergestellten Verbindungen als ausreichend angesehen. Bei VoIP ist dieses Maß jedoch unzureichend, weil bei IP-Netzen der Paketverlust eine wichtige Rolle spielt. Geht man davon aus, dass mehr als 5% Paketverlust nicht mehr ausreichend für eine Sprachverbindung sind, sinkt die Verfügbarkeit in den Messungen auf 97,48%, was über ein Jahr summiert bereits über 9 Tage an Ausfallzeiten bedeutet. Da die Verfügbarkeit von Ende zu Ende (A_{e2e}) neben der Verfügbarkeit des Telefonnetzes (A_{net}) auch von lokalen Komponenten wie dem Endgerät (A_{h_i}) und einer Telefonanlage (A_{l_i}) abhängt, berechnet sich die Gesamtverfügbarkeit für einen Signalisierungspfad unter der Voraussetzung, dass die Verfügbarkeiten der Einzelsysteme stochastisch unabhängig voneinander sind mit

$$A_{e2e} = A_{h_1} \cdot A_{l_i} \cdot A_{net} \cdot A_{l_2} \cdot A_{h_2}$$

Eine Übersicht mit Lösungsansätzen über das Thema Spam findet sich in [RJP05]. Als einer der wichtigsten Gründe für die Attraktivität von VoIP-Spam, der auch Spit[3] genannt wird, werden die deutlich niedrigeren Kosten für Anrufe genannt. IP-Leitungen gibt es zu monatlichen Pauschalpreisen, wodurch es für einen Spammer unerheblich ist, wie viele Anrufe er absetzt. Insbesondere internationale Werbeanrufe, die bisher durch hohe Verbindungskosten unattraktiv waren, werden durch VoIP so stark gesenkt, dass auch Telefon-Spam aus dem Ausland ein Problem darstellen wird. Die Erkennung solcher unerwünschten Anrufe stellt sich im Gegensatz zum E-Mail-Spam als schwierig heraus. Die Untersuchung des Gesprächsinhaltes führt nicht zum Ziel, weil das Gespräch zu diesem Zeitpunkt bereits angenommen wurde. Signalisierungspakete bieten nur wenige Merkmale, anhand denen Spam erkannt werden könnte. Sowohl das Konzept der Black- als auch der Whitelists ist nur eingeschränkt praxistauglich, weil Identitäten speziell bei Direct-IP-Calls leicht zu fälschen und im Normalfall auch unbekannte Anrufer erwünscht sind. Auch die anderen in [RJP05] vorgestellten Techniken gegen Spam sind meist unpraktikabel, weil sie die Nutzung der Telefonie erschweren. Praktikabel ist der Ansatz, Anrufe nur noch von einem Proxy des Anbieters entgegenzunehmen, der in der Lage ist, massive Anrufversuche zu erkennen und zu filtern. Ein weiterer Lösungsansatz wäre die domainübergreifende Identifizierung von Anrufern [JP05].

Diese Arbeit verfolgt, ähnlich wie andere allgemeine Arbeiten zum Thema VoIP-Sicherheit, das Ziel, einen Überblick über die verwendeten Techniken, Angriffe und Sicherungsmaßnahmen mit einer Spezialisierung auf das Signalisierungsprotokoll SIP zu geben. Dabei soll unter anderem ähnlich der in [AAGea05] vorgenommenen Bewertung für Sicherheitsmaßnahmen nach Schutzzielen eine Bewertung der Sicherheitsprotokolle aus Kapitel 5 vorgenommen werden, in der zusätzlich die Qualität der Absicherung mit einfließt.

[3] Spam over Internet Telephony

3 Grundlagen

Voice-over-IP beinhaltet unterschiedliche Protokolle. Zur Signalisierung wird hauptsächlich das Session Initiation Protocol (SIP) [HSSR02] oder Q.391 bzw. H.245, welche zur H.323-Protokollfamilie gehören, verwendet. Diese Protokolle kontrollieren Vorgänge wie die Vermittlung von Gesprächen, den Rufaufbau und -abbau sowie die Aushandlung von Parametern, z. B. den verwendeten Codec, die Bitrate oder die maximal zulässige Bandbreite. Für diese Zwecke wird zusätzlich das *Session Description Protocol* (SDP) [HJ98] verwendet, um die Eigenschaften des Mediendatenstroms festzulegen.

Für die Sprachübtertragung wird das *Real-time Transport Protocol* (RTP) [SCFJ03] verwendet, das dem Empfänger der Pakete trotz unzuverlässiger Übertragung über UDP [Pos80] ermöglicht, einen Paketverlust festzustellen und die ursprüngliche Reihenfolge der Daten wiederherzustellen. Zusätzlich zu den Mediendaten können noch Kontrollinformationen wie die Anzahl der velorenen Pakete oder der gemessene Jitter der RTP-Verbindung mit Hilfe des *RTP Control Protocols* (RTCP) übertragen und für eine dynamische Wahl der Übertragungsparameter verwendet werden. Ein weiteres Protokoll, das beide Aufgaben der Signalisierung und Sprachübertragung in sich vereint und diese über eine einzige Verbindung überträgt, ist das InterAsterisk-Exchange-Protokoll [SM05]. Um die Funktion der Gateways zu kontrollieren, existieren spezielle Protokolle wie das *Media Gateway Control Protocol* (MGCP) [AEHP99] oder *Megaco*, welches aktuell in der Version 1.0 vorliegt [CGR$^+$00].

Neben den bisher genannten Protokollen gibt es noch weitere, die nicht direkt mit der Hauptfunktionalität von Voice-over-IP verbunden sind. Eines davon ist STUN (*Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*) [HRWM00]. Mit Hilfe dieses Protokolls wird es einem Client, der sich hinter einem NAT-Gateway befindet, ermöglicht, seine von außen sichtbare IP-Adresse zu ermitteln und den verwendeten NAT-Typ zu identifizieren. Dies ist notwendig, da die Kontaktinformationen auf Anwendungsschicht in Protokollen wie SIP und SDP übertragen werden und der Client ohne Kenntnis seiner nach außen hin sichtbaren IP-Adresse falsche Informationen liefern würde. Weitere Protokolle, die ähnliche Ziele verfolgen, sind *Traversal Using Relay NAT* (TURN) [HRM03] und *Realm Specific IP* (RSIP) [BL01, BLGT01].

TURN verfolgt hierbei das Konzept, dass ein Relay zwischen den beiden NAT-Gateways die Vermittlung der Verbindungen übernimmt. Ein Client bekommt von einem TURN-Server ein öffentlich erreichbares IP-Adressen/Port-Paar zugewiesen und leitet alle an diese öffentliche Adresse ankommenden Pakete an den Client weiter. Da der Client die

öffentliche Adresse erfährt, kann er diese für seine Zwecke verwenden.

RSIP stellt sich dagegen als Alternative zu NAT dar. Hierbei fragt der Client beim RSIP-Gateway, welches das klassische NAT-Gateway ersetzt, explizit eine öffentliche IP-Adresse an und kann diese dann zur Kommunikation verwenden.

In diesem Kapitel wird zunächst der allgemeine Aufbau eines VoIP-Systems beschrieben und anschließend die Protokolle SIP/SDP, RTP und SRTP behandelt. Die Entscheidung wurde aus dem Grund gefällt, weil derzeit alle großen Anbieter am deutschen VoIP-Markt SIP einsetzen. Es ist davon auszugehen, dass SIP die ältere H.323-Protokollfamilie langfristig verdrängen wird. Ein anderes proprietäres Protokoll ist das des VoIP-Anbieters Skype. In Abschnitt 3.4 wurden alle aktuell verfügbaren Informationen zu Skype zusammengefasst, um einen Überblick über die Funktionsweise des Protokolls zu geben. Da das Protokoll nicht offen liegt und sich ein Reverse-Engineering der Skype-Software schwierig gestaltet bleiben allerdings viele Details offen, deren Erfassung außerhalb des Aufgabenbereichs dieser Arbeit liegt.

3.1 Aufbau eines VoIP-Systems

Prinzipiell kann VoIP bereits mit einer funktionierenden IP-Infrastruktur und IP-Telefonen oder Softphones, die auf einem Rechner ausgeführt werden, funktionieren. Um in so einer Konfiguration die sogenannten Direct-IP-Calls zu führen, müssen die einzelnen Teilnehmer die IP-Adressen der gewünschten Gesprächspartner kennen. Ein solcher Aufbau ist jedoch unflexibel. Bei einer Adressänderung eines Endgeräts, muss dies allen anderen Teilnehmern mitgeteilt und Adressbucheinträge entsprechend abgeändert werden werden. Auch das Hinzufügen weiterer Teilnehmer ist mit diesem Aufwand verbunden.

In der IP-Telefonie mit dem SIP-Protokoll übernehmen Proxys die Aufgabe der Vermittlungsstelle. Ein Proxy kann einen Location Service enthalten, der ankommende SIP-Pakete an das Telefon des angerufenen Teilnehmers weiterleitet oder an einen anderen Proxy weiterleitet, falls der Benutzer außerhalb des eigenen Einflussbereichs liegt. Zusätzlich können Funktionen wie Konferenzen, Voice-Mail oder Anrufweiterleitung zentral verwaltet werden. Da der Rufaufbau im Normalfall über den Proxy erfolgen sollte, können in diesem auch Daten zur Zeiterfassung und Gebührenberechnung erfasst werden.

In einem bestimmten Umfeld kann es sinnvoll sein, mehrere Proxys zu verwenden. Zum Beispiel kann ein Proxy zur Abwicklung von internen Gesprächen verwendet werden und Telefonate ins Festnetz zu einem anderen Proxy weiterleiten. Weiterhin ist es sinnvoll, Proxys redundant zu betreiben, um bei einem Ausfall auf eine Backup-Lösung ausweichen zu können oder in Lastsituationen Load-Balancing anzubieten.

Da neben VoIP auch die klassische Telefonie existiert, wird mit zusätzlichen Komponenten, den Gateways, eine Verbindung zwischen beiden Telefoniesystemen hergestellt. In Kombination mit einem Gateway kann aus einem VoIP-System ein Least-Coast-Router konfiguriert werden, der den günstigsten Festnetz-Anbieter für den jeweils aktuel-

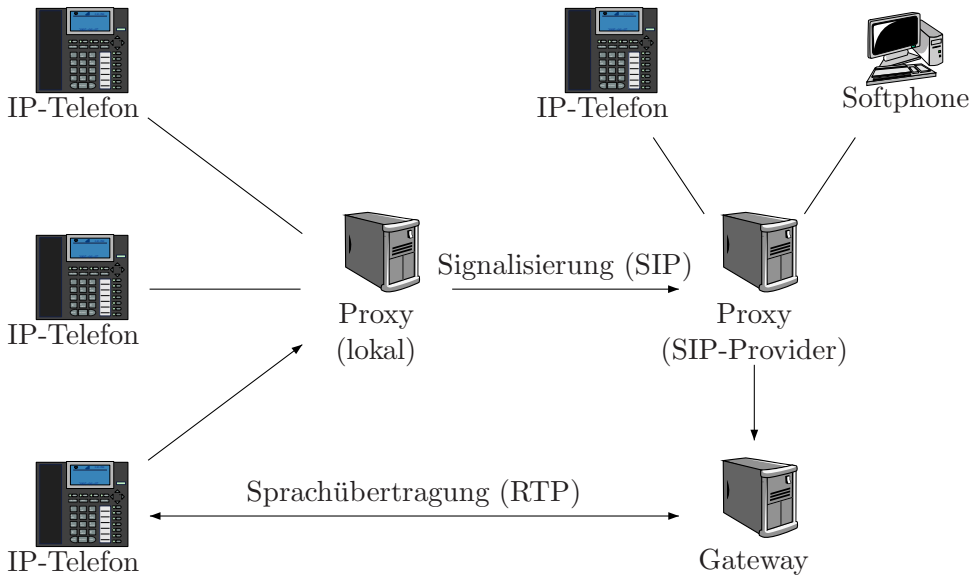


Abbildung 3.1: Aufbau eines VoIP-Systems

len Zeitpunkt auswählt. Proxys und Gateways werden häufig in einem Gerät kombiniert.

Eine nicht direkt für die Funktion von VoIP benötigte Komponente, die jedoch der Sicherheit zuträglich ist, ist ein Application-Level-Gateway (ALG), das den aus nicht vertrauenswürdigen Netzen ins interne Netz ankommenden VoIP-Datenverkehr auf Angriffsmuster untersucht und Pakete ggf. filtert. Bei der Vorstellung der Angriffe in dieser Arbeit werden Möglichkeiten erörtert, diese Angriffe anhand von Anomalien im Datenverkehr zu erkennen.

VoIP führt in Netzen mit NAT[1] häufig zu Problemen, da Kontaktdaten wie IP-Adressen und Ports bei der Signalisierung auf die Anwendungsschicht übertragen werden. NAT-Gateways, die nicht speziell für VoIP ausgelegt sind, schreiben nur IP-, TCP- und UDP-Header um, was dazu führt, dass die Kontaktdaten beim Verlassen des Netzes nicht mehr stimmen. Um solche Probleme zu umgehen, können Medien-Proxys eingesetzt werden. Diese nehmen die Medienströme der Clients entgegen und leiten sie an das richtige Ziel weiter. Ein weiteres Aufgabengebiet für Medien-Proxys ist das Konvertieren von Codecs, um eine Kommunikation von nicht zueinander kompatiblen Endgeräten zu ermöglichen.

[1] Network Adress Translation

Methode	Beschreibung
REGISTER	Anmelden eines Benutzers beim SIP-Server
INVITE	Signalisierung eines Anrufes
ACK	Bestätigung des Anrufenden zu einer Invite-Anfrage
CANCEL	Abbruch einer Invite-Anfrage
BYE	Beenden eines Anrufes
OPTIONS	Abfrage der Eigenschaften und Fähigkeiten.

Tabelle 3.1: SIP-Anfragemethoden

3.2 Session Initiation Protocol (SIP)

Das *Session Initiation Protocol* wurde zum ersten Mal im Jahre 1999 im RFC 2543 standardisiert und im Juni 2002 im RFC 3261 [HSSR02] zur Version 2.0 aktualisiert. Derzeit bietet der größte Teil der IP-Telefonie-Anbieter ausschließlich SIP als Signalisierungsprotokoll an und es scheint sich gegen die H.323-Protokollfamilie als Standard durchgesetzt zu haben.

3.2.1 Aufbau

SIP ist ein textbasiertes und damit auch leicht von Menschen lesbares Protokoll und ist stark an HTML angelehnt. Jedes SIP-Paket beginnt in der ersten Zeile mit einer Anfrage oder einem Antwortcode und ist folgendermaßen aufgebaut:

```
INVITE sip:thomas@skora.net SIP/2.0
```

An erster Stelle steht immer die Anfragemethode, die von einer URL gefolgt wird. In den meisten Fällen werden URLs des SIP-Schemas verwendet, deren Aufbau einer E-Mail-Adresse ähnelt. Abgeschlossen wird die Zeile mit der Angabe der Protokollkennung "SIP", die mit einem Schrägstrich von der Versionsnummer, aktuell 2.0, getrennt ist. Gängige SIP-Methoden, die im RFC 3261 [HSSR02] spezifiziert sind, sind in Tabelle 3.1 aufgelistet. Das Protokoll kann durch das Einführen neuer Anfragen beliebig erweitert werden. So wurde unter anderem eine Instant Messaging-Erweiterung in RFC 3428 [SRHG02] standardisiert, die auch im VoIP-Bereich verwendet wird.

Ein Antwortpaket auf eine SIP-Anfrage beginnt wie folgt:

```
SIP/2.0 200 OK
```

Am Zeilenanfang stehen die Protokollkennung und die Versionsnummer, der eine dreistellige numerische Antwortkennung mit einem beschreibenden Text folgt. Die Antwortcodes sind in mehrere Kategorien eingeteilt, die in Tabelle 3.2 zusammengefasst sind.

Nach der ersten Zeile folgt eine beliebige Anzahl von Headerzeilen. Wichtige Header sind:

Code	Beschreibung	Beispiele
1xx	Provisorische Antworten	100 Trying, 180 Ringing
2xx	Anfrage erfolgreich ausgeführt	200 OK
3xx	Umleitung	301 Moved Permanently, 302 Moved Temporarily
4xx	Fehler bei der Durchführung der Anfrage	401 Unauthorized, 404 Not Found
5xx	Serverseitiger Fehler	500 Server Internal Error, 501 Not Implemented
6xx	Globaler Fehler	600 Busy Everywhere, 606 Not Acceptable

Tabelle 3.2: Kategorien der Antwortcodes

From Dieses Feld enthält einen Namen und eine URL, die den Absender eindeutig identifizieren soll. Zusätzlich kann der *From*-Header Parameter enthalten, die mit einem Semikolon vom eigentlichen Inhalt abgetrennt werden und durch ein Gleichheitszeichen getrennte Name-Inhalt-Paare sind. Der *tag*-Parameter ist dabei ein Pflichtparameter und identifiziert mit dem *tag*-Parameter des *To*-Headers und der Call-ID einen Dialog zwischen zwei SIP-User Agents. Ein beispielhafter *From*-Header sieht wie der folgende aus:

```
From: "Thomas Skora" <sip:4498742@sipgate.de>;tag=3CA654DB
```

Hierbei wird von den meisten Endgeräten bzw. Softphones der Text zwischen den Anführungszeichen angezeigt, um den Anrufer für den Benutzer zu identifizieren.

To Der *To*-Header identifiziert den logischen Empfänger der Anfrage und sollte beim Konstruieren der SIP-Anfrage bei allen Methoden bis auf Register der URL in der Anfragezeile entsprechen. Diese Headerzeile ist ähnlich der des *From*-Headers aufgebaut. Der *tag* ist kein Pflichtparameter, da er grundsätzlich vom zum URL gehörendem User Agent konstruiert wird, der allerdings noch keinen Dialog mit der Gegenstelle, die die Anfrage sendet, aufgebaut hat.

Contact Contact gibt eine SIP- bzw. SIPS-URI an, unter der die Gegenstelle für weitere Anfragen erreicht werden kann. Das Format entspricht dem der *To*- und *From*-Header.

```
Contact: "Thomas Skora" <sip:4498742@83.129.57.123;transport=udp>;q=1.0;expires=3600
```

Ein Parameter für diese Headerzeile ist *expires*, der die Zeit in Sekunden angibt, bis der Kontakt nicht mehr verfügbar ist. Es können auch mehrere Kontakte angegeben werden, die absteigend nach dem q-Parameter, der eine Zahl aus dem Intervall $[0; 1]$ enthält, priorisiert werden können.

Call-Id Dieser Header identifiziert mit den *From-* und *To*-Tags eindeutig einen Dialog und andere zusammenhängende SIP-Pakete, wie sie bei der Registrierung und Deregistrierung bei der *Register*-Anfrage auftreten. Die Call-Id sollte so gewählt werden, dass keine Kollisionen mit den Call-Ids von anderen User Agents auftreten können. Ein Beispiel für einen Call-Id-Header sieht folgendermaßen aus:

```
Call-ID: 1618485506@83.129.50.171
```

Max-Forwards Max-Forwards gibt die maximale Anzahl der Hops an, die das SIP-Paket durchlaufen darf. Jede Zwischenstation muss den Wert des Headers um Eins dekrementieren und das Paket, falls der Wert Null erreicht wurde, mit einem Fehler zurückweisen.

CSeq Der *CSeq*-Header besteht aus einer Zahl, maximal 2^{31}, und der Methode aus der Anfragezeile. Er dient dazu, Transaktionen innerhalb eines Dialogs zu ordnen. Ist die empfangene Sequenznummer einer Anfrage niedriger als die im Zustand gespeicherte, stellt dies eine Fehlerbedingung dar. Dagegen darf die Sequenznummer im empfangenen Paket um einen beliebigen Wert gestiegen sein, da es bei SIP möglich ist, dass Proxys zwischen den zwei miteinander kommunizierenden Stellen zwischenzeitlich weitere Aktionen herbeigeführt haben, in denen die Sequenznummer erhöht wurde. Beide Endpunkte eines SIP-Dialogs verwalten jeweils ihre eigene Sequenznummer. Ein Beispiel für einen *CSeq*-Header ist:

```
CSeq: 6061 BYE
```

Via Dieser Header gibt die Transportinformation für den nächsten Hop bei der Antwort auf die Anfrage an. Die Transportinformation beinhaltet den Protokollidentifier mit Versionsnummer, das Transportprotokoll, die Ziel-IP und einen eindeutigen *branch*-Parameter, der die jeweilige Transaktion im Proxy identifiziert. Eine derartige Headerzeile kann folgendermaßen aussehen:

```
Via: SIP/2.0/UDP 83.129.57.123;branch=z9hG4bK17EA0CC2
```

Durchläuft eine Anfrage die Proxys, trägt jeder Proxy einen *Via*-Header mit seinen Transportinformationen an erster Stelle ein. Bei einer Antwort wird das SIP-Paket an die im obersten *Via*-Header angegebene Transportadresse weitergeleitet. Diese Headerzeile wird beim Empfang des Pakets vom Zielproxy entfernt.

Record-Route Eine *Record-Route*-Headerzeile wird von einem Proxy hinzugefügt, wenn dieser bei den folgenden Anfragen ebenfalls auf dem Signalisierungspfad bleiben möchte. Sie enthält eine SIP- bzw. SIPS-URL und ggf. noch weitere Parameter. Alle *Reocrd-Route*-Header werden beim Beantworten der Anfrage in gleicher Reihenfolge in die Antwort hineinkopiert. Weiterhin wird aus den Header-Zeilen der *Route*-Header konstruiert, der im folgenden Punkt beschrieben wird.

```
Record-Route: <sip:4917324333351@217.10.79.8;ftag=5AD541C9;lr=on>
```

Route Der *Route*-Header enthält eine durch Kommas getrennte Folge von SIP- bzw. SIPS-URLs, die beim Weiterleiten des Pakets durchlaufen werden müssen. Erkennt ein Proxy seine Adresse an erster Stelle in der Route, muss er sie aus der Route entfernen und das Paket an den nächsten, nun an erster Stelle stehenden, Proxy in der Route weiterleiten. Da in SIP zwei Routing-Mechanismen existieren, muss an dieser Stelle zwischen URLs, die den *lr*-Parameter enthalten, und solchen, die ihn nicht enthalten, unterschieden werden. Ist der Parameter bei der ersten URL nicht enthalten, muss die URL aus der Anfrage ans Ende der Route angehängt und durch die URL des nächsten Proxys aus dem Header ersetzt werden. Dieses Verhalten wird auch *Strict-Routing* genannt und wurde in alten SIP-Spezifikationen verwendet.

```
Route: <sip:017324333351@217.10.79.9;ftag=5AD541C9;lr=on>,
       <sip:4917324333351@217.10.79.8;ftag=5AD541C9;lr=on>
```

Authorisierungsheader SIP verwendet den im RFC 2617 [FHBH+99] standardisierten Digest-Authentifizierungsmechanismus, der auch im HTTP-Protokoll verwendet wird. Dabei liefern die Header *WWW-Authenticate, Authenticate-Info* und *Proxy-Authenticate* den Challenge für eine Authentifizierung und *Authorization* und *Proxy-Authorization* den Response des Clients.

3.2.2 Dialoge, Transaktionen und Vorgänge

Ein Dialog stellt in SIP einen Austausch von Nachrichten zwischen zwei SIP-Endpunkten dar. Dabei wird ein Dialog durch die *Call-ID*, *From-* und *To*-Tags auf jeder Seite eindeutig identifiziert. Der Grund für die Hinzunahme der beiden Tags liegt darin begründet, dass Anfragen beim Routen durch Proxys auch aufgeteilt werden können. Das führt dazu, dass aus ursprünglich einem Dialog mehrere generiert werden und diese dann nur noch eindeutig identifiziert werden können, wenn ein endpunktspezifischer Wert mit einbezogen wird. Des Weiteren wird eine Ordnung der Anfragen mit Sequenznummern vorgenommen, die im letzten Abschnitt bereits beschrieben wurde. Ein Dialog wird durch eine positive Antwort auf eine Invite-Anfrage erstellt und durch eine BYE-Anfrage wieder beendet.

Eine Transaktion wird durch eine Anfrage und ihre Antworten repräsentiert, in der die Rollen des Clients und des Servers klar festgelegt sind. In SIP werden die Begriffe "Client" und "Server" nicht im klassischem Sinne verwendet, sondern durch die ausgeführte Aktion definiert. Der Client ist hierbei immer die Seite, die eine Anfrage stellt, während der Server die Antwort auf die Anfrage liefert.

Der Host, der eine Anfrage an einen anderen Host sendet, wird für die Dauer der Transaktion *User Agent Client* oder kurz UAC genannt. Der Host, der die Anfrage entgegennimmt und sie beantwortet, heißt *User Agent Server* oder kurz UAS.

Im Folgenden werden die wesentlichen SIP-Transaktionen vorgestellt, die zum Verständnis der Vorgänge bei der Signalisierung benötigt werden.

3.2.3 Registrierung

Bevor ein Benutzer mit seinem Endgerät die Telefoniedienste eines Anbieters in Anspruch nehmen kann, muss er zunächst bei einem Proxy des Anbieters registriert werden, was durch eine Register-Anfrage erfolgt, deren Ablauf in Abbildung 3.2 dargestellt ist. Zunächst wird eine Register-Anfrage gestellt, wie sie in Ausschnitten in Beispiel 3.1 dargestellt ist. Dabei enthält die Anfrage-URL den Server, bei dem sich der Client registrieren möchte, der *To*-Header die SIP- bzw. SIPS-URL, die registriert werden soll, und der *From*-Header die Adresse der registrierenden Instanz, im Normalfall sind die Adressen in beiden Headerzeilen gleich. Unter der im *Contact*-Header angegebenen Adresse kann die registrierte logische URL erreicht werden. Wie beim *Contact* üblich können mehrere Kontaktadressen unter einer logischen URL registriert werden. *Expires* gibt an, nach wie vielen Sekunden die Bindung zwischen logischer und Kontaktadresse wieder aufgehoben wird. Eine unmittelbare Deregistrierung kann vorgenommen werden, indem eine Zeit von Null angegeben wird. Dabei ist es zulässig im *Contact* einen Wildcard "*" zu verwenden, der für alle zu einer logischen Adresse vorhandenen Kontakte steht, und somit eine logische Adresse komplett zu deregistrieren. Wird in einer Registrierung keine Kontaktadresse angegeben, werden keine Änderungen vorgenommen. Dies wird insbesondere dazu verwendet, um alle bestehenden Bindungen zu erfragen, da eine Antwort auf eine Registrierung immer alle bestehenden Bindungen in *Contact*-Headern enthält.

Beispiel 3.1 *Eine auf die wichtigsten Headerzeilen komprimierte Register-Anfrage ist wie folgt aufgebaut:*

```
REGISTER sip:sipgate.de SIP/2.0
From: "Thomas Skora" <sip:4498742@sipgate.de>
To: "Thomas Skora" <sip:4498742@sipgate.de>
Expires: 900
Call-ID: 1714359340@83.129.2.232
Contact: "Thomas Skora" <sip:4498742@83.129.2.232;transport=udp>;q=1.0;
  methods="INVITE, OPTIONS, BYE, CANCEL, NOTIFY, ACK, REFER"
```

Nachdem ein Proxy eine Registrierung empfangen hat, antwortet dieser entweder mit einem Antwortcode von "`200 Ok`" oder einem Fehlercode. Am häufigsten dürfte hierbei der Fehler "`401 Unauthorized`" vorkommen. Im *WWW-Authenticate*-Header wird ein Challenge mitgesendet, der vom Client mit dem Shared Secret dazu verwendet wird, sich

Abbildung 3.2: Ablauf einer Register-Anfrage mit Authentifizierung

mit einem korrekten Response im *Authorization*-Header beim Proxy zu authentisieren. Die gesamte Registrierungsanfrage wird noch einmal mit der zusätzlichen Headerzeile und einer um eins inkrementierten Sequenznummer gesendet. Vom Server wird dann mit einem Statuscode geantwortet.

3.2.4 Rufaufbau, Verbindung und Rufabbau

Um ein Telefongespräch aufzubauen, sendet der Anrufende ein *Invite*-Paket mit der gewünschten Ziel-SIP-URL entweder an einen Proxy oder direkt zum Ziel. Erstere Vorgehensweise ist dann notwendig, wenn die IP-Adresse des gewünschten Ziels noch nicht bekannt ist oder das Ziel SIP-Signalisierungspakete nur vom eigenen Proxy annimmt. Sie ist in Abbildung 3.2 dargestellt und wird in diesem Abschnitt beschrieben. Die zweite Vorgehensweise wird verwendet, wenn keiner der eigenen Proxys eine Weiterleitung zum Ziel bzw. dem Proxy des Ziels anbietet.

In diesem Invite-Paket wird die Anfrage-URL zusätzlich noch in den *To*-Header kopiert. Die Angabe kann vom Ziel ausgewertet werden, um festzustellen, an welche URL das Paket ursprünglich gerichtet war, da die Anfrage-URL während des Durchlaufens der Proxys umgeschrieben werden kann. Der *From*-Header enthält neben der URL des Absenders dieses Pakets optional noch einen Namen, der vom Endgerät bzw. der empfangenden Software angezeigt werden kann. Im *Contact*-Header wird eine URL mit IP-Adresse angegeben, die für darauf folgende SIP-Anfragen genutzt werden kann, je-

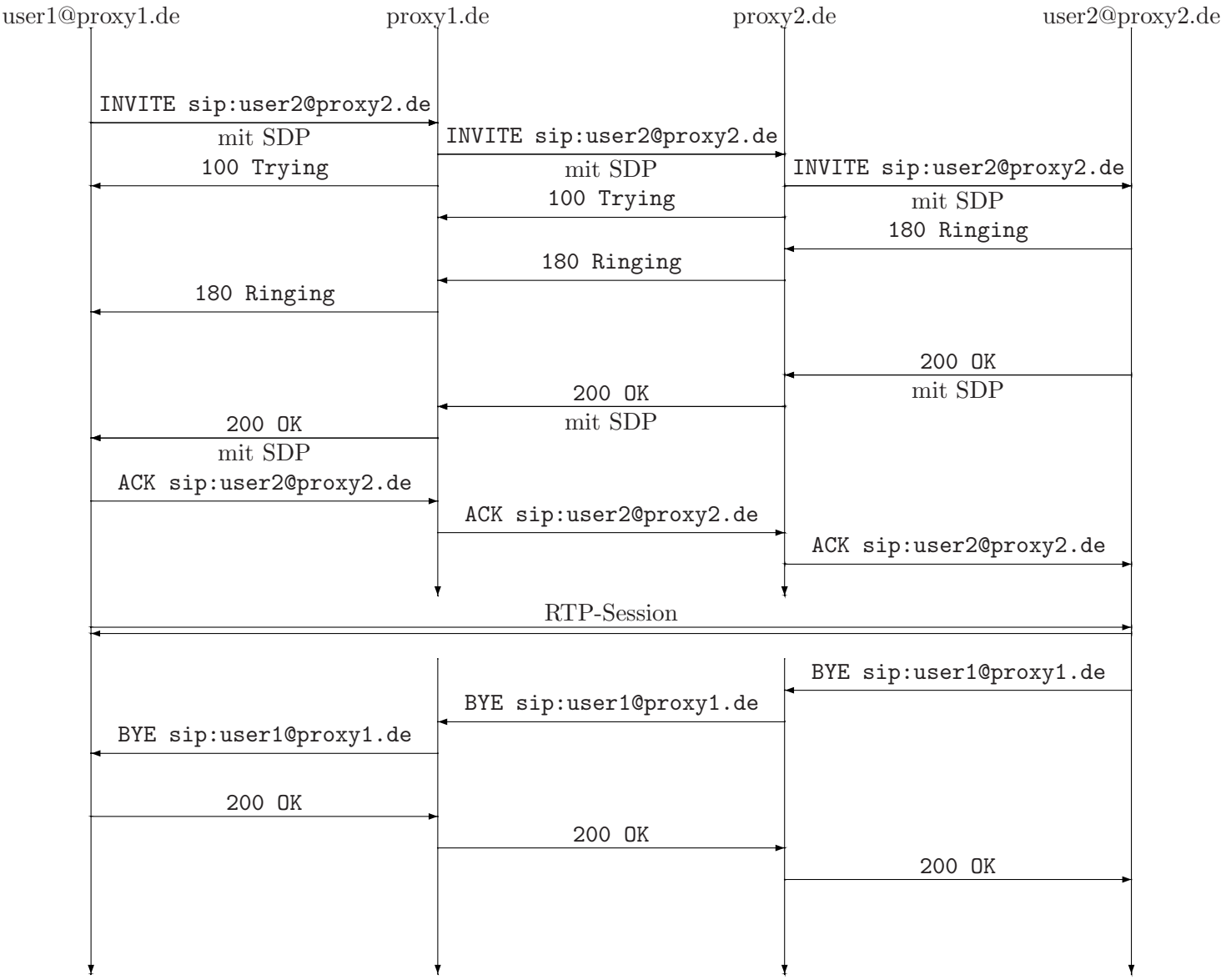


Abbildung 3.3: Rufauf- und abbau bei SIP

doch keine Bedeutung für den RTP-Mediendatenstrom hat. Die Parameter des RTP-Mediendatenstroms werden vollständig im SDP-Body definiert, dessen Format in Abschnitt 3.3 beschrieben wird. Jeder Anruf wird durch eine eigene eindeutige *Call-ID* identifiziert. Zusätzlich trägt sich der Sender noch in einem *Via*-Header ein, anhand dessen der erste Proxy die Antwort zum Absender der Anfrage routen kann.

Beispiel 3.2 *INVITE-Anfrage mit ausgewählten Headerzeilen:*

```
INVITE sip:01732433351@sipgate.de SIP/2.0
To: <sip:01732433351@sipgate.de>
From: "Thomas Skora" <sip:4498742@sipgate.de>;tag=7AED43CA
Contact: "Thomas Skora" <sip:4498742@83.129.60.207;transport=udp>
Via: SIP/2.0/UDP 83.129.60.207;branch=z9hG4bK22E74959
CSeq: 4884 INVITE
Call-ID: 510162691@83.129.60.207
Content-Type: application/sdp

[SDP-Body]
```

Gegebenenfalls wird der Proxy die *Invite*-Anfrage mit einem Statuscode "`407 Proxy Authentication Required`" beantworten und somit eine Authentifizierung durch den Client für diese Anfrage anfordern. In diesem Fall sendet der Client eine *Ack*-Anfrage und wiederholt die *Invite*-Anfrage mit einer zur Challenge passenden Response im Header *Proxy-Authorization*.

Hat der Proxy das Paket zur weiteren Verarbeitung akzeptiert, so sendet er die provisorische Antwort "`100 Trying`", die aussagt, dass die Anfrage weiter bearbeitet wird und eine endgültige Antwort noch aussteht. Die Antwort enthält dabei die gleichen To-, From-, CSeq- und Call-ID-Header, anhand denen der Absender insbesondere mit Hilfe der To- und From-Tags eine eindeutige Zuordnung zur Anfrage herstellen und den Benutzer über den Fortschritt informieren kann.

Der erste Proxy *proxy1.de* wird nun die Invite-Anfrage zum für die Zieldomain zuständigen Proxy *proxy2.de* weiterleiten. Vorher wird ein zusätzlicher Via-Header vor allen anderen Via-Headern eingefügt, um dem dann zuständigen Proxy eine Transportadresse für die Antwort zu liefern. Auch *proxy2.de* antwortet hier mit einem provisorischem "`100 Trying`" und leitet die Anfrage nach gleichem Muster zum Ziel weiter.

Am Ziel angekommen wird entweder eine endgültige Antwort wie "`486 Busy Here`" zurückgesendet, die angibt, dass der Angerufene nicht erreichbar ist oder eine provisorische Antwort "`180 Ringing`", die dem Anrufenden z.B. über ein Freizeichen mitteilt, dass dem Angerufenen nun ein eingehender Anruf z.B. über einen Klingelton signalisiert wird. Des Weiteren fügt die Gegenstelle einen To-Tag ein, mit dessen Hilfe der nun vollständig aufgebaute SIP-Dialog eindeutig identifiziert werden kann. Es kann ein Contact-Header mitgeliefert werden, der die gleiche Funktion wie der Contact-Header

des ursprünglichen Absenders der Invite-Anfrage hat. In der Praxis wird diese weitere provisorische Antwort vom Angerufenen, insbesondere bei Gateways, oft nicht mehr gesendet und direkt mit der endgültigen Antwort fortgefahren, wenn der Zustand des Angerufenen feststeht. In diesem Fall werden To-Tag und Contact in der endgültigen Antwort eingefügt. Beim Zurückrouten der Antwort entfernt jeder Proxy die oberste Via-Headerzeile, die die eigenen Transportparameter enthält und leitet diese an den nächsten Hop, der durch den neuen obersten Via-Header gegeben ist, weiter.

Bis zu diesem Punkt kann ein Invite durch eine Cancel-Anfrage abgebrochen werden, wobei jedoch die erste provisorische Antwort bereits eingetroffen sein muss. Die Cancel-Anfrage stellt eine eigene Transaktion dar. Ein Proxy bzw. Endpunkt erkennt die abzubrechende Transaktion anhand des Branch-Parameters im Via-Header, von dem in einem Cancel-Paket nur einer existieren darf, da die Antwort auf diese Anfrage nur einen Hop zurückgeroutet wird. Ist der Branch-Parameter nicht vorhanden oder nach einer alten SIP-Spezifikation konstruiert, die keine Eindeutigkeit dieses Parameters vorsah, wird die dazugehörige Transaktion anhand des Inhalts des Via-Headers, der Call-ID und der CSeq-Sequenznummer, und des To-Tags erkannt. Eine Cancel-Anfrage wird vom Proxy direkt mit einem "`200 OK`" beantwortet. Die eigentliche, abgebrochene Invite-Anfrage liefert dann zusätzlich noch eine Antwort "`487 Request Terminated`".

Nimmt der Angerufene das Gespräch an, wird auf gleiche Weise die endgültige Antwort "`200 OK`" zum Anrufenden geroutet. Bei dieser Gelegenheit fügen die dazwischen liegenden Proxys unter Umständen Record-Route-Header ein, die den anrufenden Client dazu anweisen, einen Route-Header in allen folgenden Anfragen einzufügen, der wiederum den Proxys eine Route für ein bestimmtes Paket vorgibt. Dies wird z.B. dann verwendet, wenn ein Proxy zu Abrechnungszwecken alle weiteren SIP-Anfragen mitbekommen möchte. In der Antwort fügt der Angerufene seine Medienbeschreibung in einem SDP-Body ein, um dem Anrufenden die verwendeten Medienparameter für den RTP-Datenstrom mitzuteilen. Die Aushandlung der Parameter ist in RFC 3264 [RS02] spezifiziert und wird in Abschnitt 3.3 beschrieben.

Diese Antwort wird mit einer weiteren Ack-Anfrage beantwortet, deren To- und From-Header mit den gleichen Werten wie in den vorhergehenden Anfragen und Antworten gefüllt werden. Die Sequenznummer im CSeq-Header wird nicht inkrementiert und es wird ein Via-Header hinzugefügt, obwohl eine Ack-Anfrage nicht mehr beantwortet wird. Der Hintergrund für diesen Three-Way-Handshake, der dem von TCP beim Aufbau von Verbindungen ähnelt, ist der, dass SIP-Pakete auch über unzuverlässige Transportprotokolle wie UDP übertragen werden können, was oft praktiziert wird. Das kann zur Folge haben, dass die endgültigen Antworten auf die Invite-Anfrage mehrfach wiederholt werden, da die andere Seite ansonsten nicht erfahren würde, dass das Gespräch angenommen oder aus welchem Grund es nicht angenommen werden kann. Im ersten Fall ist die Beschreibung des Medienstroms nicht vorhanden und damit wäre dieser unbrauchbar. Mit der Ack-Anfrage kann der Empfänger einer Antwort der Gegenstelle mitteilen, dass keine weiteren Sendeversuche notwendig sind. Damit dies bewerkstelligt wird, muss jede

weitere eintreffende Antwort mit einem Ack bestätigt werden.

Nach diesen Vorgängen kann die Session beginnen. Bei Voice-over-IP bedeutet dies, dass in beide Richtungen jeweils ein RTP-Datenstrom mit den im SDP-Body ausgehandelten Parametern (Codec, Bitrate) aufgebaut wird, in dem die Sprachdaten übertragen werden. Die RTP-Pakete zwischen den beiden Endpunkten werden direkt übertragen und durchlaufen im Gegensatz zu den Signalisierungspaketen keine Proxys mehr.

Beendet einer der Teilnehmer das Gespräch, sendet sein User Agent eine BYE-Anfrage mit der URL der jeweiligen Gegenstelle als Anfrage-URL. Da diese Anfrage noch zum durch die Invite-Anfrage erstellten Dialog gehört, müssen From- und To-Tags und die Call-Id aus dem Invite weiter verwendet werden, während der Branch-Parameter im Via-Header neu generiert wird und die lokale Sequenznummer inkrementiert wird. Die BYE-Anfrage wird zur Gegenstelle geroutet, welche diese mit einem Antwortcode beantwortet, der bis auf einen eventuellen Fehler in den meisten Fällen "`200 OK`" sein wird. Beide Seiten schließen beim Empfang der jeweiligen Anfrage bzw. Antwort ihren RTP-Strom und beenden damit den SIP-Dialog.

3.2.5 Forking

Eine Variation der Signalisierung eines Anrufs stellt das sogenannte Forking dar. Wie bereits beschrieben, können unter einer SIP-URL mehrere Kontaktadressen registriert werden. Das kann z.B. dafür verwendet werden, ein Telefon am Arbeitsplatz und eins Zuhause zu registrieren und beide bei einem eingehenden Anruf gleichzeitig klingeln zu lassen. Um das zu realisieren, kann ein Proxy eine Anfrage in mehrere weitere Anfragen aufteilen, die dann an die verschiedenen Kontaktadressen gesendet werden.

Die Antworten werden anschließend gesammelt und sinnvoll zusammengefasst, so dass der ursprüngliche Absender der Anfrage eine Antwort erhält. Dabei werden zunächst Fehler aus der Klasse der globalen Fehler (6xx) weitergeleitet. Sind keine Antworten aus dieser Klasse enthalten, werden Antworten aus den Klassen mit niedriger Nummer bevorzugt behandelt. Werden z.B. wie in Abbildung 3.4 die Antwortcodes "`486 Busy Here`", "`200 OK`" in dieser Reihenfolge empfangen, während ein Client noch nicht geantwortet hat, wird die Antwort "`200 OK`" ausgewählt und zum Absender der Invite-Anfrage gesendet. Antworten aus der Klasse 2xx nehmen bei Invites eine Sonderstellung ein. Beim Empfang dieser Antwort wird nicht mehr auf weitere Antworten gewartet, sondern die erfolgreiche Antwort unmittelbar weitergeleitet, um keine Verzögerungen beim Rufaufbau zu verursachen.

3.3 Session Description Protocol (SDP)

Das Session Description Protocol wird von Signalisierungsprotokollen wie SIP dazu verwendet, die Eigenschaften des Mediendatenstroms wie den verwendeten Codec und Bitrate oder auch ein Transportprotokoll und die Zieladresse mit den verwendeten Ports

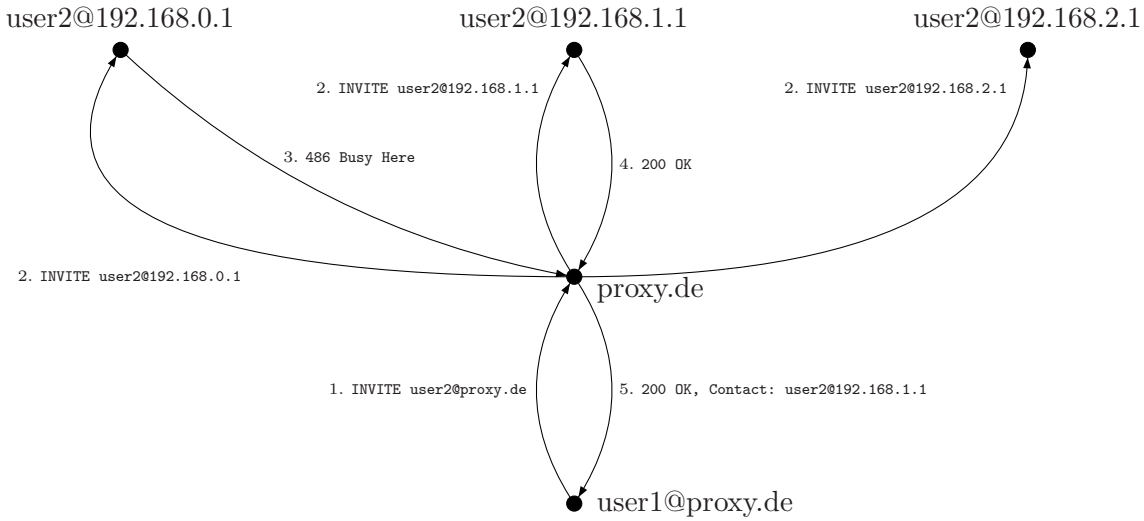


Abbildung 3.4: Forking bei SIP

auszuhandeln und festzulegen. Ursprünglich war SDP [HJ98] dafür vorgesehen, Medien in Multicast-Umgebungen zu beschreiben, z.B. Videokonferenzen, die zu einem bestimmten Zeitpunkt stattfinden. Mit dem Aufkommen von Voice-over-IP bestand Bedarf an einem Protokoll, mit dem zwei Teilnehmer in einer Punkt-zu-Punkt-Verbindung ihre Medieneigenschaften untereinander abstimmen konnten, was dazu führte, dass eine entsprechende Vorgehensweise in [RS02] spezifiziert wurde. In der heutigen Internet-Telefonie ist es üblich, dass SDP-Bodies in SIP-Paketen ausgetauscht werden.

3.3.1 Aufbau

SDP ist ein textbasiertes Format, in dem zeilenweise jeweils eine Eigenschaft und deren Wert durch ein Gleichheitszeichen voneinander getrennt zugewiesen werden. Hierbei wird zwischen globalen Eigenschaften, die für die gesamte Session gelten, und Medieneigenschaften unterschieden. Das Protokoll gibt ein strenges Schema vor, nach dem solche Beschreibungen aufgebaut sind. Zwischen der Eigenschaft, dem Gleichheitszeichen und dem Wert dürfen keine Leerzeichen auftreten und es ist eine feste Reihenfolge vorgegeben, in der die Zuweisungen zu erfolgen haben.

Eine Beschreibung beginnt immer mit den globalen Eigenschaften, der mehrere Medienbeschreibungen folgen können. Bestimmte Eigenschaften aus der globalen Sektion können von Eigenschaften der Medienbeschreibungen wieder lokal für das jeweilige Medium überschrieben werden. In Beispiel 3.3 beginnt die globale Sektion bei `v` und endet

mit dem Attribut `t`.

Beispiel 3.3 *Ein SDP-Body eines SIP-Pakets, in dem mehrere mögliche Codecs Beschrieben werden und eine Transportadresse festgelegt wird, kann folgendermaßen aussehen:*

```
v=0
o=root 13169 13169 IN IP4 62.141.41.44
s=session
c=IN IP4 62.141.41.44
t=0 0
m=audio 11570 RTP/AVP 8 0 3 97 18 2 5 110 7
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:3 GSM/8000
a=rtpmap:97 iLBC/8000
a=fmtp:97 mode=30
a=rtpmap:18 G729/8000
a=rtpmap:2 G726-32/8000
a=rtpmap:5 DVI4/8000
a=rtpmap:110 speex/8000
a=rtpmap:7 LPC/8000
a=silenceSupp:off - - - -
```

Der erste Parameter in der globalen Sektion gibt die SDP-Version an, die zurzeit 0 ist. Daraufhin folgt der Origin, abgekürzt durch *o*. Dieser Parameter gibt die Quelle der Sitzung an, die an erster Stelle mit einer Userbezeichnung beginnt. Es folgt die numerische Sitzungskennung und eine Versionsnummer, die die Erkennung aktualisierter Beschreibungen ermöglicht. Anschließend folgt der Netzwerktyp, der in den gängigen Fällen *IN* ist und für den Typ Internet steht. Danach wird der Adresstyp und eine Adresse des entsprechenden Typs angegeben. Die gängigen Adresstypen sind IP4 und IP6. Sie stehen für die jeweiligen Versionen des Internet Protocols. Der Sitzungsname, der durch den Parameter *s* gegeben ist, muss immer vorhanden sein, erfüllt bei VoIP jedoch keine Aufgabe.

Das *Connection Data*-Feld kann entweder in der globalen Sektion oder in einer Medienbeschreibung auftauchen und gibt den Netzwerk- und Adresstyp sowie die Adresse für den Medienstrom wie im o-Feld an. Das t-Feld gibt die Start- und Endzeit für eine Sitzung an und muss immer vorhanden sein. Da es bei VoIP nicht verwendet wird, wird hier jeweils eine Null für beide Zeiten angegeben.

Das m-Feld (Media Announcment) gibt an erster Stelle den verwendeten Medientyp an, der bei VoIP typischerweise *audio* ist, aber auch *video* und *data* sein kann. Gefolgt wird dies durch den für die Übertragung erwünschten Ziel-UDP-Port. Auf das obige

Code	Bezeichnung
	Statische Codes
0	G.711 PCMU (μ-Law)
3	GSM
8	G.711 PCMA (A-Law)
4	G.723
7	LPC
18	G.729
	Dynamische Codes
97	iLBC
110	Speex

Tabelle 3.3: RTP/AVP Medientypen

Beispiel bezogen würde das bedeuten, dass der Empfänger der SDP-Beschreibung die Mediendaten an Port 11570 senden soll. Das dritte Feld zeigt das verwendete Transportprotokoll, welches *udp* oder *RTP/AVP* (Real-time Transport Protocol Audio/Video Profile [SC03]) sein kann. Im Normalfall wird bei VoIP RTP/AVP verwendet. Anschließend folgt eine Liste von unterstützten Formaten bzw. Codecs, die direkt mit Medienformaten aus dem RTP/AVP-Standard korrespondieren. Das erste Element ist das bevorzugte Medienformat. Eine Auswahl von gängigen Codecs ist in Tabelle 3.3 dargestellt. In [SC03] sind alle Codes, außer 96 bis 127, statisch festgelegt oder reserviert. Codes im dynamischen Bereich können für jede Sitzung zu beliebigen Medienformaten zugewiesen werden. In Tabelle 3.3 werden verbreitete Formate mit ihren typischen Codes aufgeführt.

Um die Formate den gegebenen Kennungen zuordnen zu können, was insbesondere bei den dynamisch zugeordneten Formaten notwendig ist, werden weitere Informationen benötigt, die in a-Feldern angegeben werden. Dieser Feldtyp ist dafür vorgesehen, zusätzliche Attribute zu definieren, die nicht in SDP vorgesehen sind. Die Attribute können entweder durch ein einzelnes Schlüsselwort dargestellt werden oder durch ein mit einem Doppelpunkt getrenntes Attribut-Wert-Paar zugeordnet werden. Das für die Codeczuordnung verwendete Attribut ist *rtpmap* und gibt an erster Stelle die Medienkennung an, die vom Namen des Codecs und der verwendeten Samplingrate gefolgt wird. Optional kann noch die Anzahl der verwendeten Audiokanäle angegeben werden, die bei VoIP im allgemeinen eins beträgt und somit weggelassen wird. Ein weiteres Attribut, das verwendet wird, um zusätzliche Parameter für ein gegebenes Medienformat anzugeben, ist *fmtp*. Dieses Attribut enthält die betroffene Formatkennung und formatspezifische Attributzuweisungen, die durch *Name=Wert*-Paare dargestellt werden. Im obigen Beispiel wurde so für den iLBC-Codec der Modus 30 ausgewählt, der besagt, dass in einem RTP-Paket 30ms Sprache codiert werden sollen. Des Weiteren wurde mit dem Attri-

but *silenceSupp* festgelegt, dass keine Erkennung von Sprechpausen stattfindet. Ist diese Funktion eingeschaltet, kann mit genanntem Attribut unter anderem ein Schwellenwert für die Lautstärke und der Zeitraum festgelegt werden, ab dem keine Datenübertragung mehr stattfinden soll.

3.3.2 Aushandeln der Sitzungsparameter

SDP, wie es in [HJ98] spezifiziert wurde, war primär dafür gedacht, Multicast-Sitzungen zu beschreiben. Da VoIP im Allgemeinen eine Unicast-Anwendung ist, gibt es hier andere Anforderungen an das Protokoll, die von SDP nicht erfüllt werden können. Deshalb müssen sich beide Seiten auf Medienformate einigen, die von beiden Clients verstanden werden und die Präferenzen der jeweiligen Seite zumindest annähernd erfüllen, um den jeweiligen Bedingungen wie der verfügbaren Bandbreite gerecht zu werden. In [RS02] wurde dafür ein Ablauf definiert, in dem eine Seite ein initiales Angebot macht, das von der Gegenseite beantwortet wird.

Die Liste der Medienformate entspricht in der Reihenfolge den Präferenzen des Senders der SDP-Beschreibung. Der Empfänger eines solchen Angebots beantwortet jede durch ein m-Feld aufgeführte Medienbeschreibung durch ein damit korrespondierendes m-Feld in der Antwort. Ein solcher Medienstrom kann vom Empfänger ganz abgelehnt werden, indem als Port eine Null angegeben wird. Das muss vor allem dann geschehen, wenn keines der aufgelisteten Medienformate unterstützt wird. Ansonsten muss in der Antwort mindestens eines der angebotenen Medienformate pro akzeptierten m-Feld aufgelistet werden. Nach Möglichkeit sollte die Reihenfolge aus dem Angebot beibehalten und nur nicht unterstützte Formate ausgeblendet werden.

Nachdem so die unterstützten Formate ausgewählt wurden, können beide Seite damit beginnen, die Mediendaten zu senden. Beide Seiten sollten jeweils das bevorzugte und unterstützte Format der Gegenseite, was durch die Reihenfolge in der Formatliste determiniert ist, auswählen und die vereinbarte Kennung für die RTP-Pakete übernehmen.

Während einer Sitzung können die Parameter neu verhandelt werden, indem dieser Ablauf mit entsprechend modifizierten SDP-Beschreibungen durchgeführt wird. Neben den verwendeten Formaten können auch der Port und die IP-Adresse verändert werden.

3.4 Skype

Skype ist ein proprietärer VoIP-Dienst der Skype Technologies S.A. mit Sitz in Luxemburg. Die verwendeten Protokolle sind nicht offengelegt und es existieren nur wenige Informationen über den Aufbau und die Funktionsweise dieses Systems. In [BS04, Des] wurde der Traffic, der von einem Skype-Client erzeugt wird, untersucht, um mit den daraus gewonnenen Informationen Rückschlüsse auf die Funktionsweise des Protokolls zu bekommen. Wesentliche Merkmale des Skype-Protokolls sind:

- Es arbeitet bereits bei der Signalisierung im Peer-to-Peer-Betrieb.
- Sowohl Signalisierung, als auch Sprachübertragung finden verschlüsselt statt.
- Auch in restriktiven und NAT-Umgebungen funktioniert es problemlos.

In [Des] wurde der Aufbau der Skype-Pakete genauer untersucht. Dabei wurde festgestellt, dass ein Skype-Paket aus einem unverschlüsseltem Header und verschlüsselten Daten besteht, Die Verschlüsselung von Signalisierungspaketen mit dem RC4-Algorithmus dient hier nur zur Verschleierung, da der Schlüssel aus dem Paketheader generiert wird und die Informationen somit jedem Empfänger eines solchen Pakets zur Verfügung stehen. Die Übertragung der Sprache wird dagegen nach [Ber05] mit dem AES-Algorithmus im Counter Mode verschlüsselt, der in Abschnitt 3.6.2 für SRTP beschrieben wird. Vor der Sprachkommunikation vereinbaren beide Clients einen gemeinsamen Sitzungsschlüssel durch ein proprietäres und derzeit unbekanntes Schlüsseltausch-Protokoll.

Die Netzarchitektur von Skype ist weitgehend dezentral aufgebaut und besteht aus folgenden Komponenten:

Login-Server Die einzige zentrale Komponente, an der sich ein Client anmelden muss, um Skype zu nutzen.

Super Node Jeder Teilnehmer des Skype-Netzwerks kann zu einem Super-Node ernannt werden. Das geschieht nach bestimmten Kriterien wie der verfügbaren Bandbreite und der Uptime des Clients. Der Super-Node übernimmt zusätzliche Aufgaben für die Signalisierung und stellt für einen Client einen Location Service dar.

Client Ein einfacher Teilnehmer, der die Dienste von Skype wie die Telefonie oder Instant Messaging nutzen kann.

3.4.1 Registrierung und Login

Bei der ersten Registrierung generiert der Client ein RSA-Signaturschlüsselpaar und berechnet den Hash des vom Benutzer gewählten Passworts. Der geheime Teil des RSA-Schlüssels wird mit dem Hash und dem Benutzernamen über eine mit dem AES-Algorithmus verschlüsselte Verbindung zu einem zentralen Skype-Server übertragen, dessen Authentizität während des Verbindungsaufbaus überprüft wird [Ber05]. Der Server überprüft den Benutzernamen auf Einmaligkeit und speichert ihn mit der nochmals gehashten Version des Passworts. Der Skype-Server generiert ein Zertifikat, in dem unter anderem der Benutzternname und der öffentliches RSA-Schlüssel von Skype signiert werden, wodurch die Zuordnung zwischen Schlüssel und Benutzer von jedem Client überprüft werden kann.

Möchte sich ein Client mit dem Skype-Netz verbinden, muss er zunächst einen Super Node finden. Dies geschieht durch den Host Cache, der bei einer frischen Installation mit

vorgegebenen IPs gefüllt ist und im laufendem Betrieb bis zu 200 Super Nodes enthält [BS04]. Gelingt es einem Client nicht, einen solchen Super Node zu finden, schlägt der Login bereits an dieser Stelle fehl.

Nachdem sich der Client mit einem Super Node verbunden hat, muss er sich beim Login-Server authentisieren. Beim Login generiert der Client einen RSA-Schlüssel, mit dem das gehashte Passwort verschlüsselt wird, und sendet diese Informationen an den Login-Server, der die Identität bestätigt und das dem entsprechenden Super Node mitteilt.

Während eines Logins kann des Weiteren beobachtet werden, dass der Client UDP-Pakete mit weiteren Hosts austauscht, von denen ausgegangen wird, dass es ebenfalls Skype-Clients bzw. Super Nodes sind. Von diesem Vorgang wird angenommen, dass er zur weiteren Ankündigung der Präsenz des Skype-Clients dient. Befindet sich der Client hinter einer Firewall, die UDP-Pakete komplett filtert, wird der Login über TCP abgewickelt. In jedem Fall wird eine TCP-Verbindung zu einem Super Node offen gehalten, über die beim weiteren Betrieb von Skype Anfragen gesendet werden. Die Login-Prozedur beinhaltet eine Erkennung des NAT-Typs und der Firewall-Konfiguration, da ein Client sich im weiteren Verlauf der Netzwerkkonfiguration entsprechend verhält.

3.4.2 Rufsignalisierung und Sprachübertragung

Um einen Teilnehmer des Skype-Netzes zu finden, sendet der Client Daten an den Super Node. Nachdem dieser geantwortet hat, werden vier UDP-Pakete an bis dahin nicht beteiligte Hosts gesendet. Schlägt die Suche fehl, werden weitere Daten über TCP mit dem Super Node ausgetauscht und anschließend weitere acht Hosts über UDP abgefragt. Dies wird so lange wiederholt, bis der gesuchte Name gefunden ist oder die Suche fehlschlägt. Kann der Client nur über TCP kommunizieren, wird die Suche vom Super Node übernommen, weil in diesem Fall nur die TCP-Verbindung genutzt wird [BS04].

Wurde ein Teilnehmer auf diese Weise gefunden, dann kann er, falls verfügbar, angerufen werden. Die Signalisierung findet grundsätzlich über TCP statt. Haben beide Clients öffentliche IP-Adressen, wird die Signalisierung direkt zwischen beiden Clients durchgeführt. Sobald sich ein Client in einer NAT-Umgebung befindet, erfolgt die Signalisierung über einen Zwischenknoten. Zusätzlich wird der Super Node und weitere Hosts in die Signalisierung mit einbezogen und während des Telefonats bestehen TCP-Verbindungen mit diesen zusätzlichen Knotenpunkten, über die mehrmals in einer Sekunde Pakete ausgetauscht werden. Welchen Zweck diese zusätzlichen Verbindungen erfüllen, ist jedoch unbekannt.

Die Übertragung der Sprachdaten erfolgt bei direkter Konnektivität zwischen beiden Clients direkt und über UDP. Bei einer NAT-Umgebung werden die UDP-Pakete über einen Host geroutet, der bereits bei der Signalisierung angesprochen wurde. Wird auch UDP gefiltert, werden die Sprachdaten über eine TCP-Verbindung ausgetauscht.

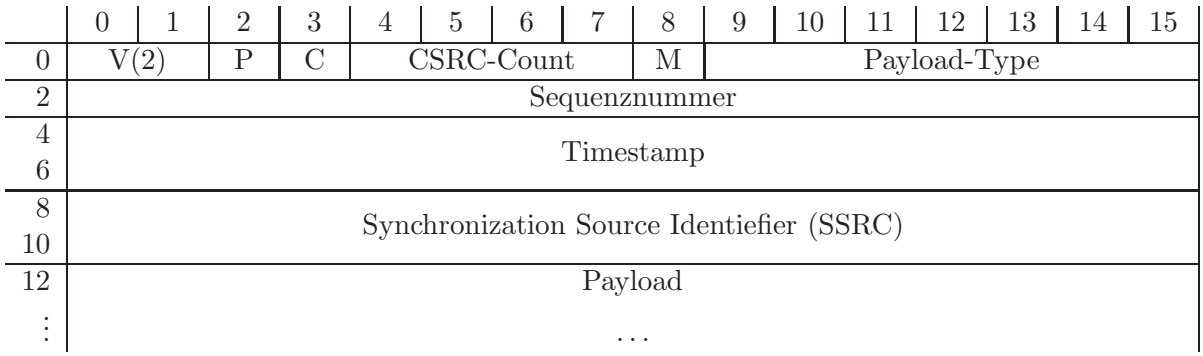


Abbildung 3.5: Aufbau eines RTP-Pakets

3.5 Real-Time Transport Protocol (RTP)

Das in RFC 3550 [SCFJ03] standardisierte Real-Time Transport Protocol wird sowohl bei SIP als auch bei H.323 das Standardprotokoll, damit die Sprachdaten in Echtzeit übertragen werden können. Bei der Übertragung von Sprache kommt es nicht auf Zuverlässigkeit an. Deshalb wird als Transportprotokoll UDP verwendet. Außerdem führen verlorene Pakete bei einer nicht zu hohen Paketverlustrate zu weniger Qualitätseinbußen als Verzögerungen oder Jitter, der durch wiederholte Übertragungsversuche entstehen würde. Der Transport von RTP über TCP ist zwar vorgesehen, wird in VoIP allerdings nicht implementiert.

3.5.1 Aufbau

Ein Header, wie er in Abbildung 3.5 dargestellt ist, enthält zusätzliche Informationen zur Verwaltung einer RTP-Session. In den ersten zwei Bit des Pakets steht die Versionsnummer des RTP-Protokolls, derzeit 2, der ein Padding-Flag folgt. Ist das gesetzt, wurde das Paket mit Bytes aufgefüllt, um eine bestimmte Blockgröße zu erreichen. Die Anzahl der irrelevanten Bytes ist im letzten Byte des Pakets angegeben. Das nächste Flag gibt an, ob sich hinter dem Header noch ein Erweiterungsheader befindet, der profilspezifische Parameter enthalten kann und in jedem Fall die Länge des zusätzlichen Headers enthält. Der Contributing Sources (CSRC) Count gibt die Anzahl der CSRC-Identifier an. Ein CSRC-Identifier kann von einem Mixer, der mehrere Sessions zu einer mischt, um dem Empfänger einer solchen Session die Möglichkeit zu geben, die einzelnen Quellen zu identifizieren. Die Liste der Contributing Sources folgt dem Synchronization Source-Identfier, der RTP-intern eine Session identifiziert und ihr einen eigenen Raum von Sequenznummern und Zeitstempeln zuordnet. Nach dem CSRC-Count folgt im RTP-Header das Marker-Bit, das anwendungsspezifisch für unterschiedliche Zwecke verwendet werden darf. Bei VoIP wird es verwendet, um nach einer Stillephase, in der kein Paket gesendet wurde, das erste Paket zu markieren [SC03]. Darauf folgt der Payload-Typ, der

Kennung	Name	Funktion
200	Sender Report	Reports von aktiven Sendern
201	Receiver Report	Reports von inaktiven Teilnehmern, die nur empfangen
202	Source Description	Beschreibung von RTP-Sessions
203	Bye	Beenden der RTP-Session
204	Application Defined	Für anwendungsspezifische Erweiterungen

Tabelle 3.4: RTCP-Pakettypen

bei VoIP den verwendeten Codec zur Übertragung angibt. Eine Auswahl von Codecs wurde bereits in Tabelle 3.3 angegeben.

Eine wichtige Funktion übernehmen die Sequenznummer und der Zeitstempel. Durch die Sequenznummer, deren initiale Wert zufällig bestimmt werden sollte, kann ein Paketverlust erkannt und die Anzahl der verlorengegangenen Pakete bestimmt werden. Weiterhin kann die korrekte Reihenfolge der Daten wieder hergestellt werden, falls diese beim Empfänger nicht mehr gegeben ist. Für den Zeitstempel wird bei Audiodaten die Länge eines Samples verwendet. Bei einer kontinuierlichen Übertragung wächst der Zeitstempel also mit jedem Paket um einen konstanten Wert. Wird die Übertragung bei Stille unterdrückt, wird zwar der Zeitstempel für nicht übertragene Samples weiter inkrementiert, die Sequenznummer jedoch nur für wirklich gesendete Pakete.

3.5.2 RTP Control Protocol (RTCP)

Das RTP Control Protocol wurde im gleichen RTP wie RTP spezifiziert. Es kann von RTP-Kommunikationspartnern verwendet werden, um aktuelle Eigenschaften des RTP-Paketstroms wie die Verlustrate oder den Jitter mitzuteilen, um so beispielsweise bei zu schlechten Werten einen ressourcenschonenderen Codec umschalten zu können. Des Weiteren ist eine rudimentäre Signalisierung mit RTCP möglich, deren Bedeutung bei VoIP aufgrund der bereits vorhandenen Signalisierungsprotokolle gering ist. Insgesamt hat RTCP im VoIP-Bereich keine große Bedeutung. Von den im Rahmen dieser Arbeit getesteten Geräten senden nur das Grandstream BudgeTone RTCP-Bye-Pakete und das Opensource-Softphone Linphone Sender Reports.

Es gibt mehrere RTCP-Pakettypen, die in Tabelle 3.4 aufgelistet sind. Mehrere RTCP-Pakete werden üblicherweise in einem Datagramm übertragen.

Weil die Teilnehmer bei VoIP generell aktiv sind, d.h. RTP-Pakete senden, ist von den Report-Pakettypen nur der Sender Report relevant. Er enthält einen Header, der wie ein RTP-Header die Version des verwendeten Protokolls und ein Padding-Flag beinhaltet. Da für jede empfangene RTP-Session mit unterschiedlichen SSRCs ein Report gesendet werden kann, folgt die Anzahl der Reports und die Länge des einzelnen RTCP-Pakets. In diesem Header identifiziert sich der Sender des Reports auch mit seiner SSRC. Im Sender Report sind folgende Daten enthalten:

NTP-Zeitstempel Dieser Zeitstempel gibt die echte Uhrzeit an und ist optional.

RTP-Zeitstempel Die NTP-Zeit in den Zeiteinheiten der RTP-Session. Dieser Wert kann zur Synchronisation unterschiedlicher Quellen verwendet werden.

Gesendete Pakete Die Anzahl der gesendeten Pakete sowie Bytes.

Dem Sender-Report folgen die Receiver-Reports, die durch die SSRCs der empfangenen RTP-Sesisons unterschieden werden. Diese Reports enthalten unter anderem folgende Informationen:

Paketverlust Sowohl der Anteil der verlorenen Pakete seit dem letzten Report als auch die Gesamtzahl der verlorenen Pakete werden angegeben.

Jitter Der Jitter kann durch die Varianz der Ankunftszeitdifferenzen der Pakete abgeschätzt und in RTP-Zeiteinheiten gemeldet werden.

Letzter Report Zeitstempel und verstrichene Zeit seit dem Empfang des letzten Reports.

Ein Receiver Report enthält ausschließlich die sinnvolle Teilmenge der Informationen und besteht daher nur aus dem letzten Teil.

Die Session Description enthält lediglich informative Werte, wie den kanonischen Namen der Session, Name des Verantwortlichen oder eine E-Mail-Adresse. Einzig der kanonische Name ist ein Pflichtfeld und soll die Unterscheidung von Sessions unabhängig vom SSRC machen.

3.6 Secure RTP (SRTP)

Secure RTP ist ein in RFC 3711 [BMN+04] standardisiertes RTP-Profil, das die Zielsetzung verfolgt, die Vertraulichkeit der übertragenen Daten herzustellen und deren Integrität zu sichern, ohne den Overhead übermäßig zu erhöhen.

3.6.1 Aufbau

Ein SRTP-Paket ist wie in Abbildung 3.6 aufgebaut und enthält die gleichen Header wie ein normales RTP-Paket. Der Unterschied besteht darin, dass der Payload verschlüsselt und dieser mitsamt des Headers authentifiziert werden kann. Zusätzlich kann sich am Ende des Pakets noch ein Master Key Identifier, welcher einen Master-Key und eine Signatur zur Authentifizierung befinden. Ob das Paket diese letzten beiden Header enthält und welche Länge sie haben, ist durch die Sitzungsparameter bestimmt, die vom SRTP-Stack verwaltet werden.

Die RTP-Sequenznummer wird in SRTP um weitere 32 Bit auf 48 Bit erweitert, wobei die hinzugefügten höherwertigen 32 Bit implizit im SRTP-Stack als Anzahl der Überläufe

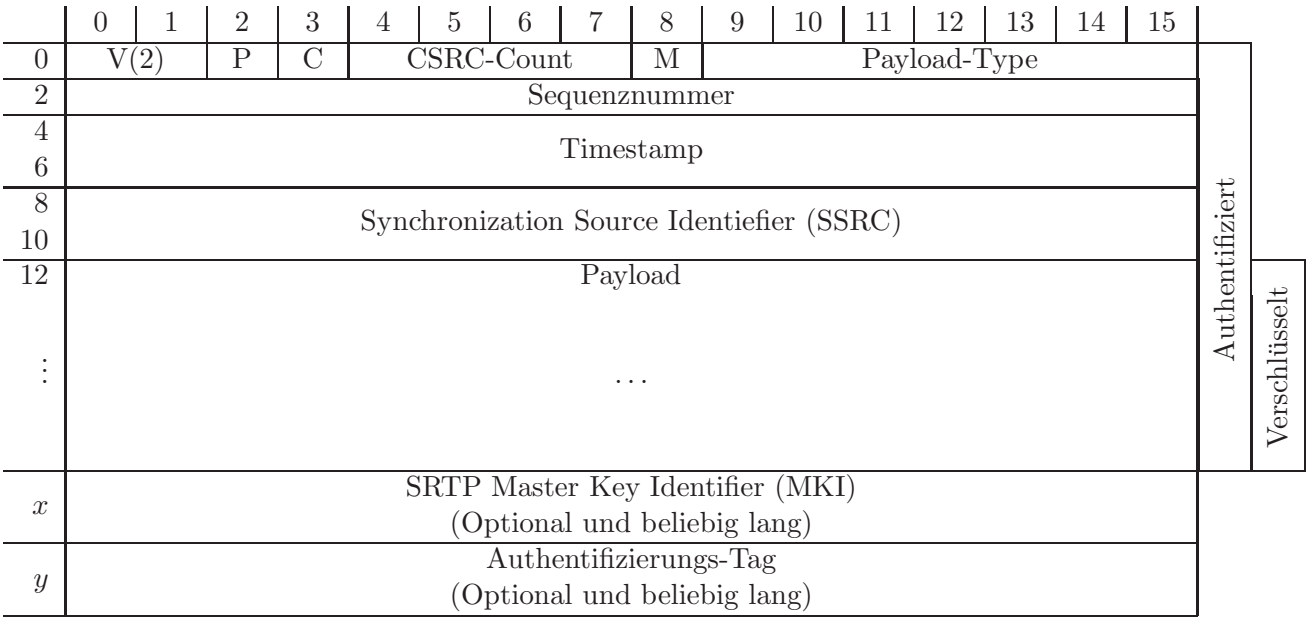


Abbildung 3.6: Aufbau eines SRTP-Pakets

ROC der Sequenznummer verwaltet und nicht in den SRTP-Paketen übertragen werden. Die erweiterte Sequenznummer wird in der SRTP-Terminologie *Paketindex* genannt und mit $i = ROC \cdot 2^{16} + SEQ$ berechnet. Da der Paketindex mit in die kryptographischen Operationen des Protokolls einfließt, darf jeder Paketindex nur einmal mit einem bestimmten Schlüssel verwendet werden. Neben den kryptographischen Operationen wird der Paketindex auch für den Schutz vor Replay-Angriffen verwendet, indem dieser für die letzten n Pakete gespeichert und bei der Ankunft von Paketen verglichen wird, ob das Paket bereits empfangen wurde. Das ist der Fall, wenn die Sequenznummer bereits bekannt oder kleiner ist als die älteste gespeicherte in den n Sequenznummern.

3.6.2 Verschlüsselung und Authentifizierung

Von der Spezifikation von SRTP ist AES im Counter Mode [LRW00] mit SRTP-spezifischen Verfeinerungen als Standard-Verschlüsselungsalgorithmus vorgesehen. Der Counter Mode ist so definiert, dass für den Schlüssel k und einen ganzzahligen Initialisierungsvektor c ein Strom von Schlüsselbits mit

$$k' = E(k, c) \cdot E(k, c + 1) \cdot E(k, c + 2) \cdot \ldots$$

erzeugt wird, wobei $\cdot$ die Konkatenation darstellt. Eine Nachricht x der Länge m wird durch eine bitweise XOR-Operation $x'_j = k'_j \oplus x_j$ auf allen Bits $0 \leq j \leq m$ verschlüsselt. Eine Nachricht (c, x') kann von einem Empfänger, der den geheimen Schlüssel k besitzt,

Label	Abgeleiteter Schlüssel
0x00	SRTP-Verschlüsselung
0x01	SRTP-Authentifizierung
0x02	SRTP-Salt
0x03	SRTCP-Verschlüsselung
0x04	SRTCP-Authentifizierung
0x05	SRTCP-Salt

Tabelle 3.5: Labels für die Berechnung der Key-Id

wieder dechiffriert werden. Für ein SRTP-Paket mit dem Paketindex i und einem Salt k_s aus dem aktuellen kryptographischen Kontext, ist der Initialisierungsvektor c durch

$$c = (k_s \cdot 2^{16} \oplus (\text{SSRC} \cdot 2^{128}) \oplus (i \cdot 2^{16})$$

bestimmt.

Um ein vorhandenes Schlüsseltauschprotokoll wie zu entlasten, werden die benötigten Schlüssel für einen kryptographischen Kontext aus dem Master Key mit einem Pseudo-Zufallsgenerator $P_n(k, x)$ abgeleitet. Dabei stellt n die Länge der generierten Zufallssequenz in Bits, k einen geheimen Schlüssel und x eine Eingabe der Länge von m Bits dar. Für P_n wird im RFC AES im Counter Mode mit dem Initialisierungsvektor $c = x \cdot 2^{16}$ empfohlen.

Zur Ableitung des Schlüssels wird zunächst ein Parameter r aus dem aktuellen Paketindex i und der Anzahl der Pakete d, nach der neue Schlüssel abgeleitet werden sollen, mit

$$r = \begin{cases} \lfloor i/d \rfloor & \text{, wenn } d > 0 \\ 0 & \text{, wenn } d = 0 \end{cases}$$

berechnet. Aus der Konkatenation von einem Label aus Tabelle 3.5 und r wird eine Key-Id kid zusammengesetzt, die mit dem Master-Salt s den Wert $x = kid \oplus s$ berechnet. Dabei werden die beiden Komponenten der XOR-Operation an ihren niederwertigen Bits aneinander ausgerichtet. Der im Label gewählte Schlüssel der Länge n wird dann mit $k' = P_n(k_m, x)$ aus dem Master-Key k_m berechnet.

Ein neuer Schlüssel muss entweder dann generiert werden, wenn ein anderer MKI im SRTP-Paket verwendet wird, oder wenn die Schlüssel durch Verlassen eines vorher festgelegten Paketindex-Bereichs nicht mehr gültig sind. Letztere Möglichkeit erspart den zusätzlichen Platzbedarf für den MKI in jedem SRTP-Paket.

Zur Authentifizierung ist in SRTP die Hashfunktion SHA1 vorgesehen, die den Authentifizierungsschlüssel, den integritätsgeschützten Teil des Pakets aus Abbildung 3.6 und den intern verwalteten *ROC* berechnet. Der berechnete Hashwert darf beliebig gekürzt werden, um nicht zu viel Overhead im SRTP-Paket zu erzeugen.

4 Angriffe

Um der Sicherheitsanalyse in Kapitel 6 eine technische Grundlage zu geben und dem Leser ein Gefühl dafür zu geben, wo Schutzbedarf in den Protokollen besteht, werden in diesem Kapitel einige Angriffe vorgestellt. Zunächst werden in Abschnitt 4.1 drei Varianten eines Denial-of-Service-Angriff gezeigt, der den Rufaufbau auf Signalisierungsebene mit verschiedenen Methoden unterbindet und damit die Verfügbarkeit einschränkt. Zwei weitere DoS-Angriffe, die zu einer hohen Auslastung von Proxys und Endgeräten führen können, werden im darauf folgenden Abschnitt 4.2 behandelt. In Abschnitt 4.3 folgen Angriffe, die durch Manipulation der Signalisierungspakete den RTP-Sprachdatenstrom oder die ganze Signalisierung zu einem beliebigen Host umleiten.

In Abschnitt 4.4 werden zwei Angriffe gezeigt, die ein Gespräch auf Teilen des Signalisierungspfads beenden können, ohne dass der Endpunkt auf der anderen Seite etwas mitbekommt. Ziel eines solchen Angriffs ist es, einen VoIP-Betreiber ein Gesprächsende vorzutäuschen, so dass keine Gebührenabrechnung für die restliche Zeit des Gesprächs stattfindet. Abgeschlossen wird dieses Kapitel durch eine Übersicht von weiteren Bedrohungen, in die zwar keine VoIP-Protokolle involviert sind, die die IP-Telefonie aber trotzdem einschränken können.

4.1 Denial-of-Service: Abbruch von Gesprächen

Die in diesen Abschnitt beschriebenen Angriffe werden oft in der Literatur als Beispiel für die einfache Angreifbarkeit von SIP genannt. An dieser Stelle wird eine detaillierte technische Beschreibung und Untersuchung durchgeführt. Anhand dieser Untersuchungen wurde die in Abschnitt 4.1.4 beschriebene Implementierung vorgenommen.

Wird SIP über UDP transportiert, kann ein aktiver Angreifer, der nicht in der Lage ist, Pakete zu manipulieren, beliebige Nachrichten in einen bestehenden SIP-Dialog einfügen. Das befähigt ihn dazu, einen Rufaufbau zu stören oder ein bestehendes Telefonat abzubrechen. Dies geschieht, indem gefälschte Anfragen oder Antworten an einen oder beide SIP-Endpunkte gesendet werden. In diesem Abschnitt werden drei Möglichkeiten vorgestellt, einen solchen Angriff durchzuführen. Sie haben jeweils unterschiedliche Qualitäten im Sinne der Erfolgswahrscheinlichkeit und Erkennbarkeit durch den Endbenutzer.

Bei allen in diesem Abschnitt vorgestellten Angriffsvarianten müssen Header teilweise gefälscht werden und unter Umständen IP-Adressen von Quell- und Zielhosts vertauscht werden, damit die Pakete authentisch wirken und von den angegriffenen VoIP-Komponenten angenommen werden. Bei der Implementierung der Angriffe wurden die

Header	Inhalt des Headers			
	Cancel	Response	Bye	
			$A \longrightarrow B$	$A \longleftarrow B$
Anfrage-URL	INVITE	nicht vorhanden	ACK	From aus ACK
Contact	INVITE	INVITE	2xx-Antwort	2xx-Antwort
From	aus INVITE	aus INVITE	From aus ACK	To aus ACK
To	INVITE	INVITE	To aus ACK	From aus ACK
Via	INVITE	INVITE	ACK	2xx
CSeq	INVITE	INVITE	ACK inkrementiert	ACK inkrementiert
Content-Length	0	0	0	0
IP-Quelladresse	Quelladresse	Quelladresse	Quelladresse	Zieladresse
IP-Zieladresse	Zieladresse	Zieladresse	Zieladresse	Quelladresse

Tabelle 4.1: Konstruktion eines SIP-Pakets bei Packet-Injection DoS-Angriffen

SIP-Pakete wie in Tabelle 4.1 angegeben konstruiert. Die Spalte für den Bye-Angriff, der in zwei Richtungen durchgeführt werden kann, wurde für die beiden Varianten in zwei Unterspalten unterteilt. A stellt in der Spaltenüberschrift den Anrufer dar und B den Angerufenen. Der Pfeil gibt die Richtung des Bye-Pakets an, in das es vom Angreifer aus gesendet wird. In der Tabelle selbst wird die Quelle der Headerinhalte angegeben. Zum Beispiel bedeutet ein Invite in der Cancel-Spalte und To-Zeile, dass der To-Header in der vom Angreifer konstruierten Cancel-Anfrage durch den To-Header aus der vorangegangenen Invite-Anfrage ersetzt wird. In den letzten zwei Zeilen der Tabelle wird außerdem die Herkunft der Quell- und Ziel-IP-Adresse bestimmt. Die Angaben beziehen sich in allen vier Fällen auf die IP-Adressen des Invite-Pakets. Neben den in der Tabelle angegebenen Headern wird bei allen Angriffen noch die Call-Id und der User-Agent aus dem Invite-Paket sowie beim Response-Angriff Route und Record-Route aus der 2xx-Antwort in das gefälschte Paket übernommen.

4.1.1 Gefälschte Antworten

Bei dieser Variante des Angriffs, die in Abbildung 4.1 dargestellt ist, sendet der Angreifer nach dem Empfang einer Invite-Anfrage eine beliebige Antwort an den Sender der Anfrage. Der Empfänger dieser Antwort kann ohne weitere Authentifizierungsmechanismen nicht zwischen einer legitimen und der gefälschten Antwort unterscheiden und wird bei Fehlercodes im Bereich 4xx bis 6xx die Invite-Transaktion sofort beenden ([HSSR02], S. 82). Der Empfänger der Invite-Anfrage, welcher ein Proxy sein kann, der die Anfrage weiterleitet, oder auch das eigentliche Ziel, setzt die Signalisierung weiter fort. Das führt dazu, dass der ursprüngliche Sender der Anfrage weitere Antworten erhält. Dies können provisorische Antworten mit einem 1xx-Code sein, eine weitere negative Antwort, falls der Empfänger wirklich nicht erreichbar ist, oder eine positive 2xx-Antwort. Da der Cli-

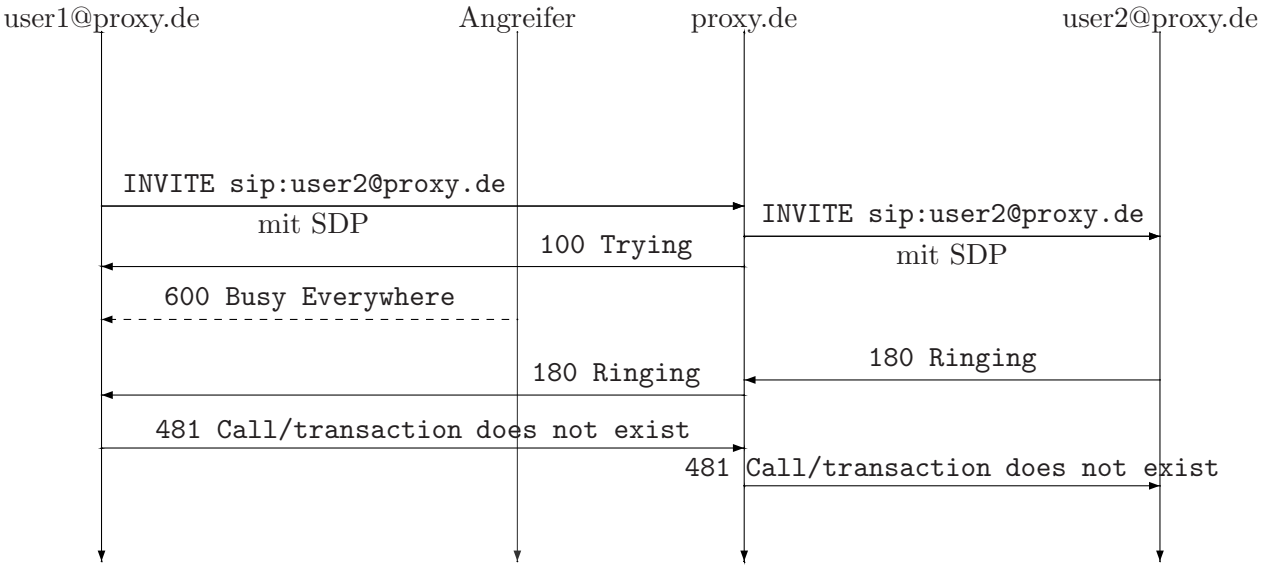


Abbildung 4.1: DoS mit gefälschter Antwort

ent die Transaktion bereits beendet hat, werden diese Antworten mit einem Fehler "481 Call/Transaction does not exist" abgewiesen, was auch auf der anderen Seite des Signalisierungspfads zum Abbruch des Rufaufbaus führt. Nimmt der Zielclient das Invite mit einem 2xx-Code an, bevor dieser Fehler ihn erreicht, beginnt er mit dem Senden der RTP-Pakete, die vom Ziel je nach Konfiguration mit ICMP-Nachrichten abgewiesen oder verworfen werden.

Dieser Angriff zeigt eine fundamentale Schwäche der meisten SIP-Implementationen. Es ist zwar möglich, Anfragen, bis auf die Ausnahmen CANCEL und ACK, mit dem HTTP-Digest-Verfahren [FHBH+99] zu authentifizieren, die Authentifizierung von Antworten mit dem rspauth-Header ist allerdings nur optional und wird häufig nicht implementiert.

Für den Angreifer bietet diese Angriffsvariante den Vorteil, dass eine beliebige Ursache für das Scheitern des Gesprächsaufbaus vorgetäuscht werden kann. Dadurch ist der Angriff für einen Benutzer nur schwer als solcher zu erkennen, so kann z.B. ein Besetzzeichen legitim sein. Nachteilig wirkt sich aus, dass der Angriff nur innerhalb des Zeitfensters zwischen dem Invite-Paket und der endgültigen Antwort der Gegenstelle vollzogen werden kann. Außerhalb dieses Zeitraums müssen keine Antwortpakete mehr verarbeitet werden, wodurch der Angreifer in eine Wettbewerbssituation mit der Gegenstelle tritt.

Die Signalisierung in die Richtung des Invites läuft auch nach einem erfolgreichen Angriff weiter. Das Telefon des Angerufenen wird klingeln, aber kein Gespräch zustande

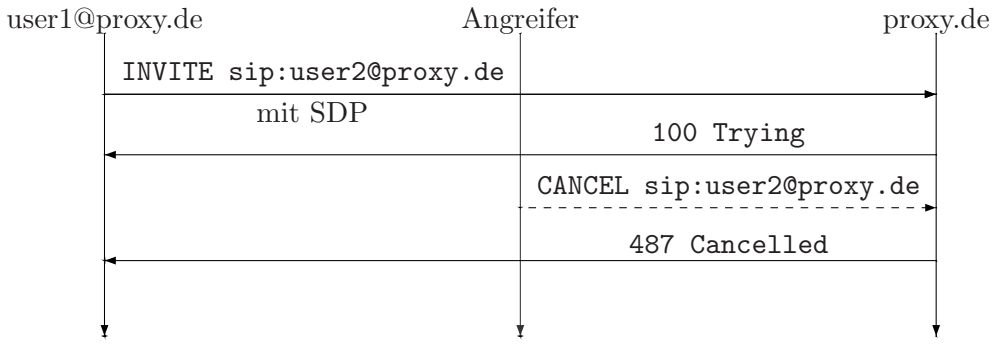


Abbildung 4.2: DoS mit einer Cancel-Anfrage

bringen, weil die positive 2xx-Antwort vom Anrufer mit einem Fehler `481 Call/transaction does not exist` abgelehnt wird. Um diesen auffälligen Effekt beim Angerufenen zu unterbinden, kann dieser Angriff mit dem im nächsten Abschnitt vorgestellten Cancel-Angriff kombiniert werden. Er beeinflusst die Signalisierung in die andere Richtung.

4.1.2 Gefälschte CANCEL-Anfragen

Die fehlende Authentifizierung für Cancel-Anfragen lässt sich auch für DoS-Angriffe nutzen. Bei diesem in Abbildung 4.2 dargestellten Angriff wird direkt nach der provisorischen Antwort zur Invite-Anfrage ein Cancel in die gleiche Richtung des Invites gesendet, was zum Abbruch der Invite-Transaktion im Proxy mit der entsprechenden Antwort führt.

Da eine Cancel-Anfrage nur zwischen der ersten vorläufigen Antwort des Proxys und der endgültigen Antwort des anderen Endpunktes erfolgen kann, ist das Zeitfenster für ein erfolgreiches Senden der Abbruchsanforderung noch kleiner als beim im letzten Abschnitt vorgestellten Response-Angriff. Invite-Anfragen, die direkt mit einer positiv mit einem 2xx-Code beantwortet werden, können mit dieser Angriffsvariante gar nicht beeinflusst werden. Diese Situation kann z.B. bei einem Direct-IP-Call einer Gegenstelle mit einer automatischen Ansage eintreten, die das Gespräch nach dem Invite direkt annimmt. Ein weiterer Nachteil für den Angreifer ist, dass der Benutzer eine relativ ungängige Antwort vom Proxy bekommt und den Angriff dadurch als solchen identifizieren kann. Der Vorteil ist jedoch, dass eine Authentifizierung von Cancel-Anfragen von der SIP-Spezifikation her nicht vorgesehen ist, da das aufgrund des zeitkritischen Charakters dieser Anfrage keinen Sinn ergeben würde.

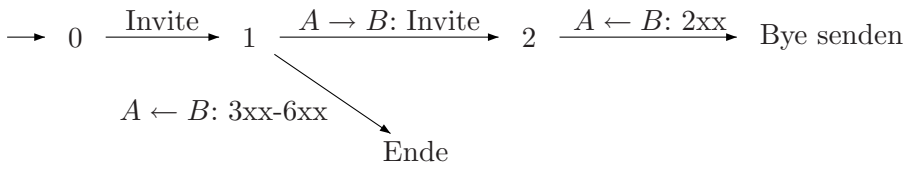


Abbildung 4.3: Zustände des Bye-Angriffs

4.1.3 Vorzeitiger Gesprächsabbruch durch gefälschte BYE-Anfragen

Die letzte Variante dieses DoS-Angriffs lässt ein Gespräch zustande kommen und beendet es während des laufenden Gesprächs durch das Einfügen gefälschter Bye-Anfragen in den bestehenden Dialog zu einem beliebigen Zeitpunkt. Die Durchführung des Angriffs gestaltet sich etwas komplexer als bei den vorangegangenen Angriffen, weil die Zustände des Rufaufbaus bis zum Zeitpunkt, in dem der eigentliche Angriff durch das Senden des Bye-Pakets erfolgt, vom Angreifer mit verfolgt werden müssen. Die Zustände mit ihren Übergängen sind in Abbildung 4.3 in Form eines endlichen Automaten dargestellt. Auf die Darstellung als Message-Sequence-Chart wurde an dieser Stelle verzichtet, weil sie weitgehend dem eines normalen Gesprächaufbaus und -abbaus (Abbildung 3.3) entspricht. Der einzige Unterschied besteht darin, dass das Bye-Paket von einem Angreifer und nicht von einem der beiden Endpunkte gesendet wird.

In Zustand 0 wartet der Angreifer auf eine Invite-Anfrage. Wird eine empfangen, muss zunächst ausgewertet werden, ob das sich im Aufbau befindende Gespräch gestört werden soll oder für den Angreifer nicht von Interesse ist. Ist das der Fall, wird in Zustand 1 übergegangen. In diesem Zustand wartet der Angreifer eine endgültige Antwort ab, die ihn im Falle einer negativen Antwort dazu bewegt, den Dialog intern zu beenden und die Zustände nicht mehr weiter zu verwalten. Ist die Antwort positiv, wird in Zustand 2 übergegangen, in dem auf die Ack-Anfrage gewartet wird, aus der das Bye-Paket gemäß den Regeln aus Tabelle 4.1 konstruiert werden kann.

Bei diesem Angriff kann eine Richtung bestimmt werden, in die die Bye-Anfrage gesendet wird. Je nach Richtung müssen dafür die Inhalte der From- und To-Header vertauscht und die Via-Header aus der 2xx-Antwort oder aus dem ACK-Paket übernommen werden. Es ist möglich, den Angriff in beide Richtungen gleichzeitig zu vollziehen. Wird der Angriff nur in eine Richtung durchgeführt, bleibt die Verbindung einseitig bestehen.

Im Gegensatz zu den anderen zwei Varianten treten bei diesem Angriff keine Wettbewerbssituationen auf. Der Angreifer muss sein Paket für einen erfolgreichen Angriff nicht schneller senden als die Gegenstelle. Ein Nachteil liegt jedoch darin, dass das Gespräch erst nach dessen Beginn abgebrochen werden kann, was verhältnismäßig auffällig ist.

4.1.4 Implementierung: sip-kill

Das Perl-Skript *sip-kill*[1] ist eine Implementierung aller drei Angriffsvarianten. Es benötigt eine installierte libpcap, um Pakete von Netzwerkinterfaces abzuhören, und Raw-Sockets, um IP-Pakete mit gefälschten Absenderadressen senden zu können. Die Bedienung des Programms erfolgt über die Kommandozeile einer Shell. Das heißt, es wird mit einem mit Minus-Zeichen vorangestellten Parameter übergeben und es werden optional reguläre Ausdrücke angegeben, die die Angriffsziele beschreiben.

Mit dem Parameter *m*, gefolgt von Cancel, Response oder Bye, wird eine der in diesem Abschnitt vorgestellten Angriffsmethoden ausgewählt. Für die Angriffsart Response kann zusätzlich noch mit dem Parameter *r* ein Fehlercode und mit *t* der dazugehörige beschreibende Text gesetzt werden. Beim Bye-Angriff gibt der Parameter *d* die Richtung des Bye-Pakets bzw. der Bye-Pakete mit den Schlüsselworten to, from oder both an. Die Richtung wird hierbei aus Sicht des Angerufenen angegeben.

Zusätzlich wurde die Option *h* geschaffen, mit der Erkennungszeichen von Dialogen, wie die Call-Id und die From- und To-Tags, entfernt oder mit zufälligen Werten gefüllt werden können. Dazu nimmt dieser Parameter eine durch Kommas separierte Liste von Beschreibungen entgegen, die jeweils aus zwei Zeichen besteht. Das erste Zeichen gibt an, was mit einem Header geschehen soll. Ein *r* bedeutet, dass das entsprechende Element entfernt wird, bei einem *m* wird es durch Zufallswerte ersetzt. Das zweite Zeichen gibt das zu verändernde Element an. *f* und *t* beziehen sich auf From- und To-Tags und *c* auf die Call-Id. Mit diesen Modifikationen der SIP-Pakete können blinde Angriffe auf SIP-Implementierungen getestet werden. Lässt sich bei einem Gerät ein Gespräch beenden, obwohl alle Dialogerkennungszeichen entfernt oder verändert wurden, könnte auch ein blinder Angreifer, der keine Kenntnis über die zum Rufaufbau ausgetauschten Pakete besitzt, trotzdem Pakete generieren, die von einem Gerät als gültig anerkannt werden. Die Entfernung oder Veränderung der Dialogerkennungszeichen kann mit `-h rf,rt,rc` oder `-h mf,mt,mc` bewirkt werden.

Die Hauptfunktionalität der Implementierung befindet sich in der Funktion `process_sip`, die ankommende Pakete zunächst vorverarbeitet und in einzelne Zeilen zerlegt. Ankommende Invite-Anfragen werden vorab darauf hin überprüft, ob sie den angegebenen Mustern entsprechen, um die Verarbeitung gegebenenfalls vorzeitig zu beenden. Wurde die Invite-Anfrage akzeptiert, werden die IP-Adressen und Ports des Pakets und das Paket selbst in einem Hash gespeichert. SIP-Dialoge werden dabei immer mit der Call-Id identifiziert. Im folgenden Teil wird je nach ausgewählter Angriffsvariante unterschiedlich vorgegangen.

Beim Cancel-Angriff wird erst überprüft, ob es sich um eine provisorische Antwort handelt und ob sie zu einer vorher empfangenen Invite-Anfrage gehört. Ist das der Fall, wird gemäß den Regeln aus Tabelle 4.1 aus dem gespeicherten Invite-Paket eine Cancel-Antwort konstruiert und mit gefälschten IP-Adressen und UDP-Ports über ein Raw-

[1] `http://skora.net/voip/sip-kill`

Socket gesendet.

Wurde der Response-Angriff gewählt, ist der Ablauf bis auf das Warten auf die provisorische Antwort der selbe, da die Antwort unmittelbar nach dem Invite-Paket gesendet werden kann. Für die Bye-Variante wurde die in Abbildung 4.3 dargestellte State-Machine implementiert.

4.2 Denial-of-Service: Erzeugen von Signalisierungstraffic

In diesem Abschnitt werden zwei Angriffe vorgestellt, mit denen ein Angreifer durch das Ausnutzen von Schwächen im SIP-Protokoll mit wenig Aufwand viel Traffic zwischen Proxys zu produzieren, was zu einem Denial of Service führen kann.

4.2.1 Via-Spoofing

In [She03] wird ein Angriff beschrieben, der zu einer hohen Auslastung von Proxys führen und auch zur Verschleierung von DDoS-Angriffen verwendet werden kann. Grundlage des Angriffs sind die Via-Header, die für eine SIP-Antwort die Route zum Ursprungshost vorgeben.

Ein Angreifer kann eine Antwort so konstruieren, dass die Antwort an einen beliebigen Host gesendet wird, indem er als erstes einen Via-Header mit den Kontaktdaten des SIP-Proxys und darauf folgend einen weiteren Via-Header mit der Zieladresse des Hosts, der das Paket erhalten soll, einsetzt. Das Paket wird so über den SIP-Proxy zum Ziel geroutet, wodurch der Angreifer Pakete an einen Host senden kann, der die Adresse des Proxys sieht.

Eine Variante des Angriffs ist es, Schleifen in die Via-Header zu konstruieren, so dass eine Antwort zwei oder mehrere Proxys durchläuft, bis es am Ende, wenn alle Via-Header verarbeitet sind, vom letzten Proxy verworfen wird. Auf diese Weise kann ein Angreifer Proxys mit wenig Aufwand stark belasten, weil er ein vorgefertigtes SIP-Paket versenden kann, um in den angegriffenen Proxys mehrfach die komplette Verarbeitung eines SIP-Pakets auszulösen.

Eine mögliche Lösung des Problems wird in [She03] beschrieben. Zusätzlich zum Branch-Parameter würde ein Proxy beim Weiterleiten einer Anfrage einen Cookie als Via-Parameter mitsenden. Die Form des Cookies kann von der Proxy-Software frei bestimmt werden, sollte aber aus einem Zeitstempel und einem Hash über die Call-Id, CSeq, Via-Header, die Anzahl der Via-Header, den Zeitstempel des Cookies und einen geheimgehaltenen Wert bestehen. Bei einer ankommenden Antwort müsste der oberste zum Proxy gehörende Via-Header den Cookie enthalten und sich aus den entsprechenden Headern und dem empfangenen Zeitstempel wieder berechnen lassen. Ist kein Cookie vorhanden, schlägt die Verifikation fehl oder zeigt der Zeitstempel an, dass der Cookie abgelaufen ist, soll das Paket verworfen werden.

Die Absicherung gegen derartige Angriffe sichert zusätzlich die Integrität der Header, die in die Berechnung des Hashwerts einfließen.

4.2.2 Massives Forking

Das in Abschnitt 3.2.5 beschriebene Forking kann von einem Angreifer, der bei zwei Proxys, die Forking von Anfragen erlauben und keine weitere Überprüfung auf Schleifen im Signalisierungspfad durchführen, zu einer exponentiellen Gesamtzahl von Anfragen führen, die sich beide Proxys gegenseitig senden [LHS05].

Dazu muss der Angreifer auf jedem Proxy, proxy1 und proxy2, jeweils zwei Accounts besitzen. Jeder Account von proxy1 verweist nach dem Registrieren durch die Angabe im Contact-Header auf die beiden Accounts in proxy2. Andersrum verweist jeder Account von proxy2 wieder zurück zu proxy1.

Wird nun eine Invite-Anfrage an einen der vier Accounts gestellt, wird diese Anfrage an jedem Proxy geforkt und an den anderen Proxy weitergeleitet. In jedem Schritt verdoppelt sich die Anzahl der aktiven Invite-Anfragen. Insgesamt muss ein Proxy, wenn n der Wert des Max-Forwards-Headers ist, $\sum_{i=1}^{n} 2^i$ Anfragen verwalten, was relativ schnell zu einem Ausfall beider Proxys führen dürfte.

4.3 Umleiten des RTP-Medienstroms

Eine weitere Familie von Angriffen, die oft in der Literatur genannt werden und die Abhörbarkeit von ungeschützten Telefonaten demonstrieren, werden in diesem Abschnitt beschrieben.

Kann ein Angreifer Signalisierungspakete abfangen, bedeutet das noch nicht, dass er den durch RTP-Pakete übertragenen Sprachdatenstrom auch abhören kann. Das liegt daran, dass beide Datenpfade in VoIP im Normalfall nicht gleich sind, weil bei der Signalisierung eine Reihe von Proxys dazwischengeschaltet sein kann. Die Medienübertragung hingegen erfolgt direkt zwischen den Endpunkten.

4.3.1 Umleiten des Gesprächs mit einer 3xx-Antwort

Der DoS-Angriff aus Abschnitt 4.1.1 kann so erweitert werden, dass dem anrufenden Client eine Umleitung empfohlen wird. Dazu muss der Angreifer ein Invite mit einem 3xx-Code beantworten und die SIP-URL, auf die das Gespräch umgeleitet werden soll, im Contact-Header angegeben werden.

Wird eine Anfrage mit dem Code 302 beantwortet, wird damit signalisiert, dass das Ziel unter der URL in der Anfrage nicht erreichbar ist. Gleichzeitig wird im Contact-Header eine URL angegeben, unter der der Teilnehmer erreicht werden kann. Die Dauer der Umleitung kann mit einem zusätzlichen Expires-Header angegeben werden. Ist dieser nicht vorhanden, ist die Umleitung nur für eine Anfrage gültig. Da das Cachen von

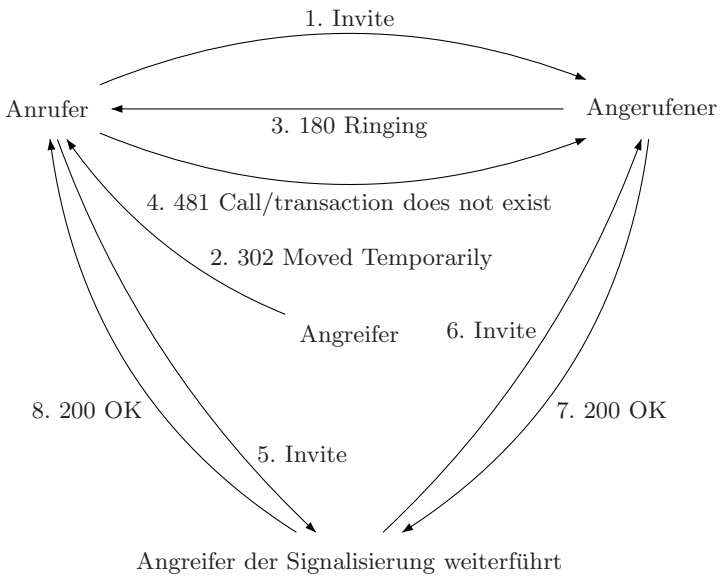


Abbildung 4.4: Abfangen eines Gesprächs mit gefälschten Antworten

Umleitungen optional ist, müssen folgende Invite-Anfragen vom Angreifer ebenfalls beantwortet werden, um den Angriff erfolgreich für mehrere Telefonate durchzuführen.

Mit einem 301-Antwortcode wird mitgeteilt, dass der Benutzer dauerhaft nicht mehr unter der alten Adresse erreichbar ist und stattdessen die URL aus dem Contact-Header weiter verwendet werden soll. Im SIP-RFC [HSSR02] wird sogar empfohlen, Adressbucheinträge entsprechend anzupassen. Obwohl das eigentlich sinnvoll erscheint, ist es aus dem Blickpunkt der Sicherheit zweifelhaft, wenn nicht garantiert werden kann, dass die Antworten authentisch sind.

Der Ablauf des Angriffs ist in Abbildung 4.4 dargestellt. Die für das Verständnis des Angriffs irrelevanten Teile wurden aus Gründen der Übersichtlichkeit ausgelassen. Schritt 4 stellt das Einschleusen der Antwort durch den Angreifer nach einem Invite des Anrufers (1) dar. Für dieses Beispiel wird angenommen, dass der Angreifer Pakete zwar mitlesen, aber nicht abfangen kann. Dadurch tritt der bereits im letzten Absatz des Abschnitts 4.1.1 beschriebene Effekt ein, bei dem der Angerufene noch eine provisorische Antwort sendet (3), diese aber vom Anrufer abgelehnt wird (4).

Wurde die Umleitung vom Anrufer oder dessen Endgerät bestätigt, setzt der Anrufer die Signalisierung mit einem Invite an das neue Ziel fort (5). Möchte der Angreifer die Verbindung mit dem Anrufer zustande kommen lassen, muss er an dieser Stelle die Signalisierung zum anderen Endpunkt fortsetzen. Die URL des Ziels ist dem Angreifer aus dem Invite im ersten Schritt bekannt. Findet der Angriff verteilt statt, muss die

URL an die Stelle, die die Signalisierung weiterführt, übermittelt werden. Sie kann auch durch die Weiterleitungs-URL bestimmt oder in sie codiert werden.

Bevor die Signalisierung mit einem Invite an das Anrufziel (6) fortgeführt wird, sollte dem Anrufer der Fortschritt des Vorgangs mit provisorischen Antworten angezeigt werden, die nicht in der Abbildung dargestellt sind. Die finale Antwort des Angerufenen (7) wird zum Angerufenen durchgestellt (8), womit die Verbindung zwischen beiden Seiten aufgebaut ist.

Für den RTP-Sprachdatenstrom zwischen beiden Seiten bestehen zwei Möglichkeiten:

- Der Angreifer ersetzt beim Weiterleiten die IP-Adressen in den SDP-Bodys durch seine eigene Adresse oder die eines RTP-Proxys, ähnlich der Vorgehensweise des als nächstes vorgestellten Angriffs. Dadurch kann auch der Medienstrom mitgeschnitten und manipuliert werden.

- Der Angreifer belässt die IP-Adressen in den SDP-Bodys auf ihren ursprünglichen Werten, wodurch der Medienstrom direkt zwischen beiden Seiten ausgetauscht wird.

In beiden Fällen hat der Angreifer volle Kontrolle über die Signalisierung und muss aus diesem Grund die Funktion, die der eines Proxys entspricht, übernehmen.

4.3.2 Manipulation des SDP-Bodys

Bei diesem Angriff werden Pakete nicht nur abgehört und in bestehende Dialoge eingefügt, sondern auch manipuliert. Um den RTP-Medienstrom, wie in Abbildung 4.5 dargestellt, zu einem beliebigen Host umzuleiten, muss im SDP-Attribut *c* die IP-Adresse und in *m* die Portnummer verändert werden. Der Angriff kann verteilt durchgeführt werden, d.h., der Angreifer modifiziert an einer Stelle die SIP-Pakete und leitet den RTP-Datenstrom an einen anderen Host um.

Da der Empfänger von RTP-Paketen aus dem Paketinhalt nicht darauf schließen kann, wohin sie weitergeleitet werden sollen, muss zwischen beiden Stellen eine Signalisierung erfolgen, um die Kontaktdaten, gegebenenfalls Änderungen der Sitzungsparameter mitzuteilen und das Gesprächsende bekannt zu geben. Der Empfänger der RTP-Pakete muss die Signalisierung beim Empfang zeitnah an die eigentlichen Empfänger weiterleiten, damit die Gesprächspartner möglichst wenig vom Angriff mitbekommen.

4.3.3 Implementation: sip-redirectrtp und rtpproxy

Der Angriff aus dem vorangegangenem Abschnitt wurde in zwei separaten Teilen implementiert. Auf der einen Seite fängt das Programm sip-redirectrtp[2] SIP-Pakete ab und manipuliert den SDP-Body so, dass RTP-Pakete an einen bestimmten Host gesendet

[2] `http://skora.net/voip/sip-redirectrtp`

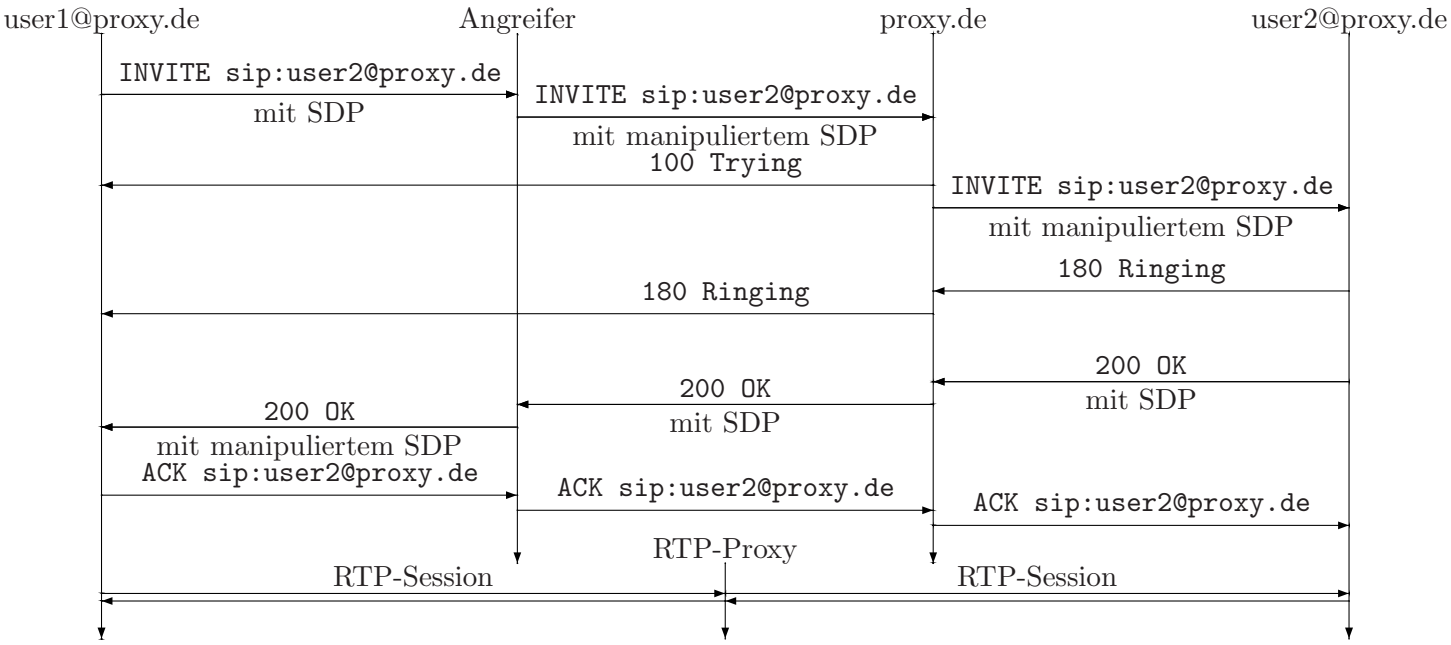


Abbildung 4.5: Umleitung eines RTP-Datenstroms

werden, dessen IP-Adresse und Signalisierungsport beim Programmstart mit dem Parameter p übergeben werden. Auf ihm läuft rtpproxy[3], nimmt die RTP-Pakete entgegen und leitet sie an das eigentliche Ziel weiter.

Die Manipulation von SIP-Paketen wird durch das Userspace-Queing des Linux-Kernels bewerkstelligt. Es ermöglicht durch Regeln des Paketfilters *IP-Tables* die Verarbeitung von IP-Paketen mittels Programmen aus dem Userspace, bevor sie gesendet oder dem dazugehörigem Socket zugestellt werden. Die Verarbeitung findet in sip-redirectrtp in der Prozedur `process_queue` statt.

Wird ein SIP-Invite empfangen, werden die IP-Adresse und der Port aus den SDP-Attributen c und m zwischengespeichert, durch die Adressen des RTP-Proxys ersetzt und das Paket zum Senden freigegeben. Gleiches geschieht beim Empfang der 200-Antwort, die ebenfalls einen SDP-Body enthält. Nachdem die manipulierte positive Antwort zum Senden freigegeben wurde, werden dem RTP-Proxy die IP-Adressen und Ports signalisiert, anhand denen der RTP-Proxy für jedes ankommende Paket entscheiden kann, an welche Adresse es weitergeleitet werden soll.

Zur Signalisierung zwischen sip-redirectrtp und rtpproxy wird ein rudimentäres und textbasiertes Protokoll verwendet. Die erste Zeile enthält den Pakettyp, gefolgt von mehreren Zeilen mit durch Gleichheitszeichen getrennten Attribut-Wert-Paaren. Grundsätzlich sendet sip-redirectrtp Anfragen, die von rtpproxy beantwortet werden. Es existieren folgende Pakettypen:

Register Diese Anfrage wird nach dem Start von sip-redirectrtp gesendet und enthält keine weiteren Parameter. Die Antwort von rtpproxy beinhaltet einen Parameter rtpport, in dem sich der Port befindet, an dem RTP-Pakete eintreffen sollen.

Announce Sobald sip-redirectrtp alle Adressdaten des Medienstroms zur Verfügung stehen, wird die Anfrage an rtpproxy gesendet. Sie enthält als Parameter die Callid und die beiden Adressen, an denen RTP-Pakete erwartet werden.

Discard Wird ein Gesprächsende signalisiert, teilt sip-redirectrtp das rtpproxy mit diesem Pakettyp mit. Als Parameter ist zur Identifikation die Call-Id des SIP-Dialogs enthalten.

Ok Dieser Pakettyp wird von rtpproxy zur Bestätigung der Anfragen verwendet. Bei der Beantwortung von Announce- und Discard-Anfragen wird als Parameter die Bezeichnung der beantworteten Anfrage und die Call-Id mitübertragen.

Wird eine Anfrage nicht innerhalb eines Zeitraums t beantwortet, werden bis zu n Wiederholungsversuche gestartet, nach denen sich t jeweils verdoppelt. Da die Übertragung insbesondere bei der Announce-Anfrage zeitkritisch ist, wurde für $t = 0,5$ gewählt. So

[3] `http://skora.net/voip/rtpproxy`

wird ein Gleichgewicht zwischen unnötigen Übertragungsversuchen und zu später Signalisierung bei Paketverlust hergestellt. Außerdem wurde $n = 5$ gewählt, was einen Abbruch nach $15,5$ Sekunden bedeutet. Diese relativ kurze Zeitspanne ist vertretbar, weil davon ausgegangen werden kann, dass der Benutzer bei einem nicht zustande kommenden Sprachdatenstrom bereits nach kurzer Zeit wieder auflegt.

Der RTP-Proxy enthält neben der Grundfunktionalität einige weitere Funktionen, über die die ankommenden RTP-Pakete manipuliert werden können, um RTP-Implementierungen zu testen. Der Parameter o vergrößert ein RTP-Paket um die angegebene Anzahl von zufällig generierten Bytes. Wird ein negativer Wert angegeben, wird das Paket entsprechend gekürzt. Mit dem Parameter x kann ein Anteil von Bytes angegeben werden, die in dem Paket durch zufällige Bytes ersetzt werden sollen. Die SSID wird mit i in jedem Paket auf einen neuen Zufallswert gesetzt.

Mit dem Parameter t können die Zeitstempel und Sequenznummern auf unterschiedliche Weise manipuliert werden. Wird als Parameter `rnd=p` übergeben, werden $100 \cdot p$ Prozent der Sequenznummern randomisiert. Bei `inc=p,s` werden $100 \cdot p$ Prozent der Sequenznummern um maximal s inkrementiert.

4.3.4 Senden eines Re-Invite

Ein Re-Invite ist ein Feature von SIP, das es erlaubt, die Parameter einer Session während ihres Bestehens zu verändern. Unter anderem kann so die Ziel-IP-Adresse und der UDP-Port für die RTP-Pakete verändert werden. Ein Einsatzgebiet dafür sind z.B. mobilen IP-Telefone, die per WLAN angebunden sind und deren IP-Adressen bei einem Handover wechseln [Vat05]. Der Unterschied zu einem Invite ist, dass ein Re-Invite die Dialogkennzeichnungen des bestehenden SIP-Dialogs hat. Um eine eventuell vorhandene Authentifizierung eines Proxys zu umgehen, kann ein Angreifer die Re-Invite-Anfrage so konstruieren, dass es erscheint, als ob die Anfrage den Signalisierungspfad bereits durchlaufen hat. Die Konstruktion ist der beim Spoofen von Bye-Paketen aus Abschnitt 4.4.2 sehr ähnlich.

Der Angriff ist vom Prinzip mit dem aus Abschnitt 4.3.2 vergleichbar, mit dem Unterschied, dass es für den Angreifer ausreicht, Pakete wie bei den DoS-Angriffen in einen bestehenden Dialog einzufügen. Manche Clients unterstützen keine Re-Invites und sind aus diesem Grund immun gegen diesen Angriff.

Ein weiterer Unterschied besteht darin, dass die Umleitung des Medienstroms erst nach dem Gesprächsaufbau folgt. D.h., der Angreifer verpasst die ersten RTP-Pakete. Findet der Angriff verteilt statt, kann der RTP-Proxy bereits vor der Umleitung des Medienstroms auf die Adressen der beiden Endpunkte eingestellt werden. Findet der Re-Invite nahtlos statt, wird ein Benutzer keine Pause bei der Umschaltung feststellen.

4.4 Beenden eines Gesprächs auf Teilen des Signalisierungspfads

In den bisher vorgestellten DoS-Angriffen ging es darum, Gespräche ganz zu beenden. Bei den in diesem Abschnitt vorgestellten Angriffen wird ein Gespräch nur auf Teilen des Signalisierungspfads beendet. Dadurch, dass ein Gespräch noch weitergeführt wird, obwohl einigen Signalisierungskomponenten ein Gesprächsende vorgetäuscht wurde, kann ein Netzbetreiber um Gesprächsgebühren betrogen werden. Im Gegensatz zu den anderen Angriffen befindet sich der Angreifer hier auf dem Signalisierungspfad und ist daher ein legitimer Teilnehmer des Signalisierungsablaufs.

4.4.1 Niedrige Max-Forwards-Werte

Bei diesem in [Sch05] vorgestellten Angriff wird der Max-Forwards-Header eines SIP-Pakets dafür genutzt, die Bye-Anfrage nur bis zu einem bestimmten Proxy des Signalisierungspfads gelangen zu lassen. Der Max-Forwards-Header soll, ähnlich dem TTL-Header in IP-Paketen, dafür sorgen, dass ein Paket, welches sich in einer Endlosschleife befindet, nach einer bestimmten Anzahl von Hops verworfen wird. Wird die Anzahl der maximal erlaubten Hops für eine Anfrage bewusst niedriger als die Anzahl der Hops des Signalisierungspfads gesetzt, wird der Proxy, bei dem Max-Forwards den Wert Null erreicht, die Anfrage verwerfen und einen Fehler „483 Too Many Hops“ an den Absender zurücksenden.

In Abbildung 4.6 ist dieser Angriff schematisch dargestellt. Der Anrufer sendet eine Bye-Anfrage mit einem Max-Forwards-Wert von 2. Auf dem Signalisierungspfad ist dieser Wert jeweils in der Klammer angegeben. Am dritten Proxy wird dieser Ablauf unterbrochen, da Max-Forwards hier den Wert 0 erreicht. Der RTP-Datenstrom zwischen beiden Gesprächsteilnehmern bleibt jedoch erhalten, weil die Bye-Anfrage den angerufenen Teilnehmer nicht erreicht.

Um diesen Angriff erfolgreich durchzuführen, müssen alle SIP-Implementierungen auf dem Signalisierungspfad den Dialog beim Empfang einer Bye-Anfrage beenden, ohne die Antwort der im Pfad folgenden Proxys abzuwarten. Des Weiteren kann der Angriff unterbunden werden, wenn Anfragen mit zu niedrigen Max-Forwards-Headern nicht akzeptiert werden.

4.4.2 Spoofen von BYE-Paketen

Der vorangegangene Angriff ist zwar einfach zu realisieren, kann aber problemlos durch Proxys auf dem Signalisierungspfad erkannt werden und durch eine korrekte Implementierung einer Bye-State-Machine unterbunden werden. SIP ermöglicht es einem Endpunkt A, Anfragen so zu konstruieren, dass dem Proxy des anderen Endpunktes B eine authentische Anfrage von B vorgetäuscht werden kann. Auf das Beispiel aus Abbil-

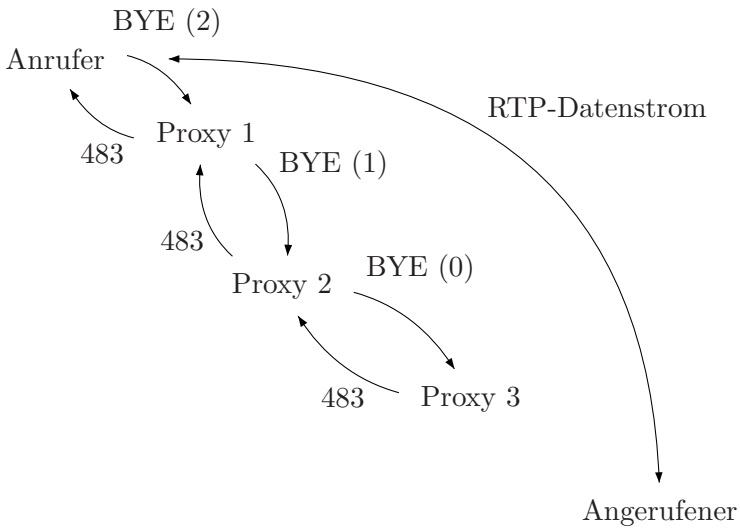


Abbildung 4.6: Partielles Beenden eines Gesprächs durch niedrige Max-Forwards-Werte

dung 4.6 bezogen bedeutet das, dass der Anrufer dem Proxy 3 eine vom Angerufenen gesendete Anfrage vortäuschen kann. Durch den im Folgenden konkretisierten Angriff kann dem Signalisierungspfad mittels eines gefälschten Bye-Pakets ein authentisches Gesprächsende vorgetäuscht werden, welches auf Anwendungsebene nicht von einem echten Gesprächsende zu unterscheiden ist.

Der Ablauf des Angriffs ist in Abbildung 4.7 dargestellt. Zunächst erfolgt ein vollständiger Rufaufbau, während dessen der Angreifer, hier der Anrufer, die benötigten Daten für die anschließende Konstruktion der Bye-Anfrage sammelt. Der Angreifer ist einer der beiden Endpunkte und damit ein legitimer Teilnehmer des Signalisierungsablaufs. Deshalb kann er die Informationen aus den SIP-Paketen, die während eines Rufaufbaus ausgetauscht werden, gewinnen und muss keine fremden Verbindungen abhören oder Pakete manipulieren. Prinzipiell kann der Angriff durch modifizierte Client-Software durchgeführt werden. Die Informationen für die Konstruktion der Bye-Anfrage können dabei aus den Zustandsinformationen des User-Agents gewonnen werden. Die Konstruktionsregeln für eine clientunabhängige Implementierung des Angriffs sind in Tabelle 4.2 zusammengefasst. Die Regeln sind so zusammengestellt, dass möglichst viele Informationen aus dem ACK-Paket bezogen werden, um den Speicheraufwand zu minimieren. Die Tabelle ist in zwei Abschnitte unterteilt. Im oberen Teil sind die Regeln für die Zusammenstellung der SIP-Anfrage, im unteren Teil die für die IP- und UDP-Header aufgelistet. Die linke Spalte gibt den Header oder eine Teilinformation eines Headers an,

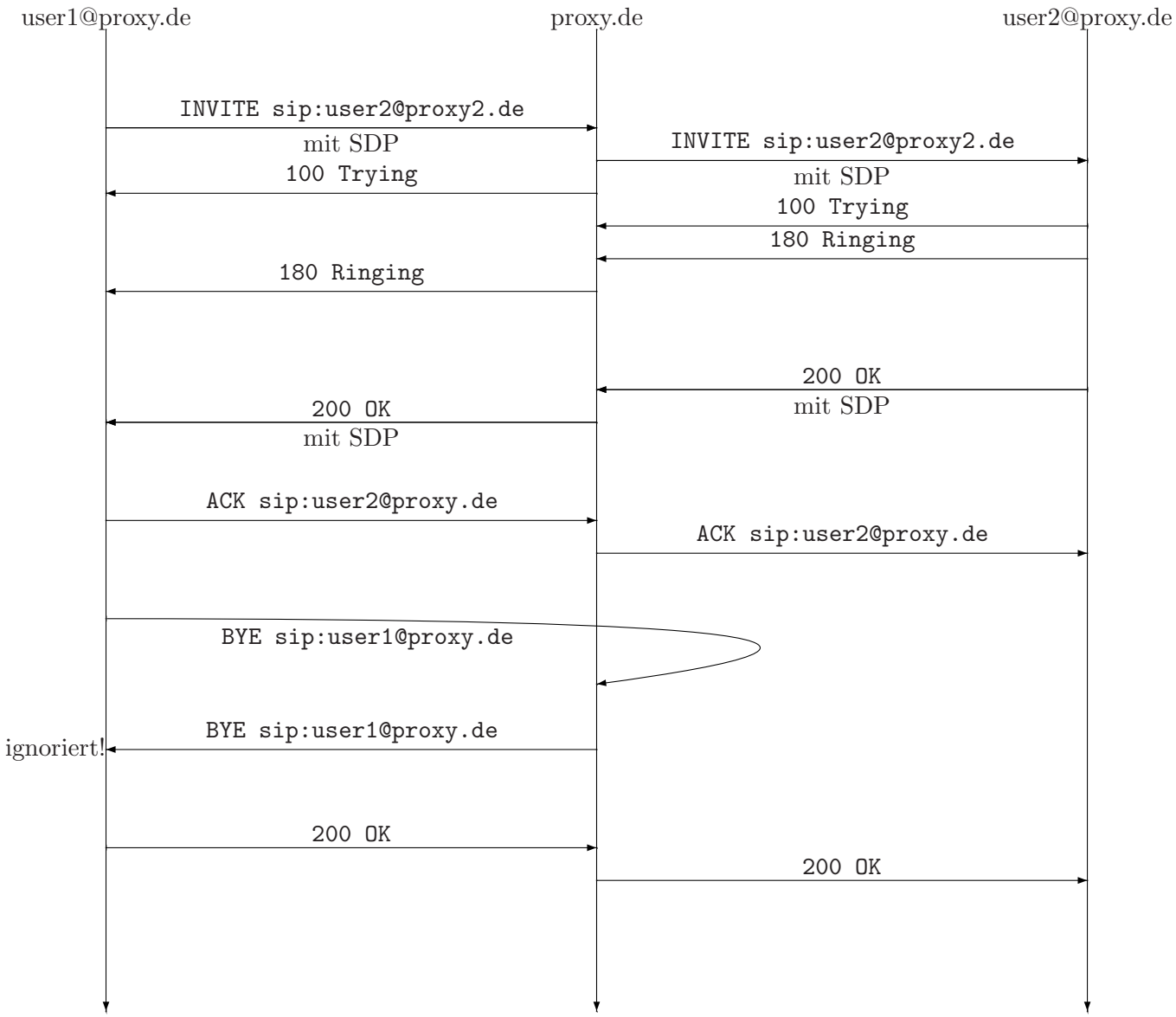


Abbildung 4.7: Injektion eines gespooftes BYE-Pakets auf dem Signalisierungspfad

Header	Inhalt wird übernommen aus
Anfrage-URL	Contact-Header der ACK-Anfrage
To mit Tag	From aus ACK
From mit Tag	To aus ACK
Contact	Aus 200-Antwort
Via (IP)	Aus Contact-Header der 200-Antwort
Via (Branch)	Zufällig generiert
Route	Umgedrehte ACK-Route
Route (ftag)	To-Tag aus ACK
CSeq	n BYE
Content-Length	0
User-Agent	Aus 200-Antwort
Quell-IP-Adresse/Port	IP aus Contact aus 200-Antwort
Ziel-IP-Adresse/Port	Letzter Hop der generierten Route

Tabelle 4.2: Konstruktion einer gefälschten BYE-Anfrage

zu der in der rechten Spalte jeweils angegeben wird, aus welchem Paket die Information entnommen wird.

Da eine Bye-Anfrage von der Gegenseite des Anrufers simuliert werden soll und der Contact-Header die URL für spätere Anfragen enthält, wird die URL aus diesem Header für diesen Zweck verwendet. Zusätzlich müssen die Inhalte der From- und To-Header aus der Ack-Anfrage vertauscht werden. Der Contact-Header wird von der Gegenseite nur in der Antwort auf die Invite-Anfrage gesendet und muss daher aus ihr übernommen werden. Die IP-Adresse für den Via-Header wird ebenfalls aus diesem Contact-Header übernommen. Der Branch-Parameter ist der einzige Teil der SIP-Anfrage, die nicht aus den vorhergehenden Paketen des Dialogs stammt. Da der Branch bei einer echten Anfrage vom Absender generiert wird, wird er auch bei diesem Angriff zufällig generiert. Für die Route-Header wird die Heuristik so verwendet, dass die Signalisierungsroute vom Anrufer zum Angerufenen in den meisten Fällen der umgedrehten Route in die andere Richtung entspricht, die der Ack-Anfrage entnommen werden kann. Die Sequenznummer im CSeq-Header kann auf einen beliebigen Wert gesetzt werden, falls von der Gegenseite noch keine Anfragen gesendet wurden. Ansonsten sollte die Sequenznummer über der der letzten Anfrage des anderen Endpunkts liegen. Der User-Agent-Header wird für eine korrekte Bye-Anfrage zwar nicht benötigt, wird aber häufig verwendet und hier mit aufgenommen, um möglichst wenig Differenzen zwischen echten und gefälschten Bye-Anfragen entstehen zu lassen.

Wird der Angriff in einem Client implementiert, dann kann die 200-Antwort so wie bei einem normalen Bye von der Gegenseite gesendet werden, mit dem Unterschied, dass der interne Zustand des Clients unverändert bleibt und das Gespräch fortgeführt wird.

Wird die Antwort außerhalb des Clients generiert, kann das vorher konstruierte Bye-Paket weitgehend übernommen werden. Nur die Zeile mir der Anforderung muss durch „SIP/2.0 200 OK“ ersetzt und der Contact-Header angepasst werden. Alle Route und Record-Route-Header sind zu entfernen.

Eine Variante des Angriffs besteht darin, dass die Bye-Anfrage nicht am Ende des Signalisierungspfades, zwischen einem Endpunkt und dem dazugehörigen Proxy, sondern in der Mitte, zwischen zwei Proxys, eingefügt wird. In diesem Fall muss die Route um die umgangenen Proxys gekürzt werden, da die Anfrage ansonsten über sie geroutet wird. Bei dem Via-Header bestehen zwei Möglichkeiten. Einerseits kann die Antwort durch Angabe der übergangenen Proxys über diese geroutet werden. Anderseits können sie weggelassen werden und statt dessen nur der Proxy als einziger Via-Hop angegeben werden, an den die Anfrage gesendet wird. Die Antwort auf das Bye wird nur so weit geroutet, wie es durch die Via-Header angegeben ist.

Die Quell-Adresse und der Quell-Port für den IP- und UDP-Header können nicht so wie in der obigen Tabelle beschrieben übernommen werden, weil der Contact-Header immer die Adresse eines Endpunkts angibt, die Bye-Anfrage bei dieser Variante des Angriffs aber scheinbar von einem Proxy ausgehen soll. Aus diesem Grund muss die Quelladresse aus dem Route-Header übernommen werden, der bei der Konstruktion der Route entfernt wird, denn er repräsentiert den Proxy, der vor dem Proxy liegt, an den die Anfrage gesendet wird. Hier ergibt sich das Problem, dass aufgrund des Loose-Source-Routing zwischen den beiden Proxys noch weitere Elemente auf dem Signalisierungspfad liegen können, die für den Angreifer nicht sichtbar sind und die beim normalen Signalisierungsablauf nicht miteinander kommunizieren.

4.5 Allgemeine Angriffe

In diesem Abschnitt wird ein Überblick über Angriffe gegeben, die zwar nicht VoIP-spezifisch sind, den VoIP-Betrieb aber ebenfalls einschränken können.

DNS-Angriffe Angriffe auf das DNS-Protokoll haben das Ziel, anfragenden Clients falsche IP-Adressen zu liefern. Dadurch kann ein Denial of Service verursacht werden, indem eine beliebige Adresse zurückgegeben wird und Anfragen an den vermeintlichen Server ins Leere laufen.

DNS-Spoofing DNS-Spoofing kann durch kompromittierte Nameserver oder durch Packet Injection erfolgen. Dazu muss der Angreifer schneller Antworten als der angefragte DNS-Server.

DNS Cache-Poisoning Eine weitere Technik zur Manipulation von Namensinformationen, das DNS-Cache-Poisoning, nutzt eine Schwäche in DNS-Implementierungen aus. Verarbeitet der DNS-Client Antworten ohne vorherige Anfrage und spei-

chert diese zur Effizienzsteigerung in einem Cache, kann ein Angreifer diese Schwachstelle zum Einschleusen falscher Datensätze nutzen.

DHCP-Angriffe DHCP dient in vielen lokalen Netzwerken zur automatischen Konfiguration der Clients. DHCP-Pakete werden nicht authentifiziert und Anfragen von Clients werden per Broadcast an alle Teilnehmer einer Broadcast-Domäne gesendet. Dadurch eröffnet dieses Protokoll einige Angriffsmöglichkeiten.

Rogue Server Nachdem ein Client eine Anfrage gesendet hat, bestimmt das erste passende Antwortpaket die Konfiguration. Sendet ein Angreifer eine gefälschte Antwort, die den anfragenden Client früher erreicht als die Antwort des DHCP-Servers, kann er die Konfigurationsparameter wie IP und DNS-Server frei bestimmen.

Starvation Fordert ein Angreifer mit ständig wechselnder MAC-Adresse IP-Adressen an, kann das dazu führen, dass alle vom DHCP-Server verwaltete Adressen aufgebraucht werden und damit keine Adressen mehr für legitime Clients zur Verfügung stehen.

ICMP-Angriffe Das Internet Control Message Protocol (ICMP) [Pos81] setzt auf IP auf und ist dafür vorgesehen, anderen Hosts Kontrollnachrichten mitzuteilen. Es bietet keine Authentifizierung der übertragenen Nachrichten und enthält auch keine Sequenznummern, die ein Angreifer erraten muss. Aus diesem Grund sind ICMP-Nachrichten sehr einfach zu fälschen und können, falls der angegriffene Host entsprechend auf die Nachrichten reagiert, für Angriffe genutzt werden.

Source Quench Die ICMP-Nachricht *Source Quench* soll einem sendenden Host mitteilen, dass ein Paket auf dem Weg zum Ziel verworfen wurde und die Sendegeschwindigkeit gesenkt werden sollte. Das könnte dafür verwendet werden, um im RTP-Datenstrom Aussetzer zu produzieren.

ICMP Redirect Ein weiterer ICMP-Nachrichtentyp *Redirect* war ursprünglich dafür vorgesehen, Routern die Möglichkeit zu geben, anderen Hosts eine bessere Route mitzuteilen. Das kann ein Angreifer bei einem IP-Stack, der diese Nachricht verarbeitet, dazu nutzen, den Datenverkehr auf ein anderes Interface umzuleiten und dadurch einen DoS-Angriff durchzuführen.

Angriffe auf den IP-Stack Angriffe auf den IP-Stack sind nicht mehr so stark verbreitet, wie sie es Ende der 90er Jahre noch waren, da die Implementierungen mittlerweile ausgereift sind. Nicht jeder Hersteller von VoIP-Hardware verwendet aber einen fertigen IP-Stack. Darauf lassen OS-Fingerprints [Fyo98] schließen, anhand deren sich bestimmte Geräte eindeutig erkennen lassen. In so einem Fall ist es möglich, dass alte Fehler wieder auftauchen und sich für DoS-Angriffe ausnutzen lassen.

Fehlerhafte Fragmentierung von IP-Paketen und Teardrop Beim Teardrop-Angriff werden Fragmente von IP-Paketen so konstruiert, dass sie sich überlappen. Fehler bei der Reassemblierung des Pakets führten dazu, dass Betriebssysteme wie Windows oder Linux zum Absturz gebracht werden konnten.

Der IP-Stack muss diese Fragmente eine Zeit lang speichern, um sie beim Eintreffen des letzten Fragments zu einem Paket reassemblieren zu können. Eine weitere Möglichkeit, Ressourcen im IP-Stack zu belegen, besteht darin, nicht alle Fragmente zu senden. Wird der belegte Speicher vom IP-Stack nach einer bestimmten Zeit nicht freigegeben, kann ein Angreifer damit Speicher belegen.

Ping of Death Mit Ping of Death wird die Tatsache genutzt, dass ein Paket durch Fragmentierung so konstruiert werden kann, dass es nach dem Reassemblieren größer ist als die maximal zulässige Größe eines IP-Pakets, die 65.535 Bytes beträgt. Im IP-Stack kann dies einen Buffer Overflow auslösen, der zum Absturz des Systems führen kann.

Land-Angriff Bei diesem Angriff wird ein TCP-Paket mit gesetztem SYN-Flag an einen offenen TCP-Port gesendet. Die Quell- und Zieladresse im IP-Header sowie die Ports im TCP-Header sind jeweils identisch. Beantwortet der so angegriffene IP-Stack das Paket mit einem TCP-Paket mit gesetztem SYN- und Ack-Flag an sich selbst und betrachtet diese Antwort als neuen Versuch, eine TCP-Verbindung aufzubauen, kann das zu einer Endlosschleife führen. Durch die Endlosschleife wird das System stark ausgelastet und Anfragen können nur noch sehr langsam oder gar nicht beantwortet werden.

ARP-Spoofing ARP-Spoofing kann dazu genutzt werden, Man-in-the-Middle-Angriffe vorzubereiten. Zur Zuordnung von IP-Adressen zu MAC-Adressen der Netzwerkhardware wird das Adress Resolution Protokoll (ARP) verwendet. Ein Client, der bei einem Ethernet über das IP-Protokoll mit einem anderen Client kommunizieren möchte, fragt zunächst mit einer ARP-Amfrage, die die IP-Adresse des Ziels enthält, dessen MAC-Adresse an. Die ARP-Antwort enthält die dafür benötigte MAC-Adresse, die dann zur Konstruktion von Ethernet-Frames verwendet wird. Um unnötige ARP-Anfragen zu unterbinden, werden die IP-MAC-Zuordnungen für eine bestimmte Zeit in einem Cache gespeichert.

Gelingt es einem Angreifer, die Antwort auf die ARP-Anfrage als Erster zu senden, kann dieser seine MAC-Adresse als Ziel angeben und empfängt fortan alle Pakete, die für den eigentlichen Zielrechner bestimmt waren. Ähnlich wie beim DNS-Cache-Poisoning gibt es ARP-Implementierungen, die ARP-Antworten ohne vorherige Anfragen verarbeiten und es einem Angreifer ermöglichen, beliebige Einträge in ARP-Caches zu manipulieren.

Ein Switch ordnet intern jeder MAC-Adresse einen Port zu, um Unicast-Pakete di-

rekt an den entsprechenden Teilnehmer zustellen zu können. Die Informationen für diese Zuordnung werden aus ARP-Paketen gewonnen. Durch die oben beschriebenen Angriffe kann ein Switch so beeinflusst werden, dass Ethernet-Frames an den Port des Angreifers zustellen.

Um unbemerkt zu bleiben, muss der Angreifer die ankommenden Pakete an das eigentliche Ziel weiterleiten und dafür die MAC-Adresse des Ziels kennen.

MAC-Spoofing Switches ordnen jeder MAC-Adresse einen Port zu, damit Pakete nicht an alle angeschlossenen Client gesendet werden müssen. Sendet ein Angreifer ein Ethernet-Paket mit einer gefälschten MAC-Adresse, kann die Zuordnung je nach Switch überschrieben werden. Das führt dazu, dass die Pakete zum Angreifer durchgeleitet werden und der eigentliche Ziel-Host nicht mehr erreichbar ist. Erst wenn der Ziel-Host wieder ein Paket sendet, kann der Switch die ursprüngliche Zuordnung wieder herstellen. Dadurch können vor allem bei RTP Paketverluste herbei geführt werden, die die Sprachqualität so stark einschränken, dass keine Kommunikation mehr möglich ist.

MAC-Flooding Der Speicherplatz, in dem der Switch die zu Ports zugehörigen MAC-Adressen speichert, ist nur begrenzt. Läuft dieser Speicher über, kann ein Switch keine Zuordnung mehr vornehmen und schaltet in einen Failsafe-Modus, in dem er sich wie ein Hub verhält und grundsätzlich alle ankommenden Pakete an alle Ports verteilt. Das kann dadurch erreicht werden, dass ein Angreifer viele Pakete mit unterschiedlichen MAC-Adressen sendet, bis es letztendlich zum Überlauf kommt.

Angriffe auf Routing-Protokolle Routing-Protokolle bieten ebenfalls Angriffspunkte. Das gilt besonders, wenn Sicherheitsmechanismen wie die Authentifizierung nicht genutzt werden, vom Protokoll nur in schwacher Form, wie die Klartextpasswörter in RIPv2, oder gar nicht, wie in der ersten Version von RIP, angeboten werden. Dadurch kann ein Angreifer Routen so manipulieren, dass der Datenverkehr zu einem beliebigen Host umgeleitet wird.

Details zu Angriffsmöglichkeiten auf die Protokolle RIP, OSPF, IGRP und EGRP werden in [rou01] vorgestellt. In [Bel89] werden die Protokolle RIP und EGP behandelt.

IP-Source-Routing Ein Angriffspunkt bei IP ist das *Source Routing*, von dem es zwei Varianten gibt. Beim strikten Routing muss das Paket die Hosts in der angegebenen Reihenfolge durchlaufen, ohne durch Hosts geroutet zu werden, die sich nicht in der Liste befinden. Beim *Loose Routing* wird letztere Anforderung gelockert und auf der Route dürfen sich auch weitere Hosts befinden. Ein Angreifer kann dies je nach Implementation des Angegriffenen Ziel-IP-Stacks dazu nutzen, den Datenverkehr über eine beliebigen Host umzuleiten. Möchte der Angreifer den von B initiierten IP-Datenverkehr der Hosts mit den IP-Adressen A und B umleiten,

muss er gespoofte Pakete mit der Quelladresse von A und einer Route, in der mindestens die Angreifer-IP enthalten ist, an B senden. Je nach Implementation des IP-Stacks von B wird beim Senden von Paketen zu A die umgedrehte Route mit im IP-Header gesendet und die Pakete somit über den Angreifer umgeleitet.

5 Sicherheitsmaßnahmen

Um Angriffen auf VoIP entgegenzuwirken, wurden einige Protokolle erweitert und neue spezifiziert, die den Datenverkehr mit unterschiedlichen Zielsetzungen absichern. Die Absicherung hat bei VoIP mehrere Aufgabengebiete. Ein Client muss sich für Abrechnungszwecke und zur Verhinderung eines Identitätsdiebstahls gegenüber einem Proxy authentifizieren können, was durch die in Abschnitt 5.1 vorgestellte HTTP-Digest-Authentifizierung geschehen kann. Um auch die Integrität der Signalisierungsnachrichten zu sichern, wurden Erweiterungen der Digest-Authentifizierung beschrieben, die in Abschnitt 5.2 vorgestellt werden. Die in Abschnitt 5.3 bis 5.5 vorgestellten Verfahren und Protokolle ermöglichen eine verschlüsselte Übertragung der Signalisierungspakete und können zusätzlich die Integrität der Pakete absichern. Die davor beschriebenen Verfahren wurden bereits in anderen Bereichen als VoIP verwendet und haben sich im Alltagseinsatz bewährt.

Eine weitere Aufgabe ist der Austausch von kryptographischen Schlüsseln zwischen zwei Endpunkten, um eine Verschlüsselung des Medienstroms zu ermöglichen. Das kann durch die Übertragung der Schlüssel im SDP-Body von SIP-Paketen geschehen (Abschnitt 5.6.1 und 5.6.2), ist allerdings auf eine weitere Absicherung der Signalisierung angewiesen. Um diese Abhängigkeiten zu umgehen, existieren spezielle Schlüsseltauschprotokolle wie Internet Key Exchange (IKE). Es zeigt sich allerdings, dass die bisher in der IP-Kommunikation verwendeten Protokolle nicht mehr den Ansprüchen von VoIP genügen und Bedarf für neue Protokolle besteht. Multimedia Internet Keying (MIKEY) ist ein Schlüsseltauschprotokoll, das speziell für Anwendungen wie VoIP konzipiert wurde, und wird im Abschnitt 5.6.3 vorgestellt.

Den Abschluss dieses Kapitels bilden generische Sicherheitsmaßnahmen wie IPsec und andere VPN-Lösungen, die zur Absicherung des gesamten IP-Datenverkehrs verwendet werden können, und eine Analyse der Sicherungsmaßnahmen, in der die hier vorgestellten Verfahren gegenübergestellt und VoIP-spezifische Vor- und Nachteile erörtert werden.

5.1 Digest-Authentifizierung

SIP sieht eine einfache Authentifizierung mit dem HTTP-Digest-Verfahren [FHBH+99] vor, die zur Zeit im Gegensatz zu TLS und S/MIME in den Implementationen weit verbreitet ist. Es ist ein Challenge-Response-Verfahren, bei dem der UAC einen Hashwert über ein gemeinsames Geheimnis, einen Challenge (den sogenannten Nonce) und einige

qop	Berechnung der Antwort
keiner	$H(A_1 : nonce : A_2)$ $A_1 = H(username : realm : secret)$ $A_2 = H(method : url)$
auth	$H(A_1 : nonce : noncecount : cnonce : qop : A_2)$ $A_1 = H(username : realm : secret)$ $A_2 = H(method : url)$
auth-int	$H(A1 : nonce : noncecount : cnonce : qop : A2)$ $A_1 = H(username : realm : secret)$ $A_2 = H(method : url : H(body))$

Tabelle 5.1: Digest-Authentifizierung: gängige qop-Werte

andere Inhalte des SIP-Pakets berechnet wird. Die Berechnung des Hashwerts ist in Tabelle 5.1 dargestellt.

Die Digest-Authentifizierung bietet zusätzlich zur Authentifizierung einen Schutz der Integrität des Pakets, die mit dem *qop*-Parameter (Quality of Protection) in verschiedenen Stufen regelbar ist. In Tabelle 5.1 sind die Berechnungsregeln für die in RFC 2617 [FHBH+99] spezifizierten *qop*-Werte aufgelistet. H stellt dabei die verwendete Hashfunktion dar. Üblicherweise wird hier MD5 verwendet. Wird statt dessen eine Hashfunktion $H_k(M)$ eingesetzt, die einen Schlüssel k erwartet, wird bei der Berechnung $H(A_1 : M)$ das A_1 als Schlüssel verwendet und $H_{A_1}(M)$ berechnet. Ist kein *qop* angegeben, wird die Digest-Authentifizierung rückwärtskompatibel zu RFC 2069 [FHBH+97] durchgeführt. Dieser Variante fehlen unter anderem die Parameter *noncecount* und *cnonce*, die zur Abwehr von Replay-Angriffen dienen. Die beiden Werte sind in $qop = auth$ gegeben. Bei beiden Varianten wird nur die Integrität der Anfragezeile gesichert, während bei $qop =$ auth-int der Hash über den Body in die Berechnung mit einfließt und damit die Integrität des Bodys überprüft werden kann.

Wichtig bei der Implementierung ist die Konstruktion des nonce-Parameters. Um einen Schutz vor Replay-Angriffen zu bieten, muss dieser Wert zufällig generiert werden, und zusätzlich sollten die IP-Adresse und die Zeit eincodiert werden. Ein Problem der Integritätsüberprüfung ist hier, dass es in manchen Szenarien legitim ist, dass ein Proxy den Body eines SIP-Pakets modifizieren muss. Das ist z.B. in NAT-Umgebungen der Fall, in denen der Router die Kontaktdaten für den RTP-Datenstrom im SDP-Body anpasst und ein Zustandekommen des Gesprächs dadurch möglich macht oder zumindest erleichtert.

Bei dieser Authentifizierungsmethode entscheidet der UAS, ob eine Anfrage authentifiziert wird oder nicht. Eine Authentifizierung der Antworten vom UAS ist zwar bei Digest vorgesehen, aber nicht vorgeschrieben. Der Angriff aus Abschnitt 4.1.1 basiert auf der Tatsache, dass Antworten durch das Fehlen einer Authentifizierung durch einen Angreifer in einen SIP-Dialog eingefügt werden können.

Für die Anfragen Ack und Cancel ist die Authentifizierung durch HTTP-Digest nicht vorgesehen und wäre aufgrund der hohen Ansprüche an eine schnelle Verarbeitung durch die Anwendung auch nicht praktikabel. Die Ack-Anfrage stellt dabei kein so großes Sicherheitsrisiko dar wie die Cancel-Anfrage, da durch ein Ack nur der Three-Way-Handshake, der in Abschnitt 3.2.4 vorgestellt wurde, komplettiert wird. Ein Angreifer kann durch das Senden gefälschter Ack-Anfragen höchstens eine wiederholte Sendung einer verlorengegangenen Ok-Antwort, die auch die Sitzungsbeschreibung der Gegenseite enthält, unterbinden. Dagegen kann ein aktiver Angreifer durch eine Cancel-Anfrage einen Rufaufbau abbrechen und damit eine DoS-Attacke durchführen. Dieser Angriff wurde bereits in Abschnitt 4.1.2 beschrieben.

Je nachdem, für welche Anfragen der UAS keine Authentifizierung erfordert, sind weitere Angriffe möglich. Offensichtlich ist der Angriff bei Register-Anfragen. Werden diese nicht authentifiziert, kann sich jeder unter einer fremden Identität bei einem SIP-Registrar anmelden und auf dessen Kosten telefonieren. Außerdem können bestehende Registrierungen aufgehoben werden, so dass der Inhaber des Accounts keine Gespräche mehr entgegennehmen kann.

Wird vom SIP-Proxy keine Authentifikation für Invite-Anfragen erzwungen, kann ein Angreifer auf Kosten eines beim SIP-Proxy registrierten Benutzers Telefonate führen. Erfolgt die Authentifikation durch die IP-Adresse des Clients, muss der Angreifer die authentifizierten Register-Anfragen des Clients mit der eigenen IP manipulieren, um diese schwache Form der Authentifizierung zu umgehen [PS]. Wird bei Bye-Anfragen keine Authentifizierung durchgeführt, kann ein Angreifer durch Einfügen eines gefälschten Bye-Pakets einen Gesprächsabbruch erzwingen. Selbst bei durch Proxy-Server authentifizierten Bye-Anfragen kann ein Angreifer versuchen, die Authentifizierung zu umgehen, indem er die gefälschte Anfrage, statt an den nächstliegenden Proxy an ein Element auf dem Signalisierungspfad sendet, das sich weiter hinten befindet. Insbesondere der Client des Gesprächspartners am Ende des Pfades ist hier interessant, da ein Client üblicherweise keine Authentifizierung seitens des Proxys anfordert, obwohl dies bei einem bestehendem Shared Secret für die normale Client-Proxy-Authentifizierung möglich wäre.

Auch Erweiterungen von SIP, wie die Subscribe-Anfrage, können für Angriffe genutzt werden. So kann sich ein Angreifer für Ereignisse anmelden, für die er nicht autorisiert ist oder durch massives Registrieren von Subscriptions unter fremden Kontaktdaten einen DoS-Angriff vorbereiten, der beim Auslösen der Ereignisse zu einer Flutung des Opfers führt.

5.2 Erweiterungen von HTTP-Digest

HTTP-Digest ist zwar für die Authentifizierung ausreichend, die Integrität der SIP-Header wird allerdings bei *qop* = auth-int nicht überprüft und können daher von einem

Angreifer manipuliert werden. Aus diesem Grund wurden für SIP Erweiterungen von HTTP-Digest vorgeschlagen, die diese Lücke schließen sollen. Diese Erweiterungen werden im Folgenden vorgestellt.

5.2.1 Header-Listen und Authentifizierung des gesamten Pakets

Eine Erweiterung von HTTP-Digest für SIP [US02] erlaubt es, mit zwei zusätzlichen qop-Optionen, die Header mit in die Berechnung des Digests einzubeziehen. Bei *qop* = auth-extd-int wird A_2 ähnlich zu *qop* = auth-int berechnet. Der einzige Unterschied besteht darin, dass $H(body)$ durch $H(m)$ ersetzt wird, wobei m die gesamte Nachricht ohne die Zeile mit der Antwort der Authentifizierung enthält. Damit auch eine Authentifizierung mit einer feineren Auswahl der zu hashenden Elemente ermöglicht wird, gibt es zusätzlich *qop* = auth-hdr-int. Bei dieser Variante werden die Header, die mit in die Berechnung von A_2 einfließen sollen, in einem zusätzlichen Parameter *header-list* angegeben. Der Hash wird anschließend über eine kanonische Form der Header und dem Body berechnet. Bei der Antwort kann der Client optional die Headerauswahl erweitern, um weitere Header zu schützen.

Zusätzlich wird noch eine erweiterte Form der Authentifizierung von Antwortpaketen spezifiziert, die den Antwortcode mit in die Berechnung der Antwort einbezieht und damit dessen Integrität sichert. Des Weiteren wird eine Erweiterung des *nonce* um eine Codierung der verwendeten Schutzmechanismen, wie qop oder der verwendeten Hashfunktion vorgeschlagen, durch die Downgrade-Angriffe verhindert werden, bei denen der Angreifer vorgeben könnte, in welchem Umfang die Integrität überprüft wird.

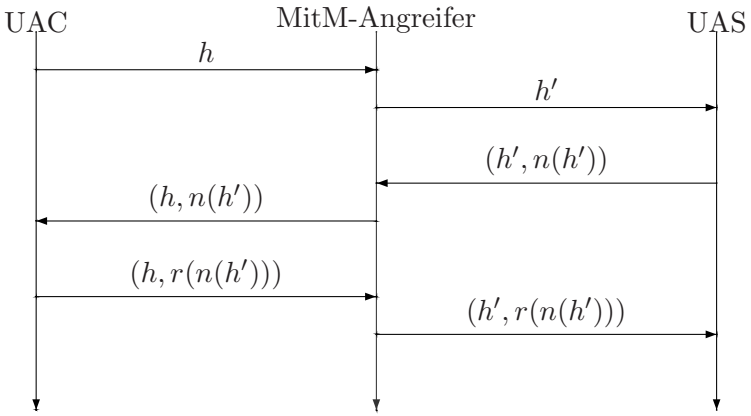
5.2.2 Predictive Nonces

Einen anderen Ansatz zur Sicherung der Integrität von SIP-Paketen wird in [Ros01] verfolgt. Im Gegensatz zur vorherigen Erweiterung benötigt dieses Verfahren keine Veränderungen am Client. Das Verfahren basiert darauf, dass sich bestimmte Header zwischen den Anfragen nicht verändern. Der UAS, der eine Authentifizierung durchführen möchte, muss die Header bestimmen[1] und diese zur Berechnung des Nonce heranziehen. Das kann beispielsweise durch eine Hashfunktion geschehen, die die Header in kanonischer Form mit einbezieht. Um wiederholende Nonces zu vermeiden, sollte eine zufällige Komponente in dessen Berechnung mit einfließen.

Beim Verifizieren der Antwort berechnet der UAS nach dem gleichen Schema und mit den gleichen Headern einen Nonce aus dem authentifizierten Paket. Nur wenn die Header zu den vorhergehenden Paketen übereinstimmen, wird der Nonce und damit die Antwort übereinstimmen.

Im Gegensatz zu den Ausführungen in [Ros01] ist dieses Verfahren nicht robust gegen Man-in-the-Middle-Angriffe. In Abbildung 5.1 ist ein solcher Angriff dargestellt. Um nur

[1] Besser: Voraussagen, daher *predictive* nonce



Zeichen	Bedeutung
h	Headern aus Anfrage des UAC
h'	Manipulierte UAC-Header
$n(h)$	Nonce in Abhängigkeit von Headern h
$r(n)$	Antwort in Abhängigkeit von Nonce n

Abbildung 5.1: MitM bei Predictive Nonces

die nötigsten Informationen der SIP-Pakete angeben zu müssen, wurden zur Darstellung Zweiertupel verwendet. Die erste Komponente gibt den Header an, die zweite die übertragene Digest-Information, die ein Nonce oder eine Antwort sein kann. Ein Nonce hängt bei diesem Verfahren von einer Auswahl von Headern ab und wird als Funktion $n(h)$ mit den Headern h als Parameter angegeben. Analog ist $r(n)$ die Antwort auf eine Authentifizierungsaufforderung mit dem Nonce n.

Der Angreifer manipuliert die erste, unauthentifizierte Anfrage mit den Headern h so, dass der UAS einen Nonce $n(h')$ für die modifizierten Header h' berechnet. Da der UAC bei diesem Verfahren nicht an der Überprüfung der Integrität beteiligt ist, wird dieser die Authentifizierung durchführen und die Antwort $r(n(h'))$ senden, auch wenn die SIP-Header an dieser Stelle nicht zum Nonce passen. Bevor diese Antwort den UAS erreicht, werden die Header vom Angreifer wieder so wie im ersten Schritt manipuliert und $(h', r(n(h')))$ weitergeleitet. Da die Integritätsbedingung Damit bieten Predictive Nonces bei der vorgeschlagenen Berechnung des Nonce nur noch einen Schutz gegen Replay-Angriffe, der jedoch durch Einweg-Nonces mit der einfachen Digest-Authentifizierung realisiert werden kann.

5.3 Transport Layer Security (TLS)

Das in RFC 2246 spezifizierte TLS-Protokoll (Transport Layer Security) [DA99] bietet einen Kommunikationskanal, der die übertragenen Daten verschlüsselt überträgt und die Integrität der Daten absichert. Das Protokoll besteht aus mehreren Teilen. Zum einen dem Record Layer, das die eigentliche Übertragung absichert, und den darauf basierenden Protokollen, wie dem Handshake Protocol oder dem zum Kapseln der Nutzdaten verwendetem Application Data Protocol. Das Handshake Protocol wird verwendet, um Kommunikationsparameter wie den Verschlüsselungs- oder Hashalgorithmus und den symmetrischen Schlüssel auszuhandeln.

Im ersten Schritt sendet der Client ein *Client Hello*-Paket, das die unterstützten Kombinationen von Verschlüsselungsalgoritmen mit Modi, Schlüssellänge und Hashalgorithmen enthält. Der Server antwortet auf dieses Paket mit einem *Server Hello*, in dem eine der angebotenen Algorithmenkombinationen ausgewählt und optional das Zertifikat des Servers übertragen werden kann, das vom Client überprüft wird. Der serverseitige Teil des Schlüsseltauschs kann implizit durch die Übertragung des Zertifikats oder explizit durch eine spezielle Nachricht erfolgen. Je nach verwendeter Schlüsseltauschmethode wird der öffentliche RSA-Schlüssel oder die öffentlichen Diffie-Hellman-Parameter gesendet. Zusätzlich kann der Server auch eine Authentifizierung des Clients mit einem Zertifikat anfordern.

Nach dem optionalen Senden eines Client-Zertifikats und der Bestätigung dessen Authentizität durch den Server erfolgt der clientseitige Schlüsseltausch, der folgendermaßen erfolgt:

- Senden eines zufällig generierten Premaster Keys, der mit dem öffentlichen Schlüssel des Servers verschlüsselt ist.

- Senden des öffentlichen Diffie-Hellman-Parameters des Clients. Dieser Schritt kann auch implizit mit dem Senden des Client-Zertifikats erfolgen.

Der daraus resultierendem Premaster Key wird als Eingabe für eine pseudozufällige Funktion verwendet, deren Ausgabe den *Master Key* darstellt und als Schlüssel für die weitere symmetrische Verschlüsselung der Anwendungsdaten dient.

Abgeschlossen wird der Handshake über eine Nachricht, die bereits mit den ausgehandelten Parametern und Algorithmen verschlüsselt ist und unter anderem alle ausgetauschten Handshake-Nachrichten erneut enthält. Anhand dieser Nachricht können beide Seiten den korrekten Ablauf des Handshakes verifizieren.

Da TLS im Allgemeinen als sicheres Protokoll akzeptiert ist und bereits in vielen alltäglichen Situationen, wie z.B. dem HTTP-Datenverkehr beim Online-Banking, verwendet wird, kann davon ausgegangen werden, dass ein über TLS geführter Dialog im Sinne der Vertraulichkeit und Integrität abgesichert ist. Voraussetzung hierfür ist, dass

die Implementation korrekt ist und der Benutzer einen Angriff erkennen und entsprechend reagieren kann bzw. der Client so konfigurierbar oder implementiert ist, dass eine fehlerhafte Bedienung im Angriffsfall seitens des Benutzers ausgeschlossen ist.

Um eine Authentifizierung durchführen zu können, wird eine Public-Key Infrastruktur benötigt, welche mit einem administrativen Mehraufwand verbunden ist. Dafür muss eine Zertifizierungsstelle[2] eingerichtet werden, die Zertifikate erstellt und mit ihrem privaten Schlüssel signiert. Dies kann durch externe Firmen erfolgen, was mit Mehrkosten verbunden ist, oder intern in der jeweiligen Organisation durchgeführt werden. Die so erstellten Zertifikate müssen mit den zur Signierung verwendeten Root-Zertifikaten auf die Endgeräte verteilt werden. Ein Betrieb ohne PKI ist zum Beispiel durch selbst- oder unsignierte Zertifikate möglich, schränkt die Authentifizierung allerdings ein. Ein Client kann in diesem Fall nicht mehr nachweisen, dass die im Zertifikat angegebenen Daten mit denen aus der Realität übereinstimmen, weil sie von keiner Instanz überprüft wurden. Der Client kann allerdings nach dem ersten Kontakt für alle weiteren Kommunikationsverbindungen überprüfen, ob es sich um den gleichen Kommunikationspartner handelt.

Bei der typischen SIP-Architektur mit einem bis mehreren Proxys auf dem Signalisierungpfad unterliegt die Absicherung mit TLS jedoch Einschränkungen. Da die Sicherung durch TLS auf der Transportschicht stattfindet, ist nur eine Punkt-zu-Punkt-Absicherung zwischen den einzelnen Hosts auf dem Signalisierungspfad möglich. Dies schützt die Signalisierung zwar vor einem externen Angreifer, ein interner Angreifer in Form eines kompromittierten oder nicht vertrauenswürdigen Proxys, kann jedoch ohne Probleme Zugriff auf den Klartext der SIP-Pakete erlangen, diese verändern oder zusätzliche senden. Insbesondere eine auf SIP-Ebene unverschlüsselte Übertragung der Medienkryptographieparameter, wie sie in [ABW05] beschrieben ist, bietet keine Ende-zu-Ende-Absicherung des Schlüsselmaterials. Dadurch wird die Möglichkeit eröffnet, dass der Mediendatenstrom abgehört und entschlüsselt werden kann, falls es dem Angreifer gelingt, sich auf dem Signalisierungspfad durch Kompromittierung eines Proxys oder Manipulation des SIP-Routings einzufügen.

Ein SIP-Client kann eine TLS-gesicherte Verbindung über den gesamten Signalisierungspfad mit Hilfe einer SIPS-URL anfordern. Die Spezifikation sieht das zwar so vor, es kann allerdings nicht ausgeschlossen werden, dass ein Proxy das absichtlich oder durch einen Implementierungsfehler missachtet und eine Anfrage ungesichert weiterleitet wird. Nachfolgende Proxys und das Ziel der Anfrage können diesen Downgrade-Angriff auf die Sicherheitsmerkmale der Signalisierung bei einem vorsätzlichen Angriff nicht erkennen, weil ein Angreifer auch die Via-Header der davor liegenden Proxys manipulieren wird. Gleiches gilt für den Absender der Anfrage.

[2] engl. Certification Authority, kurz CA

5.4 Datagram Transport Layer Security (DTLS)

Ein Nachteil von TLS ist, dass es nur über ein zuverlässiges Transportprotokoll wie TCP übertragen werden kann. Tritt während der Datenübertragung ein Paketverlust auf, verliert das TLS-Protokoll die Synchronisation und kann aufgrund dessen die folgenden Pakete nicht mehr entschlüsseln. Das führt zu einer Unterbrechung der Verbindung.

Da die Singalisierung mit SIP bevorzugt über das unzuverlässige Transportprotokoll UDP abgewickelt wird, was vor allem wegen den kürzeren Verzögerungen und der effizienteren Behandlung im Programmcode des Servers besser geeignet ist, kam ein Bedarf für eine TLS-artige Absicherung des Transports auf. Das Datagramm Transport Layer Security-Protokoll (DTLS) [MR04, RM04] basiert in großen Teilen auf TLS und erweitert es dort, wo es für einen Transport über ein unzuverlässiges Protokoll notwendig ist.

TLS funktioniert nicht komplett über ein unzuverlässiges Transportprotokoll. Der Handshake zu Beginn der Verbindung muss zuverlässig durchgeführt werden, was in DTLS durch wiederholtes Senden der Handshake-Pakete erreicht wird, falls in einem gewissen Zeitraum keine Antwort von der Gegenseite eintrifft und der Host damit von einem Paketverlust ausgehen kann. IP-Pakete müssen nicht zwingend in der Reihenfolge eintreffen, in der sie gesendet wurden, deshalb werden DTLS-Pakete mit einer Sequenznummer versehen, anhand der die eintreffenden Pakete am Zielhost geordnet werden können. Zusätzlich kann ein Handshake-Paket intern auf der DTLS-Schicht fragmentiert werden, um die MTU[3] nicht zu überschreiten und eine Fragmentierung auf der IP-Schicht zu verhindern.

Um nach einem Paketverlust eine weitere Kommunikation zu ermöglichen, wird die Sequenznummer, die bei TLS implizit im Zustandsraum der Verbindung enthalten ist, in DTLS-Paketen mit übertragen. Dadurch kann der Empfänger eines Pakets die Integrität eines Pakets auch dann überprüfen, wenn vorhergehende Pakete verloren gegangen sind.

Eine der Zielsetzungen von DTLS ist es, das ursprüngliche Protokoll nur an den nötigsten Stellen für einen unzuverlässigen Transport zu verändern. Insbesondere wurden keine Vereinfachungen vorgenommen, die eine Verschlechterung der Sicherheitseigenschaften von TLS herbeiführen können.

5.5 Secure Multipurpose Internet Mail Extensions (S/MIME)

Damit eine Ende-zu-Ende-Absicherung im Sinne der Integrität und Vertraulichkeit hergestellt werden kann, ist für SIP zusätzlich zu TLS auch S/MIME vorgesehen. S/MIME ist in RFC 2633 [Ram99] spezifiziert und erlaubt es, MIME-codierte Nachrichten zu signieren und zu verschlüsseln.

[3] Maximum Transfer Unit

Ähnlich zu TLS werden auch bei S/MIME Zertifikate verwendet, um öffentliche Schlüssel auszutauschen und deren Authentizität über Signaturen zu überprüfen. Im Gegensatz zu TLS sind die Zertifikate jedoch nicht Host-, sondern User-basiert und verwenden eine SIP-URL als Identifikationsmerkmal. SIP sieht ebenfalls eine zwingende Schlüsselübertragung per Zertifikat vor, wenn ein Teil des Bodys mit S/MIME signiert worden ist.

Da in den meisten Fällen SDP-Bodys in SIP übertragen werden, ist es naheliegend, diese mit S/MIME zu schützen. Damit kann bereits ein Angriff auf die Kontaktdaten des Medienstroms, wie dem aus Abschnitt 4.3, abgewehrt werden.

Um auch die Integrität der Header zu sichern, können SIP-Nachrichten oder Fragmente davon in einen S/MIME-Body kopiert werden. Der Empfänger kann die Integrität des Bodys überprüfen und mit den Headern aus dem SIP-Paket vergleichen. Besteht eine Differenz zwischen einer Headerzeile aus dem SIP-Paket und der Kopie aus dem Body, kann dies auf eine Manipulation eines Angreifers zurückzuführen sein. In einem beschränkten Umfang kann die Vertraulichkeit für bestimmte Header hergestellt werden, indem der Body zusätzlich verschlüsselt wird, und einige ausgewählte Header sich entweder von den echten SIP-Headern unterscheiden oder diese nur im Body mitgesendet werden. Die Header im verschlüsselten Body sind bevorzugt zu behandeln und sollten für die Verarbeitung im Endgerät, beispielsweise der Anzeige des Anrufers im Display, verwendet werden. Zusätzlich bietet S/MIME einen Schutz vor Replay-Angriffen, wenn ein Date-Header mit signiert bzw. verschlüsselt wird. Es liegt dann im Ermessen des Empfängers, wie stark das im SIP-Header angegebene Datum von dem aktuellen differieren darf, um das Paket noch als akzeptabel einzustufen. Problematisch ist, dass die Uhren der beiden Kommunikationspartner aufeinander abgestimmt sein müssen, was beispielsweise durch einen Zeitabgleich über NTP geschehen kann.

Bei der Integritätsabsicherung über S/MIME treten SIP-spezifische Probleme auf, die eine vollständige Ende-zu-Ende-Absicherung des gesamten Pakets unmöglich machen, weil dazwischenliegende Proxys bestimmte Header verändern müssen. Stark eingeschränkt ist auch die Verschlüsselung von Headern, denn es müssen viele schützenswerte Header für einen funktionierenden Betrieb auf dem Signalisierungspfad im Klartext vorliegen.

Die Verwendung von S/MIME in SIP ist nicht von der SIP-Spezifikation vorgeschrieben. Aus diesem Grund kann ein Client, der eine SIP-Nachricht mit S/MIME-Teilen erhält, diese mit dem Fehlercode “415 Unsupported Media Type” abweisen. Da eine Antwort in so einem Fall ohne jeglichen Schutz auskommt, kann sie von einem Angreifer für einen Downgrade-Angriff [HSSR02] ausgenutzt werden, was eine ungeschützte Signalisierung zur Folge hat.

Wie im Abschnitt 5.3 bereits für TLS beschrieben, benötigt auch S/MIME für die vollständige Absicherung eine PKI. Die Implikationen einer fehlenden PKI bei TLS können auf S/MIME übertragen werden.

Ein weiteres Problem von S/MIME bei der Verwendung in SIP ist die Ineffizienz im Sinne des Platzbedarfs. Da beim Schutz der SIP-Header diese in einem Paket doppelt

übertragen werden und zusätzlich Zertifikate mitgesendet werden können, kann die Größe eines solchen Pakets die Grenzen für eine Übertragung per UDP überschreiten. Ein Einsatz von S/MIME müsste also den Einsatz von TCP implizieren, was wiederum die Nachteile bezüglich der Skalierbarkeit mit sich bringen würde.

Eine Lösung für das Problem wäre es, ähnlich zu den in Abschnitt 5.2 vorgestellten Prinzipien, nur eine Auswahl von Headern zu überprüfen und die Header nicht komplett zu replizieren. Es könnte eine Liste übertragen werden, die dem Empfänger angibt, welche Header dieser in eine kanonischen Form bringen soll, um darüber eine Signatur zu generieren. Zum Vergleich müssten statt des kompletten Pakets, das in einem Body des Typs message/sip übertragen wird, lediglich der Body, die Liste der geschützten Header, deren genauer Aufbau noch spezifiziert werden müsste, und die Signatur vom Typ message/pkcs7-signature übertragen werden. Im Gegensatz zur klassischen S/MIME-Absicherung mit Headerreplikation hätte diese Vorgehensweise den Nachteil, dass eine Integritätsüberprüfung bei der Veränderung eines Headers fehlschlägt und der Client nicht überprüfen kann, was modifiziert wurde.

5.6 Schlüsseltausch

Zum Schutz der Vertraulichkeit und Integrität der Sprachübertragung ist SRTP [BMN+04] als Übertragungsprotokoll vorgesehen. SRTP spezifiziert nur den Transport der verschlüsselten Daten und nicht den Austausch von Schlüsselmaterial, der für einen sicheren SRTP-Medienstrom erforderlich ist.

Die ersten beiden Verfahren übertragen die Schlüssel im Klartext im SDP-Body des SIP-Pakets und setzen eine verschlüsselte Übertragung voraus. Um unabhängig von der Absicherung durch höhere Schichten oder andere Maßnahmen einen Schlüsselaustausch durchzuführen, existieren spezielle Protokolle, an die folgende Anforderungen gestellt werden [Bil03]:

Ende-zu-Ende-Absicherung Die wichtigste Anforderung an ein Schlüsseltauschprotokoll ist es, die Sicherheit des Schlüsselmaterials von Ende zu Ende zu garantieren. D.h., selbst wenn ein Angreifer oder ein Proxy die ausgetauschten Nachrichten mitlesen kann, kann er aus den abgehörten Daten keinen gültigen Schlüssel abgeleitet.

Schutz vor Replay-Angriffen Es darf keine Möglichkeit bestehen, dass ein Angreifer durch die Wiederholung eines Dialogs eine Sitzung aufbauen kann, in der die gleichen Schlüssel verwendet werden. Diese Anforderung ist vor allem deswegen wichtig, weil eine Wiederverwendung eines Schlüsselstroms eine Kryptanalyse des RTP-Stroms ermöglichen würde.

Authentizität Die Nachricht soll eindeutig dem Sender zugeordnet werden können.

Integrität Eine manipulierte Nachricht soll durch beide Seiten feststellbar sein.

Schutz vor Downgrade-Angriffen Ein Angreifer darf durch Manipulation der Nachrichten nicht erreichen, dass die Kommunikationspartner mit schwächeren kryptographischen Parametern kommunizieren. Diese Anforderung ist implizit in der Integritätsanforderung enthalten.

Es gibt weitere Anforderungen, die für die sichere Funktion des Schlüsseltauschprotokolls nicht notwendig, für den VoIP-Betrieb allerdings von Vorteil sind:

Effizienz Beim Rufaufbau sollten die Verzögerungen durch zusätzliche Sicherheitsmaßnahmen so niedrig wie möglich gehalten werden. Wichtig ist dafür, dass die Anzahl der ausgetauschten Nachrichten gering gehalten wird, weil das Senden von Nachrichten in einem Netzwerk im Verhältnis zu Berechnungen in einem Gerät relativ große Verzögerungen verursacht und damit den Engpass darstellt [Bil03].

Unabhängigkeit von einem Transportprotokoll Bei VoIP mit SIP werden mindestens zwei separate Informationskanäle für die Signalisierung und die Sprachübertragung benötigt. Unter anderem sorgt dieser Umstand dafür, dass VoIP mit NAT problematisch ist. Benötigt der Schlüsseltausch eine weitere Verbindung, würde das die Komplexität weiter steigern. Die erste Nachricht des Schlüsseltauschprotokolls kann bei einer separaten Durchführung erst dann gesendet werden, wenn die Kontaktdaten des anderen Endpunkts bekannt sind. Das wäre bei SIP erst nach dem Empfang der ersten provisorischen Antwort der Gegenseite möglich. Zu diesem Zeitpunkt ist beim Transport mit SIP bereits ein Round-Trip möglich, der bei MIKEY für einen Schlüsseltausch ausreichend ist.

In Anbetracht dessen, dass Erweiterungen von SDP spezifiziert wurden, die ein Framework für den Transport von Schlüsseltauschprotokollen anbieten [ACL+05], ist es vorteilhaft, wenn die Spezifikation der Protokolle unabhängig vom Transport erfolgt, und die Nachrichten in den Signalisierungspaketen gekapselt werden können.

Diese Anforderung und die der Effizienz stehen in einem unmittelbaren Zusammenhang miteinander. Bei einer Invite-Transaktion werden drei Nachrichten ausgetauscht. Ein Schlüsseltauschprotokoll darf nicht mehr als diese drei Nachrichten benötigen, weil es sonst nicht in die Invite-Transaktion transportiert werden kann.

Ein Protokoll, das alle Bedingungen erfüllt, ist MIKEY und wird in Abschnitt 5.6.3 vorgestellt. Prinzipiell könnte auch auf andere Protokolle, wie IKE, zurückgegriffen werden. Die letzten beiden Bedingungen werden dadurch nicht erfüllt. IKE benötigt für die Initialisierung insgesamt sechs Pakete im *Main Mode* oder drei im *Aggressive Mode*, der aufgrund von Sicherheitsbedenken nicht empfohlen wird. Aufgrund der Ineffizienz von IKE kann ein Schlüsseltausch mit diesem Protokoll nicht in eine Invite-Transaktion von SIP gekapselt werden.

5.6.1 SDP: k-Attribut

Eine rudimentäre Möglichkeit Schlüssel auszutauschen, ist bereits in SDP [HJ98] spezifiziert. Damit können Schlüssel mit Hilfe des k-Parameters direkt im SDP-Body übertragen werden. Der Aufbau des Parameters ist sehr einfach gehalten. Es wird immer eine Methode angegeben, die, wie im folgendem Beispiel dargestellt, durch einen Doppelpunkt vom Schlüssel getrennt ist:

```
k=base64:mQGiBD6XCeERBACuZWnuQR/3Onf2NgFvsIjAjV2Wp2HJSqe
```

In diesem Beispiel ist der Schlüssel, dessen Aufbau nicht genauer spezifiziert ist, Base64-codiert. Weitere in [HJ98] spezifizierte Methoden sind *clear*, in der der Schlüssel uncodiert übertragen wird, *uri*, bei der die Position des Schlüssels mit einer URL angegeben wird, und *prompt*, bei der der Benutzer zur Eingabe aufgefordert werden soll.

5.6.2 Security Descriptions

Die in SDP spezifizierte Methode ist nur begrenzt praxistauglich, da kein Austausch von weiteren Parametern, wie dem verwendeten Verschlüsselungsalgorithmus, vorgesehen ist. Eine umfangreichere Möglichkeit des Schlüsseltausches sind Security Descriptions [ABW05], die SDP um ein weiteres Attribut erweitern, das in einem a-Parameter übertragen wird. Die Kennung für das Attribut ist *crypto* und folgendermaßen aufgebaut:

Tag Ein eindeutiger Identifier für das Attribut.

Crypto-Suite Der verwendete Verschlüsselungs- und Hashalgorithmus mit den jeweiligen Schlüssellängen und kryptographischen Modi.

Schlüsselparameter Dieses Feld entspricht in etwa dem im letzten Absatz beschriebenem k-Feld. Es enthält eine Methode, die durch einen Doppelpunkt vom Schlüssel und weiteren Parametern getrennt ist.

Die Spezifikation geht jedoch weiter und erlaubt es, für eine SRTP-Sitzung alle notwendigen Parameter wie die maximal erlaubte Anzahl der verschlüsselten Pakete, den MKI (siehe Abschnitt 3.6) und dessen Länge anzugeben.

Optionale Sitzungsparameter Dieses optionale Feld kann die Parameter von Erweiterungen aufnehmen.

Das in Abschnitt 3.3.2 beschriebene Modell zum Aushandeln von Sitzungsparametern wird auch für Security Descriptions verwendet. Der anrufende Client kann im SDP-Body mehrere, nach absteigender Präferenz sortierte *crypto*-Attribute übertragen. Der Empfänger entscheidet sich anhand seiner Möglichkeiten und Sicherheitsrichtlinien für eines der *crypto*-Attribute und sendet es als Kopie im SDP-Body seiner Antwort. Trifft

das auf keines der angebotenen Attribute zu, muss die Anfrage abgewiesen und der Rufaufbau abgebrochen werden.

Verlangt der Signalisierungspfad Zugriff auf den SDP-Body, können die *crypto*-Attribute von den Proxys mitgelesen werden. Aus diesem Grund wird empfohlen, diese Attribute in einen separaten MIME-Part unterzubringen. Der SDP-Teil mit den Medienparametern kann dann im Klartext übertragen werden, während das Schlüsselmaterial mit dem öffentlichen Schlüssel des anderen Endpunkts verschlüsselt werden kann.

5.6.3 Multimedia Internet Keying (MIKEY)

MIKEY wurde im August 2004 in RFC 3830 [ACL+04] standardisiert und speziell für Echtzeitanwendungen wie VoIP entwickelt. Ziel war es, ein sicheres und effizientes Schlüsseltauschprotokoll zu spezifizieren, das unabhängig von der zugrundeliegenden Transportschicht verwendet werden kann. Neben dem Punkt-zu-Punkt-Einsatzgebiet kann es für Multicast-Anwendungen verwendet werden.

In Abbildung 5.2 ist der Ablauf der Generierung von kryptographischen Schlüssel mit MIKEY dargestellt. Beide Seiten vereinbaren zunächst einen *TEK Generation Key* (TGK), aus dem die teilnehmenden Kommunikationspartner den *Traffic-Encrypting Key* (TEK) ableiten. TGK, TEK und weitere Parameter wie das Rekeying-Intervall bilden gemeinsam eine *Crypto Session* (CS), die durch einen Identifier gekennzeichnet ist. Mehrere Crypto Sessions können zu einem *Crypto Session Bundle* (CSB) zusammengefasst werden, in dem ausgewählte Parameter von den dazugehörigen Crypto Sessions geteilt werden können. Im folgenden Abschnitt wird zunächst das Kommunikationsprotokoll beschrieben. Im darauf folgenden Abschnitt werden die Berechnungen beschrieben, die zum Ableiten der Sitzungsschlüssel aus dem TGK verwendet werden.

5.6.3.1 Protokoll

Zur Vereinbarung des TGK werden drei Verfahren spezifiziert, die in Tabelle 5.2 aufgeführt sind. Der Aufwand gibt die Anzahl der Nachrichten an, die ausgetauscht werden müssen. Hier wird deutlich, dass MIKEY die Schlüssel in maximal zwei Kommunikationsschritten bzw. einem Round-Trip vereinbart und dadurch für den Transport über SIP in einer Invite-Transaktion geeignet ist. Der Diffie-Hellman-Schlüsseltausch bietet den Vorteil der *Perfect Forward Secrecy* [DvOW92]. Das bedeutet, dass die Kompromittierung eines geheimen Schlüssels, der für jeden Durchgang des Protokolls verwendet wird, nicht zur Offenlegung aller bisherigen Schlüssel führt. Diese Eigenschaft ist bei PSK und PK nicht gegeben, weil der generierte TGK immer mit dem selben Schlüssel verschlüsselt wird. Bei DH wird der Schlüssel dagegen durch den Austausch von zufälligen Exponenten vereinbart.

Grundsätzlich besteht eine MIKEY-Nachricht aus mehreren Blöcken, die bis auf den Header, der der erste Block ist, und der abschließenden Signatur beliebig angeordnet

Verfahren	Aufwand	Vorteile	Nachteile
Pre-Shared Key (PSK)	1-2	Einfach und effizient, Authentifizierung.	Skaliert schlecht
Public-Key (PK)	1-2	Skaliert besser als PSK, effizient	Benötigt PKI
Diffie-Hellman (DH)	2	Perfect Forward Secrecy	Ineffizient

Tabelle 5.2: Von MIKEY unterstützte Schlüsseltausch-Methoden

werden können. In jedem Block wird der Typ des nächsten Blocks angegeben, damit diese identifiziert werden können. Die erste MIKEY-Nachricht besteht bei allen Methoden aus folgenden Komponenten:

Header Der Header enthält folgende Felder:

- Version des verwendeten Protokolls (aktuell 0x01)
- Nachrichtentyp
- Payload-Typ des Blocks, der dem Header folgt.
- Ein Flag, das angibt, ob der Sender mit einer Verify-Nachricht, die der Authentifizierung dient, antworten soll.
- Verwendete Pseudo-Zufallsfunktion
- Eine CSB-Id, die zwischen den Kommunikationspartner eindeutig sein muss.
- Anzahl der Crypto Sessions
- Typ der Crypto Sessions. Zur Zeit ist nur SRTP (0) definiert.

Zeitstempel Der Zeitstempel dient als Gegenmaßnahme zu Replay-Angriffen. Der Zeitstempel in einer Antwort entspricht immer dem aus der initialen Nachricht.

Pseudozufälliger Bytefolge Diese Bytes werden beim Ableiten von Keys aus dem TGK in die Berechnung einbezogen.

Identität der beiden Kommunikationspartner Dieser Block ist optional und kann Identifikationsstring und X.509-Zertifikate [HPFS02] enthalten oder mit URLs auf Zertifikate verweisen.

Sicherheitsrichtlinien In diesem optionalen Block werden die Richtlinien für die Crypto Session beschrieben. Für SRTP sind das z.B. die verwendeten kryptographischen Algorithmen und Schlüssellängen, Authentisierungsparameter und Rekeying-Intervalle.

Beim PSK-Verfahren wird der TGK verschlüsselt in einem KEMAC-Block übertragen. Als Verschlüsselungs-Key wird aus dem Pre-Shared-Key, wie im nächsten Abschnitt beschrieben, ein spezieller MIKEY-Verschlüsselungs-Key abgeleitet. Bei PK wird ein *Envelope Key* mit dem Public Key des Empfängers verschlüsselt. Der KEMAC-Block ist

wie bei PSK aufgebaut und enthält zusätzlich noch einen ID-Block mit einem Identifikationsstring. Der Verschlüsselungs-Key wird aus dem Envelope-Key abgeleitet.

In beiden Fällen wird MAC über die gesamte Nachricht berechnet und im KEMAC-Block unverschlüsselt mitübertragen, wodurch die Integrität der Nachricht gesichert ist. Nach der ersten Nachricht können beide Seiten bereits ihre Keys ableiten. Optional oder auf Aufforderung des Initiators des Schlüsseltauschs hin kann der Empfänger der ersten Nachricht eine Antwort senden, die den Verify-Block enthält. Zur Steigerung der Effizienz kann der Envelope-Key gecached und bei späteren Protokollläufen als Pre-Shared-Key wiederverwendet werden.

Bei der DH-Methode werden die Parameter der Gruppe G der Generator g vom Initiator des Schlüsseltausches ausgewählt. In den beiden Nachrichten werden g^{x_I} und g^{x_R} ausgetauscht und anschließend der TGK $g^{x_I \cdot X_R}$ berechnet. Auch bei DH wird ein MAC berechnet und in einem speziellen SIGN-Block gesendet.

5.6.3.2 Ableiten der Keys aus dem TGK

Zur Ableitung der Keys wird eine Pseudo-Zufallsfunktion verwendet. MIKEY definiert eine Funktion, die auf der SHA-1-Hashfunktion [NIS95] basiert und aus der folgenden Funktion berechnet wird:

$$P(s, label, m) = SHA1(s, A_1 \cdot label) \cdot \ldots \cdot SHA1(s, A_m \cdot label)$$

A_i ist eine Folge die wie folgt definiert ist:

$$A_i = \begin{cases} label & \text{, wenn } i = 0 \\ SHA1(s, A_{i-1}) & \text{, wenn } i > 0 \end{cases}$$

Der Parameter m ist die Schlüssellänge dividiert durch 160. s ist ein Schlüssel, der in n Komponenten $s = s_1 \cdot \ldots \cdot s_n$ zerlegt wird. Jedes s_i ist 256 Bit lang. Die Pseudo-Zufallsfunktion PRF ergibt sich durch bitweise XOR-Verknüpfung von P, die durch $\oplus$ dargestellt wird:

$$PRF(s, label) = \bigoplus_{i=1}^{n} P(s_i, label, m)$$

Das Label ist eine Konkatenation einer Konstante, der CS-Id, der CSB-Id und eines Zufallsbitstrings, der in der ersten Nachricht gesendet wird. Die vordefinierten Konstanten sind in Tabelle 5.3 aufgeführt. Neben den Session-Keys sind auch Labels für Keys angegeben, die nur intern von MIKEY verwendet werden. Je nach Einsatzgebiet können ein TEK oder drei unterschiedliche Session-Keys abgeleitet werden. Zum Beispiel enthält SRTP bereits Regeln für die Ableitung von Keys aus einem Master-Key, der durch den TEK repräsentiert werden kann.

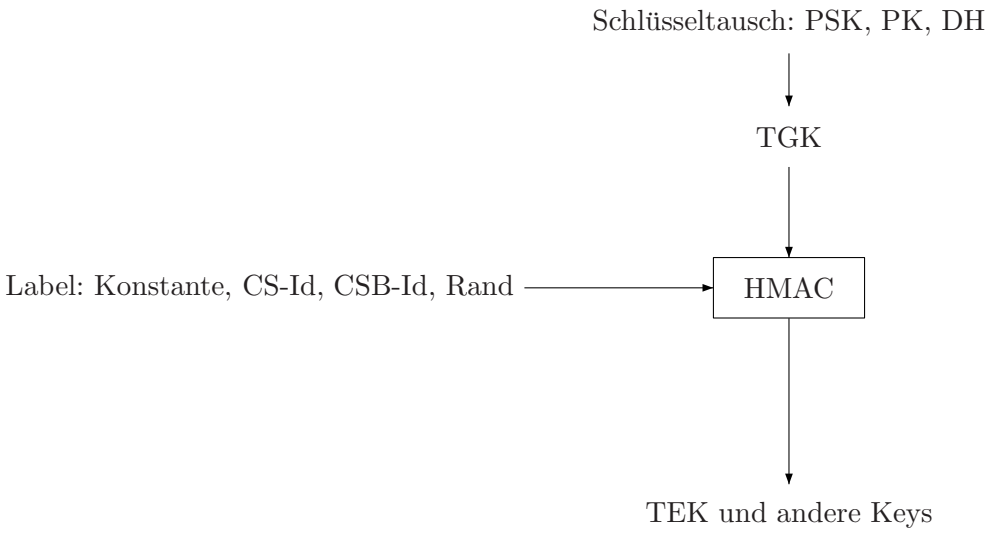


Abbildung 5.2: Generierung von Schlüsseln mit MIKEY

Konstante	Abgeleiteter Key
Session-Keys	
0x2AD01C64	TEK
0x1B5C7973	Authentifizierungs-Key
0x15798CEF	Verschlüsselungs-Key
0x39A2C14B	Salt
MIKEY-Keys	
0x150533E1	Verschlüsselungs-Key
0x2D22AC75	Authentifizierungs-Key
0x29B88916	Salt

Tabelle 5.3: MIKEY-Konstanten für die Schlüsselableitung

Zur Verschlüsselung in MIKEY wird das bereits in Abschnitt 3.6.2 beschriebene AES im Counter-Mode verwendet. Der Initialisierungsvektor s ist durch

$$c = (s \oplus (0\text{x}0000 \cdot CSB - Id \cdot t)) \cdot 0\text{x}0000$$

bestimmt, wobei s den abgeleiteten MIKEY-Salt und t den Zeitstempel aus der ersten MIKEY-Nachricht darstellt.

5.6.3.3 Transport von MIKEY in SIP

Eine Erweiterung von SDP erlaubt es, den Dialog eines Schlüsseltauschprotokolls in SDP-Attributen zu transportieren [ACL+05]. Die Daten des Schlüsseltauschprotokolls werden mit der Bezeichnung des Protokolls Base64-codiert in einem Attribut mit dem Namen "key-mgmt" übertragen. Ein Beispiel für die Übertragung von MIKEY über dieses Attribut könnte folgendermaßen aussehen:

```
a=key-mgmt:mikey AQAFgM0AdlABAAAAAAAAAAAA...
```

Es ist zulässig, mehrere dieser Attribute zu übertragen, um unterschiedliche Schlüsseltauschmethoden vorzuschlagen. Der Empfänger kann so im Rahmen des in Abschnitt 3.3.2 beschriebenen Offer/Answer-Modells eine dieser Methoden auswählen. Da die unterschiedlichen Methoden sich qualitativ unterscheiden können, bietet sich an dieser Stelle eine Angriffsmöglichkeit für Downgrade-Angriffe. um solche Angriffe zu verhindern, wird eine Liste generiert, in der die gesendeten Verfahren jeweils durch ein Semikolon getrennt aufgeführt werden. Diese Liste wird jedem Schlüsseltauschprotokoll als Eingabe übergeben und muss von diesem signiert übertragen werden, so dass ein weglassen eines Attributs festgestellt werden kann. In MIKEY wird diese Liste als *General Extension Payload*-Block übertragen und fließt mit in die Berechnung des MACs ein.

Da auch provisorische SIP-Antworten mit einer Erweiterung [SR02] zuverlässig übertragen werden können, gibt es mehrere Möglichkeiten, die MIKEY-Antwort zu übertragen. In [BEOV05] wurden die unterschieldichen Möglichkeiten untersucht.

Eine Übertragung in der 200-Antwort wird auch von Clients unterstützt, die keinen zuverlässigen Transport von provisorischen Antworten kennen. Nachteilig wirkt sich dabei aus, dass der letzte Schritt des Schlüsseltauschs erst nach der Annahme des Gesprächs erfolgt und damit zum Verlust von Sprachdaten führt. Ein weiteres Problem ist, dass das Telefon bei einer fehlgeschlagenen Verifikation zwar klingelt, das Gespräch nach der Annahme nicht zustandekommt[4]. Steht der Schlüssel bereits nach der ersten Nachricht fest, wie es bei PSK und PK der Fall ist, dann könnten beide Seiten bereits vor der Verifizierung verschlüsselte RTP-Pakete austauschen.

Bei einer Übertragung in der provisorischen "180 Ringing"-Antwort, kann der zweite Schritt während des Klingelns des Telefons vollzogen werden. Die kryptographischen

[4] wird in der Literatur oft auch als *ghost rings* bezeichnet.

Schlüssel sollten bei einer natürlichen Zeit zwischen Klingeln und Gesprächsannahme zur Verfügung stehen. Das zweite Problem, dass ein Gespräch nach dem Klingeln abgewiesen wird ist dadurch noch nicht gelöst. Aus diesem Grund wird in [BEOV05] vorgeschlagen, die MIKEY-Antwort in einer “183 Session in Progress”-Antwort zu senden, die vor dem Klingeln beim Benutzer gesendet wird. Diese Problemlösung wird mit einer Verzögerung zwischen der Anwahl und der Anrufsignalisierung beim Benutzers erkauft.

Ein anderer Ansatz, der diese Probleme löst sind die sogenannten Preconditions [CMR02]. Durch sie können beide Seiten Bedingungen vereinbaren, die bis zur Signalisierung des Benutzer erfüllt sein müssen. Unter anderem werden in [AW05] Security Preconditions spezifiziert.

5.7 IPsec und VPN-Lösungen

Eine generische Methode zur Absicherung von Datenverkehr bieten VPN-Lösungen, bei denen der gesamte IP-Verkehr verschlüsselt und authentisiert werden kann. VPNs werden im Normalfall dazu verwendet, lokale Netze über WANs sicher zusammenzuschalten. Einzelne Verbindungen abzusichern ist nicht das primäre Aufgabengebiet von VPNs. In dieser Arbeit wird hauptsächlich IPsec [KA98] behandelt, weil es in vielen Betriebssystemen verfügbar und ein offener Standard ist.

IPsec besteht im wesentlichen aus zwei zusätzlichen Headern, die dem IP-Header hinzugefügt werden. Ein Header ist der Authentication Header (AH), der die Authentizität des gesamten Pakets inklusive der Felder im IP-Header, die auf der Route nicht verändert werden dürfen, absichert. Da die IP-Adressen zu den authentisierten Headern gehören, kann IPsec in NAT-Umgebungen nur ohne AH verwendet werden.

Das Encapsulated Security Payload Protokoll (ESP) stellt den zweiten Header dar und ist für die Verschlüsselung des IP-Payloads zuständig. Eine Authentisierung der Daten wird ebenfalls durchgeführt, im Gegensatz zum AH wird der IP-Header nicht mit in die Berechnung einbezogen.

Das für IPsec vorgesehene Schlüsseltauschprotokoll ist IKE, welches bereits in Abschnitt 5.6 behandelt wurde. Grundsätzlich sollte bei IPsec zwischen einer Absicherung der Signalisierung und der des Medienstroms unterschieden werden. Wird bei der Signalisierung immer nur mit einem Proxy kommuniziert, was bei VoIP häufig der Fall ist, dann kann vorher eine einmalige Assozierung mit dem Proxy erfolgen, ohne dass der zeitaufwändige Schlüsseltausch bei jedem Rufaufbau erfolgen muss.

Die Ergebnisse aus [BEOV05] bezüglich der Verzögerungen beim Beantworten von Gesprächen und der Folgerung des Authors lassen zum Schluss kommen, dass eine Absicherung von VoIP mittels SRTP der über IPsec vorzuziehen ist. Auch die BSI-Studie [AAGea05] empfiehlt eine Absicherung über TLS oder S/MIME und SRTP.

6 Grundlagen der Sicherheit

Sicherheit kann nach mehreren Kriterien unterteilt und unter verschiedenen Gesichtspunkten bewertet werden. In diesem Kapitel werden in den Abschnitten 6.1 und 6.2 die Grundlagen für die Bewertung gelegt, indem Schutzziele und Angreifermodelle vorgestellt werden. Abschnitt 7 befasst sich mit den Absicherungsmaßnahmen aus Kapitel 5. Dafür wird zunächst analysiert, welche Schutzziele durch die vorgestellten Protokolle abgedeckt werden und in welcher Qualität sie abgesichert werden. Da zwischen den einzelnen Maßnahmen Abhängigkeiten bestehen, die bei Nichterfüllung zur Wirkungslosigkeit einer Absicherung führen können, werden sie auch untersucht. Anschließend wird begutachtet, welche Schutzziele durch Kombinationen der einzelnen Protokolle erfüllt werden. Dabei werden unter anderem Szenarien betrachtet, die derzeit häufig in der Realität anzutreffen sind, und eine obere Grenze ermittelt, bis zu der ein VoIP-System beim aktuellen Stand der Technik abgesichert werden kann.

In Abschnitt 8 wird mit den Attack Trees ein formales Modell zur Analyse der Sicherheit eines Gesamtsystems eingeführt. In Abschnitt 8.1 wird ein Kostenmaß definiert, dass die Schwierigkeit eines Angriffs anhand von technischen Anforderungen für den Angreifer bewertet. Für die drei Schutzziele Verfügbarkeit, Vertraulichkeit und Integrität werden Attack Trees konstruiert und in Abschnitt 8.5 bewertet. In diesen Abschnitten findet eine Bewertung des proprietären VoIP-Dienstes Skype statt, um einen Vergleich zwischen den derzeit am weitesten verbreiteten VoIP-Technologien zu ermöglichen. Abschlie0end werden die Ergebnisse der Bewertung kritisch betrachtet und deren Aussagekraft bewertet.

6.1 Schutzziele

Allgemein lassen sich Schutzziele in drei Kategorien unterteilen: Vertraulichkeit, Integrität und Verfügbarkeit. Das erste Schutzziel besagt, dass die Inhalte der zwischen Kommunikationspartnern ausgetauschten Nachrichten nur diesen zugänglich sein sollen. Bei VoIP bedeutet das bei der Betrachtung der Signalisierung, dass ein Außenstehender keine Informationen über die Teilnehmer eines Anrufs erfahren kann (Anonymität). Es soll auch die Möglichkeit geschaffen werden, die gegenseitige Anonymität beider Seiten zu gewährleisten. Weiter fallen unter dieses Kriterium die ausgetauschten Parameter für die Medienverbindung, besonders schützenswert sind hier kryptographische Schlüssel. Auf die Übertragung der Sprachdaten bezogen bedeutet die Vertraulichkeit letztendlich, dass die Inhalte des Gesprächs von keiner dritten Partei abgehört werden können.

Die zweite Anforderung, die *Integrität*, bedeutet, dass Nachrichten nicht ohne eine Erkennungsgelegenheit durch den Empfänger gefälscht oder manipuliert werden dürfen. Das ist besonders bei der Signalisierung wichtig, weil dadurch Angriffe realisiert werden können. Beispiele dafür sind der Denial-of-Service in 4.1 durch das Einfügen gefälschter SIP-Pakete in einen Dialog oder der Angriff aus 4.3, in dem die Zieladressen des RTP-Medienstroms im SDP-Body manipuliert werden. Wichtig ist, dass der signalisierte Anrufer und der Angerufene ihre Identität beweisen können (Authentizität). Das gilt insbesondere dann, wenn der Anrufer unbekannt ist und eine Wiedererkennung anhand der Stimme ausgeschlossen wird. Fehlt diese Eigenschaft, können Phishing-Angriffe auch auf die IP-Telefonie ausgeweitet werden. Bei Medienströmen können die übertragenen Sprachdaten manipuliert werden, was bei Gesprächen jedoch durch die Teilnehmer erkannt werden kann. Anders sieht es dagegen bei automatischen Ansagen aus, auf die ein Angreifer beispielsweise einen Replay-Angriff durchführen kann. Weiterhin besteht die Möglichkeit, DTMF-Töne über einen speziellen Payload-Typ zu übertragen [SP00], deren Manipulation auf beiden Seiten weniger auffällig ist als die der Sprachdaten.

Ein weiterer Punkt, der der Integrität zuzuordnen ist, ist die korrekte Erfassung von Accounting-Daten. Darunter fällt insbesondere die Gebührenabrechnung. Ein Betrug ist hier von zwei Seiten möglich. Auf der einen Seite durch den Benutzer eines VoIP-Systems, der versucht gebührenpflichtige Gespräche kostenlos zu führen. Ein anderer Grund, um von Benutzerseite einen Betrug durchzuführen, ist es, die Existenz eines geführten Gesprächs zu verleugnen oder dessen erfasste Dauer zu manipulieren. Auf der Seite des Betreibers kann dagegen das komplementäre Interesse bestehen und mehr Gebühren zu berechnen.

Der letzten Anforderung, der *Verfügbarkeit*, wird bei der Telefonie eine besonders wichtige Rolle eingeräumt. Sie besagt, dass jederzeit eine Kommunikation zwischen den Teilnehmern ermöglicht werden muss. Die Verfügbarkeit des Gesamtsystems hängt von der Verfügbarkeit aller Komponenten ab, d.h. bei VoIP, dass die VoIP-Geräte selbst betriebsbereit sein müssen, was wiederum eine stabile Stromversorgung erfordert. Des Weiteren muss das verwendete IP-Netz funktionieren und das hängt von der Verfügbarkeit der Infrastrukturelemente wie Switches und Routern ab.

Eine andere Anforderung, die nicht in dieses Schema fällt, ist der Schutz Dritter vor Gefahren, die durch den Einsatz von Voice-over-IP entstehen. VoIP-Endgeräte erzeugen einen konstanten Paketstrom, der eine gewisse Bandbreite konsumiert. Gelingt es einem Angreifer, einen solchen Paketstrom an einen bestimmten Host weiterzuleiten, sind darüber DDoS[1]-Angriffe denkbar.

[1] Distributed Denial of Service

6.2 Angreifermodelle

Zur Beurteilung der Sicherheit von Protokollen ist eine Differenzierung der möglichen Angreifer notwendig. Im Allgemeinen versucht ein Angreifer, Informationen zu erhalten, Nachrichten abzufangen oder zu manipulieren, einen Dienst zu behindern oder ganz außer Betrieb zu setzen.

Man unterscheidet zwischen zwei verschiedenen Typen von Angreifern: passive und aktive. Ein *passiver Angreifer* kann die Kommunikation nur abhören und verhält sich ansonsten still. Ein aktiver Angreifer hingegen kann zusätzlich Nachrichten senden. Dieser Angreifertyp kann noch unterteilt werden. Einerseits gibt es die Angreifer, die Nachrichten abfangen und manipulieren, andererseits die, die diese Fähigkeit nicht besitzen und nur in der Lage sind, weitere Nachrichten in einen bestehenden Dialog einzufügen. Das ist der Fall, wenn die Nachricht unabhängig von der Ankunft beim Angreifer in ihrer Originalform beim Empfänger zugestellt wird, was z.B. bei Medien, die Nachrichten grundsätzlich an alle Netzteilnehmer zustellen, der Fall ist. Zu solchen Medien gehören Funk, Hubs und Netze mit Bus-Topologie.

Eine weitere Unterscheidung kann zwischen den Informationen, die der Angreifer mitlesen und darauf Einfluss nehmen kann, erfolgen. Da bei VoIP die Signalisierung bei den gängigen Protokollen bis auf IAX unabhängig von der Sprachübertragung abläuft und diese Nachrichten auf unterschiedlichen Pfaden geroutet werden, bietet sich hier eine Differenzierung zwischen den beiden Informationskanälen an.

Ein anderes Angreifermodell ist der *blinde Angreifer*. In diesem Angreifermodell kann der Angreifer die Datenströme der Kommunikationspartner nicht mitlesen, weil er außerhalb des Routingpfads liegt. Dennoch kann er Pakete an mindestens eine der beiden Seiten senden. Ein solcher Angreifer verfügt über keine dialog- bzw. sitzungsspezifischen Informationen. Deshalb müssen die für einen erfolgreichen Angriff fehlende Werte entweder erraten oder ganz weggelassen werden, was durch Fehler in der Implementierung und Protokollspezifikation vereinfacht werden kann.

Eine Unterteilung kann auch in *interne* und *externe* Angreifer erfolgen. Der *interne Angreifer* ist dadurch definiert, dass er der angegriffenen Organisation angehört und ein legitimer Teilnehmer des VoIP-Systems ist. Dieser Angreifertyp besitzt oft tiefer gehende Informationen über das angegriffene System, verfügt über mehr Rechte und genießt ein gewisses Vertrauen in der Organisation. Der *externe Angreifer* ist dagegen kein Mitglied der Organisation und hat dadurch weniger Vorteile gegenüber denen eines internen Angreifers.

In dieser Arbeit werden die Begriffe interner und externer Angreifer auch zur Unterscheidung verwendet, ob sich der Angreifer auf dem Signalisierungspfad befindet oder nicht. Ein Anrufer kann z.B. durchaus Angreifer sein, wenn er durch Manipulation der Signalisierung versucht, die Abrechnung eines kostenpflichtigen Gesprächs zu seinen Gunsten zu beeinflussen.

7 Analyse der Sicherheitsmaßnahmen

Jede der in Kapitel 5 vorgestellten Sicherheitsmaßnahmen deckt bestimmte Sicherheitsziele ab. Tabelle 7.1 bietet eine Übersicht über alle in dieser Arbeit vorgestellten Sicherheitsmaßnahmen und bewertet die Absicherung der Schutzziele Authentizität, Vertraulichkeit und Integrität. Das Schutzziel der Verfügbarkeit wurde an dieser Stelle bewusst ausgelassen, weil es nicht im Aufgabenbereich der Protokolle liegt. Deckt ein Verfahren kein Schutzziel einer der Teilgebiete ab, wird in den jeweiligen Bereichen aus Gründen der Übersichtlichkeit auf Markierungen verzichtet. Eine weitere Spalte gibt an, ob ein Verfahren zum Schlüsseltausch angewendet werden kann. Im Gegensatz zu der Bewertung in [AAGea05] wurde hier eine Unterteilung in die beiden Teilbereiche Signalisierung und Medienübertragung vorgenommen, um so eine feinere Sicht auf die Fähigkeiten der Protokolle zu bekommen.

Bei der Bewertung der Verfahren wird zwischen unterschiedlichen Abstufungen der Absicherung und besonderen Eigenschaften differenziert. Bei der Signalisierung wird zum Beispiel unterschieden, ob ein Verfahren über beliebige Teile des Signalisierungspfades wirkt oder nur eine Punkt-zu-Punkt-Absicherung wie TLS bietet. Die Markierung "x+" bezieht sich auf die Integrität der Signalisierungspakete und besagt, dass nur ein Teil des Pakets abgesichert wird. Eine vollständige Absicherung ist aufgrund der durch Proxy vorgenommenen Veränderungen der Pakete nur von Punkt zu Punkt möglich, was durch die Markierung "x-" angezeigt wird. Die Authentifizierung kann benutzer- oder hostbasiert erfolgen und wird entsprechend durch die Markierung B und H gekennzeichnet. Aufgrund dieser Unterscheidung ist es sinnvoll, eine benutzer- und eine hostbasierte Authentifizierung parallel einzusetzen. Ein bei TLS häufig anzutreffendes Szenario ist, dass sich ein Server gegenüber dem Client authentifiziert und der Benutzeraccount durch eine Digest-Authentifizierung identifiziert wird. Prinzipiell kann eine hostbasierte Authentifizierung auch einen einzelnen Benutzer identifizieren, falls dieser der einzige Benutzer des jeweiligen Hosts ist oder je nach Nutzer unterschiedliche Zertifikate vorgewiesen werden. Auf der anderen Seite kann ein Host nur einen Benutzeraccount enthalten, welcher dadurch den Host identifiziert. Der Hauptunterschied besteht darin, dass die hostbasierte Authentifizierung auf der Transportschicht und die benutzerbasierte auf der Anwendungsebene stattfindet.

TLS und DTLS führen beide einen Schlüsseltausch zwischen Komponenten des Signalisierungspfades durch, um die jeweiligen Verbindungen abzusichern. Das wurde durch ein "+" in der Spalte für den Schlüsseltausch vermerkt. Für die Medienübertragung ist dieser Schlüsseltausch jedoch nicht von Bedeutung. Schlüsseltauschverfahren, die mit

“x-” markiert sind, benötigen eine zusätzliche Absicherung der Vertraulichkeit und Authentizität des Signalisierungskanals. Die Verfahren, die dafür verwendet werden, können aus den entsprechenden Spalten für die Signalisierung abgelesen werden.

7.1 Analyse von Absicherungsszenarien

Die Symbole in der Legende der Tabelle 7.1 sind nach absteigender Absicherung sortiert, so dass bei einer Verknüpfung der Verfahren die Markierung gewählt werden kann, die am höchsten im jeweiligen Teil der Tabelle steht. Dadurch kann für beliebige Kombinationen die Gesamtabsicherung des VoIP-Systems ermittelt werden. Da einige Markierungen eine gleichwertige Absicherung repräsentieren, wurden sie Kategorien zugeordnet, die bei einer Analyse eine Sicht erlauben, die von technischen Details, wie sie beispielsweise bei der Authentifizierung gegeben sind, abstrahiert, ohne sie für eine spätere Verfeinerung ganz außen vor lassen zu müssen. Die vier Kategorien sind in Tabelle 7.2 aufgelistet. Befindet sich eine Sicherungsmaßnahme für ein Schutzziel in der Kategorie 1, werden keine weiteren Maßnahmen benötigt. Kategorie 2 besagt, dass der Schutz zwar gegeben ist, das Schutzziel jedoch nicht vollständig erreicht wird. Z.B. ist das bei der Absicherung der Vertraulichkeit mittels TLS gegeben. Ein außenstehender Angreifer kann zwar nicht auf den Inhalt der Signalisierungspakete zugreifen, Proxys auf den Signalisierungspfad können es dagegen. Kategorie 3 besagt, dass kein Schutz des jeweiligen Ziels gegeben ist. Natürlich muss beachtet werden, dass eine Absicherung nur dann funktioniert, wenn sie richtig konfiguriert wurde bzw. verwendet wird.

Bei der Analyse der Gesamtabsicherung müssen Abhängigkeiten zwischen den Sicherheitsmaßnahmen betrachtet werden. Nicht erfüllte Abhängigkeiten können eine Absicherung wertlos oder unmöglich machen. In Abbildung 7.1 sind diese Abhängigkeiten graphisch dargestellt. Ein eingerahmter Knotenpunkt beschreibt ein Protokoll, das zur Absicherung dient. Von einem solchen Knoten geht eine Kante zu einem nicht umrandeten Knoten, der eine Aufgabe beschreibt, von der das Protokoll abhängt. Die ausgehenden Kanten einer Aufgabe verweisen wieder auf Protokolle, die diese Aufgabe erfüllen. Ist eine der Abhängigkeiten bei einer Kombination aus Sicherheitsmaßnahmen nicht erfüllt, dann bietet das abhängige Protokoll keinen Schutz und ist unabhängig von der Bewertung aus Tabelle 7.1 in die Kategorie 3 einzuordnen. Grundsätzlich gilt hier, dass das schwächste Glied der Abhängigkeitskette die Gesamtsicherheit vorgibt. D.h., falls der Schlüssel im Klartext übertragen wird und nur eine Punkt-zu-Punkt-Absicherung auf einer tieferen Schicht erreicht wird, fällt die Sicherheit des RTP-Stroms in die zweite Kategorie.

In Tabelle 7.3 wurde die Analyse für einige Beispielszenarien durchgeführt. Das erste Szenario ist das zum aktuellen Zeitpunkt bei allen VoIP-Anbietern anzutreffende. Ein Teil der SIP-Anfragen wird vom Proxy des Anbieters mittels HTTP-Digest authentifiziert. Hier wird deutlich, dass bis auf die Authentifizierung der Signalisierung

		SIP				RTP		
Maßnahme	Vorgestellt in	Authentizität	Vertraulichkeit	Integrität	Schlüsseltausch	Vertraulichkeit	Integrität	Authentizität
HTTP-Digest	5.1	B	-	x+				
- erweitert	5.2.1	B	-	x+				
- Predictive Nonces	5.2.2	B	-	-				
SIPS/TLS	5.3	H	x-	x-	+			
DTLS	5.4	H	x-	x-	+			
S/MIME	5.5	B	x+	x+	x			
SDP: k-Attribut	5.6.1				x-			
SDP: Security Descriptions	5.6.2				x-			
MIKEY	5.6.3				x			
SRTP	3.6					x	x	x
IPsec	5.7	x-	x-	x-	x	x	x	x

Legende:

Kat.	Symbol	Bedeutung
		Signalisierung
1	x	Wird vollständig abgesichert
1	B	Benutzerbasierte Authentifizierung
1	H	Hostbasierte Authentifizierung
2	x-	Punkt-zu-Punkt-Absicherung
2	x+	Teilweise Abgesichert
3	-	Wird nicht abgesichert
		Schlüsseltausch
1	x	Ende-zu-Ende-Schlüsseltausch
2,3[1]	x-	Benötigt weitere Sicherheitsmaßnahmen
3	+	Schlüsseltausch auf Signalisierungspfad
3	-	Kein Schlüsseltauschverfahren
		Medienübertragung
1,2,3[1]	x	Wird abgesichert
3	-	Wird nicht abgesichert

[1]Unter berücksichtigung der Abhängigkeiten aus Abbildung 7.1

Tabelle 7.1: Vergleich der Sicherungsmaßnahmen

Kategorie	Beschreibung
1	Vollständige Absicherung
2	Absicherung mit Einschränkungen
3	Keine Absicherung

Tabelle 7.2: Absicherungskategorien

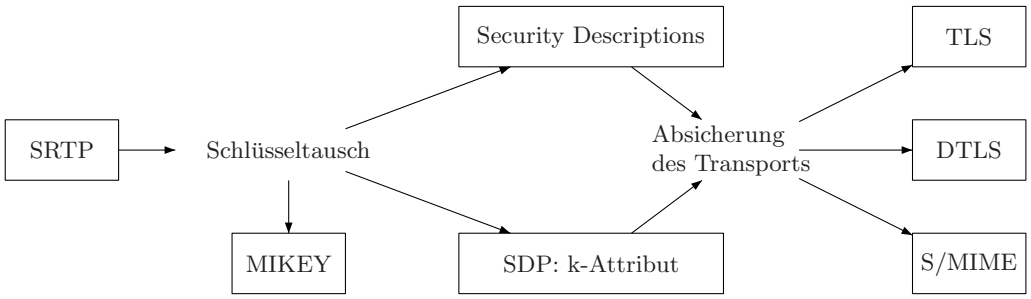


Abbildung 7.1: Abhängigkeiten der Sicherheitsmaßnahmen

keine weitere Sicherheit geboten wird. D.h., ein Angreifer kann Gespräche abhören und die Signalisierungspakete so manipulieren, dass der Medienstrom an einen beliebigen Host gesendet wird. Das zweite Szenario beschreibt eine per SRTP verschlüsselte Medienübertragung, deren Parameter mit Security Descriptions vereinbart werden und deren Signalisierung mit TLS gesichert wird. Hier ist ersichtlich, dass zwar eine Absicherung alle Schutzziele erreicht, diese allerdings nur eingeschränkt ist, weil jeder Proxy auf dem Signalisierungspfad die übertragenen Schlüssel lesen und damit den SRTP-Medienstrom dechiffrieren kann. Eine Fehlkonfiguration wird in Szenario 3 untersucht. Es wird SRTP verwendet, der Schlüsseltausch erfolgt allerdings über einen ungeschützten Signalisierungskanal. Im Gegensatz zum vorherigen Szenario kann hier ein außenstehender Angreifer die Schlüssel mitlesen und hat somit Zugriff auf den Klartext des Medienstroms. Die Fehlkonfiguration führt also dazu, dass die Sicherheit trotz des Einsatz von zusätzlichen Sicherheitsprotokollen der aus Szenario 1 entspricht. Obwohl in Szenario 4 der Transport der kryptographischen Parameter nicht abgesichert wird, bietet MIKEY trotzdem eine Ende-zu-Ende-Sicherung des Schlüsselmaterials. Da aus diesem Grund nur die Endpunkte die Schlüssel kennen, ist der Medienstrom sicherer als in Szenario 2.

Zuletzt wird in Szenario 5 eine Absicherung modelliert, die alle untersuchten Sicherheitsmechanismen enthält, um eine obere Grenze der erreichbaren Sicherheit herzuleiten. Dabei wird deutlich, dass die Vertraulichkeit und Integrität der Signalisierung nur eingeschränkt abgesichert werden können. Die Ursache liegt darin, dass ein Proxy zur Weiterverarbeitung einer Anfrage Teile des SIP-Pakets sowohl lesen als auch modifizieren muss. Den Proxys auf dem Signalisierungspfad muss aus diesem Grund vertraut

Nr.	Szenario	SIP				RTP		
		Authentizität	Vertraulichkeit	Integrität	Schlüsseltausch	Vertraulichkeit	Integrität	Authentizität
1	HTTP-Digest	1	3	2	3	3	3	3
2	HTTP-Digest, TLS, Security Descriptions und SRTP	1	2	2	2	2	2	2
3	HTTP-Digest, SDP-k-Attribut und SRTP	1	3	2	3	3	3	3
4	HTTP-Digest, MIKEY, SRTP	1	3	2	1	1	1	1
5	Alle	1	2	2	1	1	1	1

Tabelle 7.3: Absicherungsszenarien

werden. An dieser Stelle sollte auch beachtet werden, dass sich die vom einfachen HTTP-Digest gebotene Absicherung der Integrität je nach qop-Wert nur auf die erste Zeile eines SIP-Pakets bezieht und so z.B. Angriffe wie die Umleitung des Medienstroms über die Manipulation des SDP-Bodys weiterhin möglich sind.

7.2 Absicherung gegen spezifische Angriffe

Mit Hilfe der hier vorgestellten Sicherheitsmaßnahmen kann ein VoIP-System gegen die Angriffe, die in Kapitel 4 behandelt wurden, abgesichert werden. In Tabelle 7.4 ist dargestellt, welche der Sicherheitsmaßnahmen zu einer Absicherung gegen einen Angriff dienen können. Die Angriffe aus Abschnitt 4.2 werden an dieser Stelle nicht behandelt, weil ihre Ursache in Schwächen des SIP-Protokolls begründet ist und eine Absicherung nur durch eine restriktive Konfiguration oder durch zusätzliche Implementierung von Algorithmen für die Erkennung von Routing-Schleifen erfolgen kann. Es muss beachtet werden, dass die Absicherung in den Kategorien der DoS- und Umleitungs-Angriffe zwischen dem Endpunkt und dem Proxy erfolgt, während für die Angriffe auf dem Signalisierungspfad alle Kommunikationskanäle abgesichert werden müssen. Der Cookie-Ansatz zur Lösung des Problems von DoS-Angriffen über speziell konstruierte Via-Routen wurde hier weggelassen, weil er nur dieses eine spezifische Problem löst.

Bei HTTP-Digest hängt es stark von der Konfiguration ab, welche Angriffe damit abgewehrt werden. Derzeit ist es eine gängige Konfiguration bei VoIP-Anbietern, nur Register- und Invite-Anfragen zu authentifizieren. Außerdem wird der Body eines SIP-Pakets nicht in die Berechnung des Digest einbezogen. Die Authentifizierung von Antworten ist in vielen Implementationen nicht vorhanden. Mit dieser Einstellung bietet HTTP-Digest keinen Schutz gegen die Angriffe. Dass die erweiterte Digest-Authentifizierung nicht vor mehr als den hier vorgestellten Angriffen schützt, bedeutet nicht, dass sie nicht

		DoS				Umleiten			Sig.pfad	
Sicherheitsmaßnahme	Abhören des Medienstroms (8.3)	Gefälschte Antworten (4.1.1)	Gefälschte CANCEL-Anfragen (4.1.2)	Gefälschte BYE-Anfragen (4.1.3)	Via-Manipulation (4.2	mit 3xx-Antwort (4.3.1)	SDP-Manipulation (4.3.2)	Re-Invite (4.3.4)	Kleine Max-Forwards (4.4.1)	Gefälschte BYE-Anfrage (4.4.2)
HTTP-Digest		x[1]		x[2]		x[1]	x[3]	x[4]		x[2]
- erweitert		x[1]		x[2]		x[1]	x[3]	x[4]		x[2]
- Predictive Nonces		x[1]		x[3]		x[1]	x[3]	x[4]		x[2]
SIPS/TLS und DTLS		x	x	x	x	x	x	x		x
S/MIME		x	x	x	x	x	x	x		x
SRTP	x[5]									
IPsec (Signalisierung)		x	x	x	x	x	x	x		x
IPsec (Medienstrom)	x									
Konfiguration	x[6]								x	

[1] Bei beidseitiger Authentifizierung der Antworten auf dem gesamten Signalisierungspfad

[2] Bei beidseitiger Authentifizierung von BYE-Anfragen auf dem gesamten Signalisierungspfad

[3] Wenn SIP-Body mit in die Berechnung des Digest einfließt

[4] Bei beidseitiger Authentifizierung von Invite-Anfragen auf dem gesamten Signalisierungspfad

[5] Falls der Schlüsseltausch gesichert erfolgt

[6] Voreingestellter Pre-Shared-Key

Tabelle 7.4: Absicherung gegen Angriffe

doch einen Sicherheitsgewinn bringt. Der Integritätsschutz der Header ist allerdings nicht relevant bei diesen Angriffen. Weiterhin muss beachtet werden, dass HTTP-Digest auf dem kompletten Signalisierungspfad verwendet werden muss, da sich ein Angreifer auch zwischen zwei Proxys befinden kann. Erfolgt die Authentifizierung einer Anfrage nicht beidseitig, d.h., ein Client überprüft beim Proxy nicht die Authentizität einer empfangenen Anfrage, kann Digest keinen Schutz bieten. Die Ursache dafür ist, dass eine Anfrage so gefälscht werden kann, dass für den Client der Anschein entsteht, die Anfrage hätte den Signalisierungspfad durchlaufen.

TLS und DTLS verhindern alle Angriffe, die auf Packet Injection in Dialogen basieren oder Pakete manipulieren müssen. Das trifft bis auf zwei der hier vorgestellten Angriffe zu. S/MIME bietet unterschiedliche Optionen der Absicherung. Da die Headermanipulation bei den untersuchten Angriffen keine Rolle spielt, reicht der Einsatz von S/MIME bereits aus, um davor zu schützen, da S/MIME grundsätzlich eine Identifikation impliziert. Die Bewertung von IPsec wurde in die Bereiche Signalisierung und Medienübertragung aufgeteilt, weil je nach geschütztem Kommunikationskanal unterschiedliche Angriffe abgewehrt werden.

Der letzte Punkt beschreibt, welche Angriffe durch eine bestimmte Konfiguration verhindert werden können. Einige VoIP-Clients wie z.B. Minisip[1] bieten an, feste Schlüssel für den SRTP-Datenstrom vorzugeben. Das ist für ein normales Kommunikationsmuster, in dem Anrufe zu unbekannten Endpunkten erfolgen, zwar unpraktikabel, weil alle Kommunikationspartner im Vorfeld einen Schlüssel über einen sicheren Kanal austauschen müssten, für kleine Gruppen aber die einfachste Lösung für eine sichere Kommunikation über SRTP. Der Max-Forwards-Angriff auf den Signalisierungspfad kann durch Abweisen von Paketen mit einem niedrigen Wert für Max-Forwards verhindert werden.

[1] `http://minisip.org/`

8 Attack-Tree-Analyse

In diesem Abschnitt werden die bei VoIP verwendeten Protokolle und Komponenten unter den drei Gesichtspunkten der Sicherheit, Verfügbarkeit, Vertraulichkeit und Integrität analysiert. VoIP bietet eine Vielzahl von Angriffsflächen. So setzen die hier behandelten Protokolle SIP und RTP auf bereits bestehenden Protokollen wie IP, TCP und UDP auf und verwenden Protokolle wie DNS und DHCP. Eine weitere Angriffsfläche bieten die verwendeten Soft- und Hardwarekomponenten, deren Implementierungen ebenfalls Fehler enthalten können. Diese Angriffspunkte müssen bei der Analyse eines Gesamtsystems ebenfalls einbezogen werden.

Attack Trees sind ein formales Modell, um Angriffe auf ein System zu finden und diese anhand verschiedener Kriterien zu bewerten [Sch99]. Ein Attack Tree enthält an der Wurzel ein Angriffsziel G, Unterziele G_i in den Knoten und Angriffe A_j als Blätter. Ein Ziel oder Unterziel kann dabei Oder- bzw. Und-Verknüpft sein. Um das Ziel G in Abbildung 8.1 zu erreichen, muss der Angreifer entweder das Unterziel G_1 oder G_2 erreichen. Da G_1 durch den Querstrich als Und-Verknüpft markiert ist, müssen zum Erreichen dieses Unterziels alle Angriffe A_{1j}, $1 \leq j \leq 3$ ausgeführt werden. Zum Erreichen des Unterziels G_2 reicht dagegen ein erfolgreicher Angriff A_{21} oder A_{22}.

Bei komplexeren Systemen wird die Baumdarstellung schnell unübersichtlich, weshalb zur Darstellung eines Attack Trees oft eine hierarchische, nummerierte Listendarstellung verwendet wird. Ein (OR) zeigt dabei an, dass die nachfolgenden Knoten Oder-verknüpft sind. Analog dazu stellt ein (AND) eine Und-Verknüpfung der folgenden Knoten dar. Weil der Fall der Oder-Verknüpfung am häufigsten auftritt, wird diese Markierung in vorliegender Arbeit der Übersichtlichkeit halber weggelassen.

Attack Trees können in anderen Attack Trees wieder verwendet werden. Erstellt beispielsweise ein Unternehmen einen Attack Tree mit dem Angriffsziel *Vertraulichkeit*, kann für einen Unterknoten, der das Unterziel „Buffer Overflow“ enthält, ein bereits ausgearbeiteter Baum [MEL01] zu diesem Angriffsziel angehangen werden. Solche Attack Trees für VoIP zu konstruieren, ist ein Ziel dieser Arbeit. Ein weiteres Beispiel für eine derartige Konstruktion findet sich in [Mob00] Lotus Notes und Domino.

Um die Gesamtsicherheit des Systems zu bewerten, werden jedem Angriff A_j Kosten $c(A_j)$ zugeordnet. Die Kostenfunktion $c(A)$ kann beliebige Zahlenwerte annehmen, wie z.B. die geschätzten Kosten, einen Angriff auf das analysierte System durchzuführen oder auf boolsche Werte, wie z.B. „möglich“ und „unmöglich“ oder „Nur durch internen Angreifer durchführbar“ und „Kann von einem externen Angreifer durchgeführt werden“ beschränkt sein.

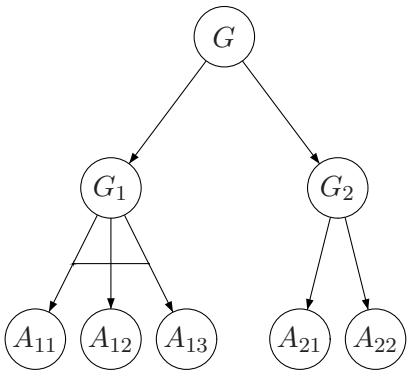


Abbildung 8.1: Aufbau eines Attack Trees

Wurden die Kosten für jeden Angriff bestimmt, können sie nach vorher definierten Regeln nach oben hin zur Wurzel propagiert werden und ergeben die minimalen Kosten, um das Angriffsziel zu erreichen. Für die oben genannten Beispiele wäre eine sinnvolle Propagationsregel für den kontinuierlichen Fall, einem Ziel die Kosten des minimalen Unterziels zuzuordnen, wenn die Kindknoten Oder-verknüpft sind und bei einer Und-Verknüpfung die Kosten aller Kindknoten zu addieren. Für den boolschen Fall „interner Angreifer“/„externer Angreifer“ wäre ein Ziel dann durch einen externen Angreifer durchführbar, wenn bei einer Und-Verknüpfung allen Kindknoten des Zielknotens der Wert „externer Angreifer“ zugeordnet ist, weil die Eigenschaft „externer Angreifer“ als schwächer angesehen wird. Dagegen reicht es bei einer Oder-Verknüpfung aus, wenn ein Ziel durch einen externen Angreifer erreichbar ist. Bei der Propagation können auch mehrere Kostenmaße gleichzeitig betrachtet werden, um so z.B. den günstigsten Angriff eines externen Angreifers zu ermitteln.

In dieser Arbeit wird zusätzlich eine andere Sichtweise verwendet, die es ermöglicht, die Kosten für die Absicherung gegen bestimmte Angriffe und der davon abhängigen Angriffsziele abzuschätzen. Dazu wird jedem Angriff eine Menge der Maßnahmen und Protokollerweiterungen zugeordnet, die vor einem spezifischen Angriff schützen. Die Kosten der günstigsten Absicherung gegen einen Angriff werden als Kostenmaß für diesen übernommen. Nicht immer ist es möglich, sich vor jedem Angriff für ein bestimmtes Ziel zu schützen. Derartige Angriffe können entsprechend markiert und für die Propagation ignoriert werden. Dabei muss jedoch beachtet werden, dass die ausgeschlossenen Angriffe nicht vollständig aus der Bewertung entfallen. Demzufolge kann in so einem Fall nicht die Aussage getroffen werden, dass ein Ziel von einem Angreifer nicht mehr erreichbar ist. Angaben bezüglich einer Verteuerung eines erfolgreichen Angriffs unter Berücksichtigung der zu tätigenden Investitionen können jedoch gemacht werden. Dazu müssen für

die Betrachtung der Angriffskosten nur die Angriffe beachtet werden, die nicht durch Absicherungen betroffen sind. Des Weiteren sollte beachtet werden, dass eine Absicherung gegen einen Angriffspunkt zu einem späteren Zeitpunkt wieder wirkungslos werden kann. Geschehen kann das z.B. durch das Brechen eines kryptographischen Protokolls wie TLS.

8.1 Kostenmaße

Um eine Sicherheitsbewertung durchzuführen, ist es wichtig, sinnvolle Kostenmaße für den jeweiligen Fall zu finden. Kostenmaße wie der entstehende monetäre Schaden können bei einer allgemeinen Betrachtung, wie sie in dieser Arbeit vorgenommen wird, nicht verwendet werden, da diese zu stark vom individuellen Fall abhängen.

Ein einfaches und aussagekräftiges boolsches Maß ist die Möglichkeit eines Angriffs, insbesondere im Zusammenhang mit der Betrachtung, ob dieser Angriff bei Implementierung einer Sicherheitsmaßnahme noch möglich ist. Weiter verfeinert werden kann dies durch das in Tabelle 8.1 dargestellte Kostenmaß. In die Tabelle fließt die Schwierigkeit des Angriffs für den Angreifer mit ein. Der Fokus wurde auf Angriffe gelegt, die über das Netzwerk erfolgen. Bei einem Angriff der Stufe 1 muss der Angreifer Pakete nur empfangen und decodieren können. Der Übergang von Stufe 1 zu 2 stellt auch den Übergang von passiven zum aktiven Angreifer dar. Stufe 2 repräsentiert die blinden Angriffe. Der Angreifer kann einen solchen Angriff ohne weitere Kenntnis über den Inhalt der Kommunikation zwischen den legitimen Kommunikationspartnern durch das Senden von speziell konstruierten Paketen durchführen. Es wird trotzdem vorausgesetzt, dass dieser Angreifertyp Informationen besitzen kann, die nicht in Echtzeit während des Angriffs gewonnen werden müssen. Solche Informationen können z.B. Accountnamen sein. Auch Angriffe, in denen versucht wird, bestimmte Daten zu erraten, wie Brute-Force-Angriffe auf Passwörter, fallen in diese Kategorie. Ein interner Angreifer wird ebenfalls dieser Kategorie zugeordnet, weil er Zugriff auf die benötigten Informationen hat und deswegen keinen Aufwand betreiben muss, um sie zu erhalten. In der nächsten Stufe kommt die Anforderung hinzu, dass für die Durchführung des Angriffs weitere Informationen benötigt werden, die aus der vorhergehenden Kommunikation gewonnen werden können. In der Stufe 4 muss der Angreifer Pakete abfangen und in manipulierter Form senden können. Der Fall, dass ein Angreifer Pakete mitlesen und nicht manipulieren kann, kann z.B. bei einem ungesicherten WLAN auftreten. Das gesendete Paket kommt ohne den Einsatz von Spezialequipment in jedem Fall unverändert beim Empfänger an. Die letzte Stufe ist für Angriffe bestimmt, die physische Eingriffe benötigen oder über die Kommunikation im Netzwerk hinausgehen. Die Stufe 6 sagt aus, dass ein Angriff unmöglich ist. Diese Stufe wird in vorliegender Arbeit nur dann verwendet, wenn gleichzeitig eine ausreichende Absicherung vorausgesetzt wird.

Da mit diesem Maß die Schwierigkeit des einfachsten Angriffs für ein Ziel angegeben

Wert	Kurzbezeichnung
1	Passiv
2	Blinder/Interner Angreifer
3	Packet Injection
4	Paketmanipulation
5	Physischer Zugriff
6	Unmöglich

Tabelle 8.1: Abgestufte Angriffsmöglichkeit

werden soll, wird beim Propagieren bei Oder-Verknüpfung der Kindknoten der minimale Wert für ein Ziel übernommen und bei einer Und-Verknüpfung der maximale Wert.

Analog dazu kann ein Kostenmaß zur Absicherung von Angriffspunkten verwendet werden. Dieses kann ein einfaches boolsches Maß sein, das aussagt, ob eine Absicherung bereits möglich ist oder diese noch entwickelt werden muss. Wegen des hohen Berechnungsaufwands und der zusätzlichen Round-Trips im Protokollablauf führen Sicherungsmaßnahmen meistens zu Verzögerungen beim Rufaufbau und bei der Übertragung von RTP-Paketen. Weil beide Faktoren zu einer Verschlechterung der Dienstqualität führen, ist eine Betrachtung dieser Verzögerungen sinnvoll. Anhand dieser Werte kann eine Kosten-Nutzen-Analyse durchgeführt werden.

8.2 Verfügbarkeit

In Abbildung 8.2 ist ein Attack Tree für die die Verfügbarkeit eines VoIP-Systems dargestellt. Aus Gründen der Übersichtlichkeit wurde hier bewusst darauf verzichtet, konkrete Angriffe komplett in den Blättern des Baumes darzustellen. Die Angriffe werden detailliert in Abschnitt 4 beschrieben, um eine Grundlage zur Berechnung der in Abschnitt 8.1 definierten Kostenmaße zu geben.

Im Folgenden wird auf die Details der einzelnen Angriffspunkte des Attack Trees aus Abbildung 8.2 eingegangen.

1. **Verfügbarkeit der Hardwarekomponenten** Ein VoIP-System besteht im Allgemeinen aus mehreren Hardwarekomponenten. In diesem Ast des Baumes wird das Ziel beschrieben, diese Komponenten außer Betrieb zu setzen, so dass die Verfügbarkeit nicht mehr gegeben ist und entweder ein Teil oder alle Telefonate nicht mehr geführt werden können. Aufgrund der verteilten Struktur einer VoIP-Installation genügt zum Erreichen dieses Ziels ein einziger erfolgreicher Angriff auf eine Komponente. Aus diesem Grund sind die Unterziele Oder-verknüpft.

1.1. **Unterbrechen der Stromversorgung** Da jede Komponente auf eine funktionierende Stromversorgung angewiesen ist, kann die Verfügbarkeit bereits mit einem er-

1. Verfügbarkeit der Hardwarekomponenten
    1.1. Unterbrechen der Stromversorgung
    1.2. Physisches Zerstören von Hardware
        1.2.1. Proxys und Gateways
        1.2.2. Endgeräte
        1.2.3. Trennen von Netzwerkverbindungen
    1.3. Einschleusen korrupter Firmware
        1.3.1. Einschleusen von Paketen in einen TFTP-Download
        1.3.2. Benutzern korrupte Firmware zusenden
    1.4. DoS auf Netzwerkebene
2. Verfügbarkeit von Softwarekomponenten
    2.1. Fluten mit Anfragen.
        2.1.1. Auf Netzwerkebene: TCP SYN-Flood
        2.1.2. Auf Anwendungsebene: Fluten mit SIP-Anfragen
        2.1.3. Auslasten der Netzwerkanbindung (Flooding)
    2.2. Senden von fehlerhaften Paketen
        2.2.1. Fehlerhafte Signalisierungspakete
            2.2.1.1. Überlange Header
            2.2.1.2. Fehlerhafte Längenangaben
            2.2.1.3. Unzulässige Zeichen
            2.2.1.4. Ungeprüfte Verarbeitung von Formatstrings
            2.2.1.5. Unzulässige Formatierung des Pakets
        2.2.2. Fehlerhafte Medienübertragungspakete
            2.2.2.1. Überlange Pakete
            2.2.2.2. Manipulierte Zeitstempel und Sequenznummern
            2.2.2.3. Manipulierte Identifier
            2.2.2.4. Manipulierte Mediendaten
        2.2.3. Fuzzing
        2.2.4. Angriffe auf den IP-Stack
        2.2.5. Ausnutzen von Softwarefehlern
3. Angriffe auf Signalisierungsprotokolle
    3.1. Filtern von Paketen
    3.2. Deregistrierung
    3.3. DoS-Angriffe auf Proxys
        3.3.1. Via-Spoofing
        3.3.2. Massives Forking
    3.4. Angriffe auf die Signalisierung von Telefonaten
        3.4.1. Beantworten von Invite-Anfragen
        3.4.2. Senden einer Cancel-Anfrage während des Rufaufbaus
        3.4.3. Senden einer Bye-Anfrage nach abgeschlossenem Rufaufbau
        3.4.4. Senden eines RTCP-Bye-Pakets in einen RTP-Medienstrom
4. Bestechung von Systemadministratoren und Verantwortlichen

Abbildung 8.2: Attack Tree: Verfügbarkeit

folgreichen Angriff auf die Energieversorgung eingeschränkt werden. Durch USV-Hardware kann die erfolgreiche Durchführung verzögert und bei entsprechender Reaktion ganz verhindert werden. Auch eine sichere Positionierung der energieversorgenden Komponenten kann einen solchen Angriff deutlich erschweren.

1.2. Physisches Zerstören der Hardware Das physische Zerstören von Hardwarekomponenten kann durch einen Angreifer oder durch Naturgewalten herbeigeführt werden und zu langfristigen Ausfällen führen, falls keine Backup-Lösung vorhanden ist. Um sich vor diesen Angriffen zu schützen, sollten VoIP-Proxys und Gateways in speziell abgesicherten Räumen aufgestellt werden.

1.3. Einschleusen korrupter Firmware Gelingt es einem Angreifer, ein Firmwareupdate zu manipulieren, kann eine VoIP-Komponente auf diese Weise außer Betrieb gesetzt werden. Diese Angriffe werden erleichtert, wenn die Firmware nicht signiert ist und die Integrität somit nicht gesichert werden kann.

1.3.1. Packet Injection in einen TFTP-Download TFTP wird häufig von Endgeräten für den Download von Firmware-Updates genutzt. Das Protokoll bietet allerdings keine Möglichkeit, den Dateitransfer abzusichern. Ein Angreifer kann bei einer laufenden Übertragung Pakete mit einer passenden Sequenznummer und beliebigen Daten in den Paketstrom einfügen, was beim Empfänger zu einem korrumpierten Firmware-Image führt. Auch ein blinder Angreifer, der keine Kenntnis über die aktuell zulässige Sequenznummer hat, kann versuchen, die passende Sequenznummer aus einem Raum von 2^{16} möglichen Sequenznummern zu erraten.

1.3.2. Benutzern korrupte Firmware zusenden Führt der Benutzer des VoIP-Geräts die Updates selbst durch, kann, ähnlich zu der Vorgehensweise von Mailwürmern, versucht werden, dem Benutzer eine falsche Firmware zu senden und diesen dazu zu bewegen, sie auf das Gerät zu laden. Um solche Angriffe auszuschließen, sollte nur der Administrator Updates vornehmen und auf die Konfiguration zugreifen können.

1.4. DoS auf Netzwerkebene Es existieren Angriffe auf unteren Protokollschichten, die zum Abschneiden eines Geräts vom Netzwerk führen können. Zum Beispiel können Pakete per ARP-Spoofing zu einem beliebigen Host umgeleitet und dort verworfen werden. Abschnitt 4.5 enthält eine Übersicht über solche Angriffe.

2. Verfügbarkeit von Softwarekomponenten Angriffe auf die Hardware erfordern einen physischen Zugriff auf die Komponenten und sind aus diesem Grunde relativ schwer durchzuführen. Da ein VoIP-System komplexe Software enthält, die während des Betriebs mehrere Protokolle verarbeiten muss, gibt es hier eine Vielzahl von Angriffspunkten. Im Gegensatz zu physischen Angriffen fällt ein Angriff auf die Implementierungen oft nicht so leicht auf. Es kann für den Benutzer so aussehen, als

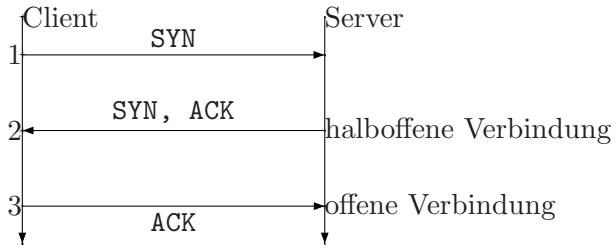


Abbildung 8.3: TCP Three Way Handshake

träte der Fehler bei der Gegenseite auf oder die Gegenseite sei nicht erreichbar, wie es beispielsweise in den Angriffen aus Punkt 3.4. der Fall ist.

2.1.1. TCP SYN-Flood Derzeit wird bei der SIP für den Medientransport immer und für die Signalisierung meistens das verbindungslose UDP als Transportprotokoll verwendet. Trotzdem sind Angriffe auf TCP relevant, da auf VoIP-Geräten oft auch auf TCP basierende Dienste wie HTTP als Konfigurationsinterface angeboten werden.

Ein DoS-Angriff auf einen TCP-Stack ist der sogenannte SYN-Flood. Dazu führt der Angreifer nur den ersten Teil des in Abbildung 8.3 dargestellten Three-Way-Handshake durch. Dies führt beim TCP-Server dazu, dass er weiterhin den Zustand für die *halboffene Verbindung* verwaltet und auf das abschließende Ack des Clients für den vollständigen Verbindungsaufbau wartet. Flutet ein Angreifer nun den TCP-Server mit SYN-Paketen, summieren sich der Verwaltungsaufwand und der Speicherverbrauch schnell auf und es kann letztendlich zu einem Absturz des Geräts kommen, wenn sämtliche Ressourcen belegt sind. Das kann vor allem bei Embedded Devices wie VoIP-Telefonen schnell eintreten.

Als Gegenmaßnahme können SYN-Cookies [Ber] eingesetzt werden. Bei diesem Verfahren wird bis zum vollständigen Aufbau der TCP-Verbingung kein Zustand verwaltet und somit werden keine Ressourcen belegt. Ein Teil der Zustandsinformationen wird dazu mit einer vom Server ausgewählten, geheimen Funktion in der initialen Sequenznummer codiert, um beim abschließenden Ack die Verbindung identifizieren zu können.

Wird TCP nur für Hilfsdienste, wie die Konfiguration benötigt, kann eine Begrenzung der maximalen Anzahl der zulässigen halboffenen Verbindungen festgelegt werden. Damit wird bei einem Angriff nur die Verfügbarkeit des jeweiligen Dienstes eingeschränkt. Allgemein sollten Konfigurationsinterfaces nicht von außen zugänglich sein. Wird an den Netzgrenzen entsprechend gefiltert, sollte ein

solcher Angriff nur noch netzintern möglich sein. Bedingung dafür ist, dass nur UDP für die VoIP-spezifischen Protokolle verwendet wird.

2.1.2. Auf Anwendungsebene: Fluten mit SIP-Anfragen Ähnlich wie bei TCP werden auch auf Anwendungsebene Zustandsinformationen und Datenstrukturen verwaltet. In SIP wird ein Three-Way-Handshake (s. Abbildung 3.3) durchgeführt, wodurch sich bei diesem Protokoll die gleichen Angriffsmöglichkeiten wie bei TCP bieten. Das SIP-Gegenstück zur halboffenen TCP-Verbindung ist ein bis zur endgültigen Antwort durchgeführter Rufaufbau, der vom Client nicht mit einer Ack-Anfrage beantwortet wird.

Das Fluten mit Invite-Anfragen nimmt mehr Ressourcen in Anspruch als ein SYN-Flood-Angriff. Der UAS muss bei einer ankommenden Invite-Anfrage zunächst überprüfen, ob die URL bekannt und erreichbar ist. Gegebenenfalls wird eine Digest-Authentifizierung vorbereitet, was das Generieren eines Zufallswertes (Nonce) beinhaltet. Ein Angreifer kann mit relativ wenig Aufwand eine falsche Antwort darauf erzeugen. Der Server muss diese dagegen überprüfen und damit einen Hash über eine zusammengesetzte Zeichenkette berechnen. Die Funktionsweise der Digest-Authentifizierung ist in Abschnitt 5.1 beschrieben, einige Erweiterungen in 5.2.

Existieren die URLs aus der Invite-Anfrage, entsteht noch weiterer Overhead für den UAS, da dieser einen Dialog verwalten muss und mehrere Antworten generiert werden. Gelingt es einem Angreifer, viele gespoofte Invite-Anfragen zu generieren, die von einem oder mehreren UAS angenommen und weiter verarbeitet werden, kann durch die generierten Antworten und entstehenden RTP-Datenströme ein DDoS-Angriff auf einen beliebigen Host gestartet werden.

Die Angriffe aus Abschnitt 4.2, in denen durch speziell konstruierte SIP-Pakete ein Vielfaches an SIP-Datenverkehr zwischen den Proxys erzeugt werden kann, stellt für einen Angreifer eine besonders effektive Methode des Flooding dar.

2.1.3. Auslasten der Netzwerkverbindung Die Verfügbarkeit kann auch dann beeinträchtigt werden, wenn die Netzwerkverbindungen, die Medienströme transportieren, ausgelastet werden. Die VoIP-Komponenten müssen dafür nicht direkt angegriffen werden. Das Fluten eines Routers mit beliebigen Paketen kann bereits zur Beeinträchtigung der Medienströme, die diesen Router ebenfalls durchlaufen, führen. Zwar werden in so einem Fall Gespräche zustande kommen, die Sprachqualität kann allerdings durch den resultierenden Paketverlust so stark beeinträchtigt werden, dass keine Verständigung mehr möglich ist. Die Ursache hierfür ist, dass die Sprachübertragung gegenüber der Signalisierung relativ empfindlich gegen Paketverlust und Jitter ist. Beides sind Werte, die bei einer stärkeren Auslastung von Netzwerkkomponenten beeinträchtigt werden.

2.2. Senden von fehlerhaften Paketen Dieses Angriffsziel deckt Angriffe auf die Verarbeitung von Paketen und insbesondere beim Parsen des Paketinhalts ab. Bei der Implementierung dieser Programmteile werden Fehler häufig dadurch produziert, dass von korrekten Angaben ausgegangen wird und die übertragenen Daten eine bestimmte Länge nicht überschreiten.

Die Angriffe in den Blättern der Teilziele stellen hier eine Auswahl der häufig anzutreffenden Implementierungsfehler dar und es wird kein Anspruch auf Vollständigkeit erhoben. Im Allgemeinen sollte ein Parser so implementiert sein, dass unzulässige Pakete erkannt und verworfen werden oder zumindest keinen Einfluss auf die Verfügbarkeit des gesamten Geräts haben.

2.2.1. Fehlerhafte Signalisierungspakete Besonders textbasierte Protokolle wie SIP bieten viele Angriffspunkte, weil der Parser aus dem menschenlesbaren Klartext Informationen in interne Datenstrukturen extrahieren muss.

2.2.1.1. Überlange Header Wird intern für Datenfelder nur ein Puffer statischer Größe alloziert und beim Verarbeiten des Pakets nicht geprüft, ob diese Grenzen überschritten werden, kann dies zu einem Angriffspunkt für Buffer-Overflow-Angriffe führen. Ein Angreifer kann dann versuchen, interne Datenstrukturen zu überschreiben, fremden Code einzuschleusen und dadurch die VoIP-Komponente zum Absturz zu bringen.

2.2.1.2. Fehlerhafte Längenangaben Bei SIP wird im Content-Length-Header eine Längenangabe für den Body mitgesendet. Wird anhand dieser Angabe intern Speicher reserviert, kann der Angreifer einen Buffer-Overflow-Angriff provozieren, der bereits in Punkt 2.2.1.1. vorgestellt wurde. Auch unlogische Angaben, wie zum Beispiel negative Werte, können zu einem undefinierten Verhalten der Implementierung und damit zu einem DoS-Angriffspunkt führen.

2.2.1.3. Unzulässige Zeichen Enthält ein Paket Zeichen, die vom Parser nicht erwartet werden, kann dies bei fehlerhafter Implementierung zu einem undefinierten Zustand führen.

2.2.1.4. Ungeprüfte Verarbeitung von Formatstrings Ein Formatstring enthält Informationen über den Datentyp der übergebenen Parameter oder das erwartete Format der Eingabe. Ein Beispiel für eine Funktion aus der Programmiersprache C, die einen Formatstring verwendet, ist `printf("%s", string);`.

Wird ein Formatstring aus Benutzereingaben oder Daten aus Paketen konstruiert, ist es einem Angreifer möglich, damit Speicherbereiche auszulesen und mit Formatangaben, die Speicherbereiche beschreiben, einen DoS-Angriff herbeizuführen. Eine solche Formatangabe in C ist `%n` und schreibt die Anzahl der ausgegebenen Bytes in den Speicher.

2.2.1.5. Unzulässige Formatierung des Pakets Bei der Verarbeitung von Paketen können auch Anomalien, wie fehlende Header oder überflüssige Leerzeilen, zu einem Programmfehler und damit zum Absturz der Komponente führen.

2.2.2. Fehlerhafte Medienpakete RTP-Pakete sind binär codiert und geben für die meisten Felder feste Längen vor. Nur die Größe des Payload und die Liste der CSRC haben eine variable Länge. Aus diesem Grund bieten sich nicht so viele Angriffspunkte für Buffer-Overflow-Angriffe wie beim textbasierten SIP.

2.2.2.1. Überlange RTP-Pakete Ein Angriffspunkt, bei dem ein Buffer-Overflow-Angriff denkbar ist, ist das Senden von überlangen RTP-Paketen. Erwartet die Implementierung nur RTP-Payloads bis zu einer bestimmten Größe, ist ein Überschreiben von dafür nicht vorgesehenen Speicherbereichen möglich.

2.2.2.2. Manipulierte Zeitstempel und Sequenznummern Injiziert ein Angreifer Pakete in einen bestehenden RTP-Paketstrom, deren Zeitstempel und Sequenznummer etwas höher sind als die aktuellen, kann ein Endpunkt die falschen Pakete unter Umständen als authentisch anerkennen und sie anstatt der später eintreffenden echten Pakete verarbeiten. Der nächste Angriff 2.2.2.3. zielt ebenfalls darauf ab, eventuell vorhandene Implementierungsfehler durch Senden von Paketen mit zufälligen SSRC-Identifiern auszunutzen.

2.2.2.4. Fehlerhafte Mediendaten Auch bei der Decodierung des Medienstroms sind Implementierungsfehler denkbar, die zu einem Abbruch des Gesprächs führen können. Werden zur Decodierung Daten aus vorhergehenden Paketen verwendet, kann ein in den RTP-Strom eingefügtes Paket eine längere Passage von Audio-Daten zerstören.

2.2.3. Fuzzing Ein einfacher Angriff, der jedoch bei vielen Implementationen zu einem Absturz führt, ist das sogenannte Fuzzing. Beim Fuzzing werden zufällig generierte Pakete an offene Ports gesendet mit der Absicht, dadurch Fehler aufzudecken.

2.2.4. Angriffe auf den IP-Stack Angriffe auf den IP-Stack sind nicht mehr so stark verbreitet, wie sie es Ende der 90er Jahre noch waren, da die Implementierungen mittlerweile ausgereift sind. Nicht jeder Hersteller von VoIP-Hardware verwendet aber einen fertigen IP-Stack. Darauf lassen OS-Fingerprints [Fyo98] schließen, anhand deren sich bestimmte Geräte eindeutig erkennen lassen. In so einem Fall ist es möglich, dass alte Fehler wieder auftauchen und sich für DoS-Angriffe ausnutzen lassen.

2.2.5. Ausnutzen von Softwarefehlern Bestimmte Fehler bei der Implementierung der Software können dazu führen, dass ein Angreifer erweiterten zugriff bekommt. Beispiele dafür sind:

- Buffer Overflows, die durch zu knappes Bemessen von Puffern im Arbeitsspeicher die Möglichkeit bieten, dafür nicht vorgesehene Speicherbereiche zu beschreiben. Häufig können Buffer Overflows dazu genutzt werden, Programmcode des Angreifers auf dem angegriffenem Gerät auszuführen.
- Command Injection-Angriffe basieren darauf, dass Eingaben, die Befehlstrennzeichen enthalten ohne Überprüfung an interne Funktionen wie einen Shell-Interpreter oder SQL-Server übergeben. Dadurch kann es zur Ausführung unerwünschter Aktionen kommen.

3. Angriffe auf Signalisierungsprotokolle Auch Signalisierungsprotokolle bieten Angriffsmöglichkeiten, wenn sie nicht durch entsprechende Maßnahmen abgesichert werden. Da mit der Absicherung Unannehmlichkeiten wie ein verlängerter Rufaufbau oder eine höhere Belastung der Geräte auftreten, wird die Absicherung häufig unterlassen. In diesem Ast des Attack Trees sind Angriffsmöglichkeiten auf das SIP-Protokoll zusammengefasst.

3.1. Filtern von Paketen Hat der Angreifer entsprechenden Zugriff auf die Netzwerkleitungen, kann er Signalisierungspakete gezielt filtern und damit nur bestimmte Teilnehmer eines VoIP-Systems von der Kommunikation ausschließen.

3.2. Deregistrierung Ein Deregistrierungs-Angriff ist dem Registration Hijacking (Vertraulichkeits-Attack-Tree, 3.1.1.) sehr ähnlich. Im Gegensatz zum Hijacking wird kein neuer Kontakt registriert, sondern der Expires-Header auf Null gesetzt. Das bedeutet für einen Registrar, dass die Registrierung aufgehoben werden soll, womit der Benutzer bis zur nächsten Registrierung nicht mehr erreichbar ist.

3.3. DoS-Angriffe auf Proxys Die Angriffe dieses Ziels wurden in Abschnitt 4.2 behandelt.

3.4. Angriffe auf die Signalisierung von Telefonaten Eine Möglichkeit, Gespräche auf Anwendungsebene zu unterbinden, ist das Einfügen von Paketen in einen SIP-Dialog oder das Senden von RTCP-Paketen. Bei diesen Angriffen werden Anfragen oder Antworten so konstruiert und eingefügt, dass sie dem Endgerät eine authentische Antwort der Gegenseite vortäuschen.

In der Beschreibung der einzelnen Blätter wird nur kurz auf die Angriffe eingegangen. Technische Details werden ausführlich in Abschnitt 4.1 behandelt.

3.4.1. Beantworten von Anfragen Bei diesem Angriff beantwortet der Angreifer Anfragen, bevor es die legitimen Teilnehmer tun können. Dadurch können beliebige Fehlerbedingungen vorgetäuscht werden, die den Benutzer zunächst in dem Glauben lassen werden, die entsprechenden Fehler seien wirklich aufgetreten.

3.4.2. Senden einer Cancel-Anfrage während des Rufaufbaus Dieser Angriff besteht darin, einen angehenden Rufaufbau durch eine eingefügte Cancel-Anfrage zu beenden. Der Vorteil für den Angreifer besteht darin, dass die erste Anfrage gar nicht zum Ziel durchgestellt wird und dieses dadurch auch nichts vom Angriff mitbekommt.

3.4.3. Senden einer Bye-Anfrage nach abgeschlossenem Rufaufbau Hierbei wird der Rufaufbau vollständig durchgeführt. D.h., die Gesprächspartner können bis zur aktiven Durchführung des Angriffs miteinander sprechen. Im Gegensatz zu den zwei vorhergehenden Angriffen gibt es hier keine Wettbewerbsbedingung mit den echten Antworten.

3.4.4. Senden eines RTCP-Bye-Pakets in einen RTP-Medienstrom RTCP sieht eine rudimentäre Signalisierung vor. Unter anderem existiert ein Pakettyp, der die Beendigung einer RTP-Session anfordert. Wird es von einer RTP-Implementation akzeptiert, kann das unabhängig von der SIP-Signalisierung zu einem Gesprächsabbruch führen.

4. Bestechung von Systemadministratoren und Verantwortlichen Auch die Bestechlichkeit der Systemadministratoren muss als Angriffsmöglichkeit betrachtet werden, da diese das System beliebig umkonfigurieren und damit auch außer Betrieb setzen können.

8.3 Vertraulichkeit

Wie in Abschnitt 6.1 bereits beschrieben wurde, umfasst die Vertraulichkeit sowohl die Gesprächsinhalte als auch Signalisierungsinformationen. Der Attack Tree für die Vertraulichkeit deckt beide Teilgebiete ab. Eine weitere Aufteilung wäre zwar denkbar, würde allerdings viele Überschneidungen enthalten.

Ist die Vertraulichkeit der Signalisierungsinformationen nicht gegeben, kann ein Angreifer darüber Informationen erhalten, welche Teilnehmer in welchem Zeitraum miteinander telefoniert haben. Aus SIP-Paketen können auch Informationen gewonnen werden, die ein Angreifer für DoS-Angriffe aus Punkt 3.4. im Verfügbarkeits-Attack Tree nutzen kann, um die falschen Pakete zu generieren.

Im Folgenden werden die einzelnen Punkte des Attack Trees aus Abbildung 8.4 erläutert:

1. Direktes Abhören der Pakete Kontrolliert ein Angreifer die Leitungen, über die die abzuhörenden Datenpakete verkehren, können diese direkt mit einem Netzwerksniffer abgefangen werden. Dieses Ziel umfasst die passiven Angriffe, bei denen vom Angreifer keine Pakete gesendet werden.

1.1. Abhören von Netzwerkleitungen Ist das Netzwerk mit Hubs oder als Bus aufgebaut, so dass grundsätzlich sämtliche Pakete an alle angeschlossenen Geräte verteilt

1. Direktes Abhören der Pakete
    1.1. Abhören auf Netzwerkleitungen
    1.2. Abhören von Funkverbindungen
2. Pakete auf Netzwerkebene umleiten
3. Manipulation der SIP-Signalisierung
    3.1. Anrufe umleiten
        3.1.1. Registration Hijacking
        3.1.2. Einen Anruf umleiten: „302 Moved Temporarily“ senden.
        3.1.3. Alle folgenden Anrufe umleiten: „301 Moved Permanently“ senden.
    3.2. Sprachdatenstrom umleiten
        3.2.1. SDP-Contact manipulieren
        3.2.2. Re-Invite mit neuen RTP-Kontaktdaten senden
4. Kompromittieren von VoIP-Komponenten
    4.1. Ausnutzen von Softwarefehlern
    4.2. Verschaffen von Zugriff durch Login mit Passwörtern (Und-Verknüpft)
        4.2.1. Beschaffen von Passwörtern
            4.2.1.1. Standardpasswörter verwenden
            4.2.1.2. Schwache Passwörter erraten
            4.2.1.3. Abhören von Passwörtern
            4.2.1.4. Social Engineering
        4.2.2. Einloggen
        4.2.3. Stehlen von Endgeräten
5. Physisches Austauschen von Hardwarekomponenten
6. Bestechung von Systemadministratoren und Verantwortlichen

Abbildung 8.4: Attack Tree: Vertraulichkeit

werden, kann jeder Teilnehmer des Netzwerks Sprachdatenströme und Signalisierungspakete abhören. Auch in geswitchten Netzen, in denen dies nicht der Fall ist, können Clients gezielt durch das Anbringen von Hubs abgehört werden.

1.2. Abhören von Funkverbindungen Ein auf Netzwerkebene durch WEP oder WPA abgesichertes WLAN ist von der Funktionsweise her mit einem Hub vergleichbar. Da alle Clients eines Access Points einen Netzwerkschlüssel miteinander teilen, können die verbundenen Clients den Datenverkehr der anderen Teilnehmer mitlesen.

2. Pakete auf Netzwerkebene umleiten In einem geswitchten Netzwerk erhält jeder Client nur die Pakete, die für ihn bestimmt sind. Je nach Qualität der verwendeten Netzwerkhardware ist es möglich, dass ein Angreifer auch nicht für ihn bestimmten Datenverkehr an den eigenen Switch-Port umleiten kann. Diese Angriffe werden durch dieses Angriffsziel repräsentiert und sind in Abschnitt 4.5 zusammengefasst.

3. Manipulation der SIP-Signalisierung Die Angriffe, die dieses Ziel erreichen, wurden bereits bis auf das Registration Hijacking in Abschnitt 4.3 beschrieben.

3.1.1. Registration Hijacking Werden Register-Anfragen nicht authentifiziert, können beliebige Accountdaten im From-Header der Anfrage angegeben werden. Einem Angreifer wird dadurch ermöglicht, die Anrufe des gestohlenen Accounts auf sein Endgerät umzuleiten.

Auch authentifizierte Register-Anfragen, deren Integrität nicht abgesichert ist, können durch einen Man-in-the-Middle-Angreifer abgefangen und deren Kontaktdaten manipuliert werden [PS].

4. Kompromittieren von VoIP-Komponenten Das Kompromittieren von VoIP-Komponenten wie z.B. Proxys ist bei VoIP interessant, da viele Sicherheitsmaßnahmen wie die TLS nur eine Punkt-zu-Punkt-Absicherung ermöglichen. Ist ein Punkt auf der Signalisierungskette kompromittiert, können SIP-Pakete beliebig manipuliert werden. Das kann ohne zusätzliche Ende-zu-Ende-Absicherung, beispielsweise durch S/MIME, nicht festgestellt werden.

Durch eine Kompromittierung von Gateways oder Media-Proxys kann ein Angreifer die Sprachdaten abhören. Während bei Media-Proxys, die keinen Zugriff auf die Nutzdaten benötigen, mit SRTP noch eine Ende-zu-Ende-Sicherheit gegeben werden kann, sind Gateways der Punkt, an dem ein Medienstrom terminiert und damit unverschlüsselt zur Verfügung stehen muss.

Ein Angriff auf ein Endgerät kann von einem Angreifer dazu genutzt werden, gezielt Gespräche eines bestimmten Teilnehmers abzuhören oder Anrufe nachzuvollziehen.

4.1. Ausnutzen von Softwarefehlern Dieses Angriffsziel ist bereits in Punkt 2.2.5. des Attack Trees für Verfügbarkeit erläutert.

4.2. Verschaffen von Zugriff durch Login mit Passwörtern Dieses Ziel beschreibt die Angriffe, die durch die Verwendung von Passwörtern ermöglicht werden. Passwörter werden bei VoIP zum Absichern von Konfigurationsinterfaces und zur Authentifizierung von Anfragen bei der Signalisierung verwendet. Der Angriff besteht aus zwei Und-verknüpften Zielen, dem Beschaffen von Passwörtern und dem eigentlichen Einloggen.

4.2.1.1. Standardpasswörter verwenden Konfigurationsinterfaces werden im Auslieferungszustand von vielen Geräten durch ein Standardpasswort abgesichert. Wird dieses für den späteren Gebrauch nicht geändert, kann sich ein Angreifer durch Beschaffen der oft frei verfügbaren Gerätedokumentation oder Konsultieren einschlägiger Quellen mit Listen solcher Passwörter, Zugriff auf solch ein fehlkonfiguriertes Gerät verschaffen.

4.2.1.2. Schwache Passwörter erraten Oft werden als Passwörter leicht zu merkende Zeichenketten wie Vornamen oder gängige Worte statt zufälliger Passwörter mit Sonderzeichen verwendet. Das eröffnet einem Angreifer die Möglichkeit von Wörterbuch-Angriffen, bei denen insbesondere für HTTP fertige Implementierungen existieren oder die sich mit einfachen Skripten selbst realisieren lassen.

4.2.1.3. Abhören von Passwörtern Besonders Geräte der niedrigen und mittleren Preisklasse verfügen nur über unverschlüsselte Zugänge zu den Konfigurationsinterfaces. Durch Abhören der Leitung und Zuhilfenahme der Techniken des Angriffsziels 2. kann ein Angreifer selbst starke Passwörter abhören und sie anschließend verwenden.

Wird bei Geräten, die TLS verwenden, das Erstellen und Verteilen von Zertifikaten unterlassen, kann keine authentifizierte Verbindung hergestellt werden. Dadurch ist eine vermeintlich sichere Verbindung durch Man-in-the-Middle-Angriffe verwundbar und kann zu einer Offenlegung der Passwörter führen.

Des Weiteren kann das automatisierte Abhören von Passwörtern mit Hilfe von Trojanern zu diesem Angriff gezählt werden.

4.2.1.4. Social Engineering Beim so genannten Social Engineering handelt es sich nicht um technische Schwächen des Systems, sondern der Mensch wird als Schwachstelle ausgenutzt. Das kann auf mehreren Wegen geschehen. Angreifer täuschen z.B. eine falsche Identität vor und gewinnen so das Vertrauen eines Benutzers. Auch Bedrohung und Erpressung sowie Phishing fallen unter diese Angriffsmethode.

4.2.3. Stehlen von Endgeräten Im Gegensatz zur klassischen Telefonie über ein PSTN ist ein VoIP-Account nicht an eine Anschlussbuchse, sondern an ein Gerät gebunden. Wird ein VoIP-Telefon entwendet, dann ist der Angreifer in der Lage, alle Telefonate bis zur Accountsperrung oder Passwortänderung anzunehmen.

Bei Softphones, die auf einem PC laufen, werden Accountdaten häufig in Konfigurationsdateien oder der Windows-Registry abgelegt. Gelingt es einem Angreifer, auf diese Bereiche zuzugreifen, kann der Account unauffälliger als bei einem Hardware-Diebstahl missbraucht werden.

5. **Physisches Austauschen von Hardwarekomponenten** Ist es einem Angreifer nicht möglich, die VoIP-Komponenten zu kompromittieren, kommt unter Umständen ein Ersetzen in Frage. Dazu muss zunächst eine ähnlich konfigurierte Komponente eingerichtet werden, die sich zumindest im Normalfall so verhält wie die ersetzte Komponente und deswegen nicht auffällt.

   Am Beispiel eines SIP-Proxys könnte dies bedeuten, dass grundsätzlich alle Register-Anfragen angenommen werden, weil der Angreifer die Passwörter der Benutzer nicht kennt und dadurch deren Authentizität nicht überprüfen kann. Die Funktion des Proxys und Registrars wäre nicht beeinträchtigt, wenn der Proxy sich selbst nicht gegenüber anderen Komponenten authentifizieren muss.

   Sind die Komponenten physisch zugänglich, kann das Ziel durch das Austauschen dieser Teile erreicht werden.

8.4 Integrität

Zum Angriffsziel Integrität können verschiedene Unterziele zugeordnet werden. Alle in Abschnitt 6.1 vorgestellten Sicherheitsanforderungen sind im Attack Tree aus Abbildung 8.5 enthalten, der im Folgenden erläutert wird:

1. **Absenderkennung fälschen** Dieses Angriffsziel kann verschiedene Intentionen seitens des Angreifers haben. Ein Angreifer, der einen Social Engineering-Angriff durchführen möchte, kann mit einer gefälschten Identität Vertrauen beim Angegriffenen erwecken. Auch bei Spam-over-Internet-Telephony (SPIT) kann wie bei E-Mail-Spam ein Interesse bestehen, die Identität des Anrufers zu fälschen. Ein weiterer Grund Identitäten zu fälschen, besteht darin, auf fremde Kosten zu telefonieren oder Dienste zu nutzen, die anhand der Anruferkennung authentifiziert werden.

   Im Zusammenhang mit dem Übergang in ein PSTN besteht das Problem, dass Diensteanbieter in der klassischen Telefonie oft davon ausgingen, dass die übertragenen Angaben bezüglich der Identität des Anrufers vertrauenswürdig sind. Da die Absenderkennung vom VoIP-Anbieter übernommen wird, kann sich ein Sicherheitsproblem aus der IP-Telefonie so auch auf die klassische Telefonie auswirken.

1.1. **From-Header manipulieren** In SIP-Paketen wird die Identität des Anrufers im From-Header übertragen. Insbesondere bei direkten Anrufen, d.h., ohne zwischen-

1. Absenderkennung fälschen
    1.1. From-Header manipulieren
    1.2. Übernahme eines fremden Accounts
        1.2.1. Stehlen von Endgeräten
        1.2.2. Verschaffen von Zugriff durch Login mit Passwörtern
            1.2.2.1. Standardpasswörter verwenden
            1.2.2.2. Schwache Passwörter erraten
            1.2.2.3. Abhören von Passwörtern
            1.2.2.4. Social Engineering
            1.2.2.5. Einloggen (Und-Verknüpft mit einem der Vorgänger)
        1.2.3. Kompromittierung des Registrars
        1.2.4. IP-Spoofing
    1.3. Kompromittierung von Proxys
2. Manipulation von Accounting-Daten
    2.1. Direkte Manipulation
        2.1.1. Kompromittierung eines Servers, auf dem Accounting-Daten abgelegt sind.
        2.1.2. Manipulation während einer Übertragung
    2.2. Manipulation des Signalisierungsablaufs
        2.2.1. Vortäuschen eines beendeten Gesprächs auf Teilen des Signalisierungspfads.
            2.2.1.1. Senden von Bye-Paketen mit niedrigen Max-Forwards-Werten.
            2.2.1.2. Spoofen von Bye-Paketen, um der Gegenseite ein Gesprächsende vorzutäuschen.
            2.2.1.3. Gefälschte Bye-Transaktion
        2.2.2. DoS-Angriff auf Teile des Signalisierungspfades
        2.2.3. Direct-IP-Calls
3. Integrität des Medienstroms
    3.1. Replay-Angriff
    3.2. Manipulation des Paketinhalts
    3.3. Fälschen der Zieladresse des Medienstroms im SDP-Body.
4. Bestechung von Systemadministratoren und Verantwortlichen

Abbildung 8.5: Attack Tree: Integrität

geschaltete Proxys, kann diese Headerzeile beliebig gewählt werden. Werden Anrufe über einen VoIP-Anbieter getätigt, kann der From-Header bei Anfragen mit HTTP-Digest authentifiziert werden. Das wird in der Praxis bei Register- und Invite-Anfragen häufig so praktiziert.

1.2. Übernahme eines fremden Accounts Dieses Angriffsziel behandelt die gezielte Übernahme eines Accounts. Gelingt einer dieser Angriffe, kann unter der übernommenen Identität und auf Kosten des Inhabers telefoniert werden.

1.2.1. Stehlen von Endgeräten Im Gegensatz zur Telefonie über ein PSTN, in dem ein Account inklusive Telefonnummer an einen physischen Anschluss gebunden ist, wird die Authentifizierung bei VoIP vom Endgerät aus durchgeführt. Dadurch ist es bei VoIP ausreichend, in den Besitz eines Telefons oder der darin gespeicherten Zugangsdaten zu kommen, um einen fremden Account nutzen zu können.

1.2.2. Verschaffen von Zugriff durch Login mit Passwörtern Dieser Punkt entspricht dem Punkt 4.2. aus dem Angriffsbaum für Vertraulichkeit in Abbildung 8.4.

1.2.3. Kompromittierung des Registrars Gelingt es dem Angreifer einen Registrar zu kompromittieren, der normalerweise eine Einheit mit einem SIP-Proxy bildet, kann der Angreifer beliebige Client-Authentifizierungen bestätigen und gegebenenfalls die Passwortdatenbank kopieren.

1.2.4. IP-Spoofing Die Protokolle auf Netzwerkebene bieten in den Grundversionen keinerlei Sicherheit. Beim Internet Protocol können Quell- und Zieladressen beliebig gefälscht werden. Bei Manipulation der Quelladresse nimmt der Angreifer so die Identität eines fremden Hosts an und kann sich dadurch höhere Rechte verschaffen oder bestehende Authentifizierungen nutzen, falls diese nicht für jede Anfrage erneut überprüft werden. Diese auf IP-Spoofing basierenden Angriffe zu unterbinden, liegt im Aufgabengebiet des IP-Zugangsproviders des Angreifers [FS00], was jedoch selten praktiziert wird.

Werden Invite-Anfragen nur anhand der IP-Adresse authentifiziert, kann ein Man-in-the-Middle-Angreifer bei authentifizierten Register-Anfragen die Quell-IP ändern, sich damit für folgende Invite-Anfragen authentifizieren und Telefonate auf Kosten des angegriffenen Teilnehmers führen [PS].

1.3. Kompromittierung von Proxys Bei Anrufen, die über den Proxy eines VoIP-Anbieters initiiert werden, kann beim Verbindungsaufbau die Invite-Anfrage authentifiziert werden. Dies führt beim Angerufenen unter Umständen dazu, dass der angezeigten Identität vertraut wird. Ist nun ein Proxy auf dem Signalisierungspfad kompromittiert, kann an diesem Proxy eine beliebige andere Identität eingesetzt werden.

Dem Angreifer ist es so möglich, sich z.B. mit seinen Zugangsdaten beim SIP-Provider zu authentifizieren und die authentifizierte Identität auf einem auf dem Signalisierungspfad weiter hinten liegenden Proxy durch die von ihm gewünschte Identität zu ersetzen. Aus Sicht des Angerufenen scheint die Identität im From-Header vom ersten Proxy bestätigt zu sein, obwohl sie durch eine andere ersetzt wurde.

2. Manipulation von Accounting-Daten Bei der Manipulation von Accounting-Daten verfolgt ein Angreifer das Ziel, den erfassten Zeitpunkt und die Dauer eines Gesprächs zu verändern. Ziel ist hierbei, den Betreiber eines VoIP-Systems um die Gebühren für kostenpflichtige Gespräche zu betrügen.

2.1. Direkte Manipulation Dieses Angriffsziel beinhaltet die Angriffe, die die sogenannten Call Detail Records (CDR) direkt in der entsprechenden Datenbank manipulieren. Den Zugriff auf CDRs kann ein Angreifer durch Kompromittierung eines entsprechenden Servers (refat-i:kompraccserver) erlangen.

2.1.2. Manipulation während einer Übertragung Eine weitere Möglichkeit diese Daten zu verändern, besteht darin, sie während des Transports über das Netzwerk oder auf Datenträgern zu verändern. Hierbei kann sich ein Angreifer der Mittel bedienen, die bereits in Punkt 2. des Attack Trees für Vertraulichkeit in Abbildung 8.4 behandelt wurden.

2.2. Manipulation des Signalisierungsablaufs Eine indirekte Beeinflussung der Accounting-Informationen kann bei nicht ausreichender Absicherung durch die Manipulation der Signalisierung vorgenommen werden. Diese Angriffe sind insofern einfacher, dass dafür keine Systeme kompromittiert werden müssen. Stattdessen werden Schwächen von Protokollen genutzt, die es bezüglich eines Telefonats ermöglichen, Teilen des Signalisierungspfads einen anderen Zustand als den tatsächlichen vorzutäuschen.

2.2.1. Vortäuschen eines beendeten Gesprächs auf Teilen des Signalisierungspfads. Gelingt es, während eines Telefonats den Proxys auf dem Signalisierungspfad das Gesprächsende zu signalisieren, ohne dass die Endpunkte das Gespräch beenden, können unter Umständen falsche Abrechnungsdaten und damit auch zu niedrige Gebührenabrechnungen entstehen.

Die Angriffe dieses Ziels lassen sich auf zwei Arten durchführen. Bei der ersten Variante können beide Endpunkte miteinander kooperieren und den Signalisierungspfad auf diese Weise umgehen. Die Kooperation kann z.B. durch einen gesonderten Kanal, zusätzliche Informationen im SIP-Paket oder implizit durch einen fest vorgegebenen Ablauf durchgeführt werden. In vielen Fällen ist jedoch nur ein Endpunkt am Angriff beteiligt, weil der gegenüberliegende Endpunkt zum System

des angegriffenen VoIP-Betreibers gehört, wie es beim VoIP-Gateways der Fall ist. Diese Angriffe sind schwieriger zu bewerkstelligen, weil die Nachricht einen Teil des Signalisierungspfades durchlaufen muss, ohne das eigentliche Ziel zu erreichen.

2.2.1.1. Senden von Bye-Paketen mit niedrigen Max-Forwards-Werten. Dieser Angriff ermöglicht es dem Angreifer unter Umständen, die Berechnung der Gesprächsgebühren zu manipulieren. Er wird in Abschnitt 4.4.1 ausführlich behandelt.

2.2.1.2. Spoofen von Bye-Paketen, um einen Gesprächsende der Gegenseite vorzutäuschen. Beim in Abschnitt 4.4 vorgestellten Angriff wird ein Bye-Paket derart konstruiert, dass ein Gesprächsende von der Gegenstelle vorgetäuscht wird. Beim Wiedereintreffen des Bye-Pakets beim eigentlichen Sender wird es einfach verworfen. Ab der Stelle, an der das Bye-Paket auf dem Signalisierungspfad eingefügt wurde, ist es auf dem restlichen Pfad bis zum Endpunkt beendet.

Auch bei diesem Angriff reicht die Beteiligung eines Endpunktes aus, weil das Paket auf dem Signalisierungspfad hinter einem Endpunkt eingefügt wird und damit für ihn verborgen bleibt.

2.2.1.3. Gefälschte Bye-Transaktion Kooperieren beide Endpunkte miteinander, kann eine Bye-Transaktion von einem der Endpunkte gesendet und vom anderen Endpunkt ignoriert werden, um nach dem dadurch eintretenden scheinbaren Gesprächsende auf dem Signalisierungspfad weiter zu kommunizieren. Die zum Schein durchgeführte Transaktion kann vorher abgesprochen sein oder durch zusätzliche Informationen im Paket angezeigt werden.

2.2.2. DoS-Angriff auf Teile des Signalisierungspfades Fällt ein Proxy des Signalisierungspfads aus, gehen unter Umständen Accounting-Informationen verloren oder werden ungenau, da der Proxy keine weiteren Signalisierungspakete mehr empfängt. Angriffe auf die Verfügbarkeit werden durch den Attack Tree in Abbildung 8.2 beschrieben.

2.2.3. Direct IP-Calls Ist die IP-Adresse eines Telefons bekannt, kann ein Anruf bei entsprechender Konfiguration der Endgeräte direkt, unter Umgehung von Proxys, getätigt werden. Dadurch wird der Anruf nicht mehr von den Proxys erfasst. Je nach Einsatzgebiet vom VoIP muss dies keinen Angriff darstellen und kann erwünscht sein.

3. Integrität des Medienstroms Ein Angriff auf die Übertragung der Sprachdaten stellt hohe Anforderungen an den Angreifer. Dieser muss nicht nur jedes gesendete Paket bearbeiten, sondern auch dafür sorgen, dass die beiden Gesprächspartner nichts von der Manipulation mitbekommen. Dazu gehört aus technischer Sicht, dass die Manipulation in Echtzeit erfolgt und damit weder zu großen Verzögerungen noch

zu Paketverlusten führt. Des Weiteren dürfen sich für die beiden Gesprächspartner keine Auffälligkeiten in der Stimmlage und dem Inhalt des Gesprächs ergeben.

Der letzte Punkt führt bei sich bekannten Partnern dazu, dass ein solcher Angriff nur schwer durchzuführen ist. Befindet sich auf einer Seite jedoch eine automatische Ansage, die eventuell Eingaben über DTMF-Töne entgegennimmt, kann ein Angriff für Phishing von Daten wie PINs und TANs genutzt werden, indem der menschliche Benutzer auf der anderen Seite durch entsprechende Ansagen zur Herausgabe der Daten bewegt wird.

3.1. Replay-Angriff Bei einem Replay-Angriff wird eine vorher aufgenommene Sprachaufzeichnung wiedergegeben. Das ist vor allem bei automatischen Ansagen sinnvoll, da der Benutzer bei solchen Diensten immer die gleiche Ansage erwartet, was weitgehend dem Charakter dieses Angriffs entspricht.

3.2. Manipulation des Paketinhalts Bei diesem Angriff werden bereits bestehende Pakete manipuliert. Mögliche Angriffspunkte wurden bereits in Punkt 2.2.2. des Attack Trees für Verfügbarkeit aufgezeigt. Ein Angriff auf die Integrität hat dabei meistens das Ziel, die Verfügbarkeit einzuschränken, weil das sinnvolle Manipulieren der Sprachdaten zu aufwändig und weniger vielversprechend ist.

3.3. Fälschen der Zieladresse des Medienstroms im SDP-Body. Dieser in Abschnitt 4.3 beschriebene Angriff erfolgt während der Signalisierung, wirkt sich aber hauptsächlich auf den Medienstrom aus. Die Verletzung der Integrität stellt hierbei nur ein Nebenziel dar. Das Hauptziel ist die Umleitung der Sprachdaten zu einer anderen IP-Adresse, um sie dort abzuhören oder weiter zu verarbeiten.

8.5 Bewertung der Attack Trees

In diesem Abschnitt werden die Attack Trees aus den vorgehenden Abschnitten mit dem Kostenmaß aus Abschnitt 8.1 bewertet. Für alle Attack Trees wurden Annahmen getroffen und mehrere Szenarien entworfen, in denen unterschiedliche Sicherungsmaßnahmen einbezogen werden. Das erste Szenario entspricht einem Worst-Case, der in der Realität nicht anzutreffen sein wird, es aber ermöglichen soll, eine untere Schranke bezüglich des verwendeten Kostenmaßes anzugeben. Die Bewertung der einzelnen Punkte ergibt sich in den meisten Fällen aus den Beschreibungen der Attack Trees oder Angriffen aus Kapitel 4.

Folgende Szenarien wurden für die Bewertung der Attack-Trees verwendet:

1. Worst-Case: Alle Angriffe, die auf Implementierungsfehlern basieren, sind durchführbar. Insbesondere werden Dialog-Identifier von SIP nicht überprüft. Zwischen den Kommunikationspartnern findet keine Verschlüsselung und Authentifizierung statt.

2. Es wird von einer fehlerfreien Implementation ausgegangen, um die Bewertung um deren Auswirkungen zu bereinigen und ausschließlich VoIP-spezifische Faktoren einfließen zu lassen. Wie derzeit üblich werden Register- und Invite-Anfragen zwischen Endpunkt und Proxy mittels HTTP-Digest authentifiziert.

3. Der gesamte VoIP-Verkehr wird verschlüsselt übertragen. D.h., SIP wird über TLS transportiert, für den Medienstrom wird SRTP und für den Schlüsseltausch MIKEY verwendet. Ein Endpunkt nimmt Signalisierungspakete nur noch über die TLS-Verbindung zum Proxy entgegen.

Zu jedem Attack Tree wird, soweit möglich, eine Bewertung für das VoIP-System Skype durchgeführt. Da sich das Skype-Protokoll und die gesamte Systemarchitektur stark von SIP unterscheidet und viele Details unbekannt sind, werden bei dieser Bewertung zu spezifische Punkte ausgelassen. Weil die Proxys bei SIP ähnliche Funktionen übernehmen wie die Supernodes und der zentrale Authentifizierungsserver bei Skype, wurde bei Angriffszielen, in denen Proxys eine Rolle spielen, auf die entsprechenden Elemente in Skype abgebildet. Bei Skype wird stets von einem realistischen Szenario ausgegangen, in dem angenommen wird, dass das Gerät, auf dem der Skype-Client läuft, gegen nicht-VoIP-spezifische Angriffe ausreichend abgesichert ist. Des Weiteren wird angenommen, dass der Skype-Client fehlerfrei ist, weil hier der Fokus auf die Bewertung der Gesamtarchitektur gelegt wird und Fehler in der Software, wie z.B. Buffer Overflows, nur kurzfristig bestehen und schnell korrigiert werden sollten.

Wie aus dem Attack Tree ersichtlich ist, hängt die Verfügbarkeit eines VoIP-Systems stark von der Verfügbarkeit der IP-Infrastruktur und der Angreifbarkeit der niedrigen Protokollschichten des VoIP-Endgeräts ab. Das wird besonders in Szenario 3 klar, in dem das Auslasten der Netzwerkanbindung und der TCP-SYN-Flood die Verfügbarkeit einschränken können. Beide Probleme können z.B. durch QoS in den Routern und TCP-SYN-Cookies [Ber] gelöst werden.

Eine Anomalie trifft bei den Punkten 2.2.2.2. „Unzulässige Zeitstempel und Sequenznummern“ und 2.2.2.3. „Falsche Identifier“ auf. Werden beide Angriffe zusammengenommen, entspricht das einem blinden Angriff, weil dadurch alle Erkennungsmerkmale einer RTP-Session abgedeckt sind. Da die manipulierten Felder bei den Angriffen auf den Medienstrom explizit voneinander abgegrenzt sind, wurde auch im ersten Szenario vorausgesetzt, dass ein Angreifer die Pakete abhören können muss.

Bei den Packet-Injection-Angriffen auf SIP wird im ersten Szenario davon ausgegangen, dass die Implementation keine Überprüfung der Dialog-Identifier vornimmt. Hier wird deutlich, dass die Nichtauthentifizierbarkeit von Cancel-Angriffen mit HTTP-Digest bis zum Einsatz von TLS ein Sicherheitsproblem darstellt. Da zwischen den Proxys auf dem Signalisierungspfad im zweiten Szenario keine Authentifizierung der ankommenden Anfragen erfolgt, könnte ein Angreifer die Pakete unabhängig von der Absicherung der Kommunikation zwischen Endpunkten und Proxys an einem beliebigen Punkt zwischen zwei Proxys injizieren.

Bei der Bewertung des Angriffs „Massives Forking“, sollte beachtet werden, dass ein Angreifer auf zwei Proxys jeweils zwei Accounts besitzen muss, um diesen Angriff durchführen zu können.

Die Bewertung von Skype bezieht sich bei der Verfügbarkeit von 1.2.1. „Proxys und Gateways“ auf die entsprechenden Signalisierungselemente bei Skype. Hier ist die Bewertung unklar, da sich Skype zum Einloggen zwar mit einem zentralen Server verbindet, im laufenden Betrieb jedoch mit Supernodes kommuniziert, die ein Peer-to-Peer-Netzwerk bilden. Weil in so einer Netzarchitektur ein ausfallender Knoten durch einen anderen ersetzt wird, kann von einer hohen Ausfallsicherheit ausgegangen werden. Ein Angreifer müsste zunächst alle Supernodes lokalisieren und sie physisch zerstören, um diesen Angriff erfolgreich durchführen zu können. Daher kann in dem Fall, in dem der Client bereits Teilnehmer des Skype-Netzwerks ist, davon ausgegangen werden, dass dieser Angriff unmöglich ist. Trotzdem muss beachtet werden, dass Skype eine zentrale Komponente zum Zugang in dieses Netz verwendet und hier das gleiche Potential wie bei einer SIP-Infrastruktur besteht. Die Konsequenzen eines Ausfalls des oder der zentralen Server wären sogar weitreichender, weil sich weltweit kein Skype-Client in das Netzwerk einloggen könnte.

Aus der Bewertung des Attack Trees für die Vertraulichkeit in Tabelle 8.3 kann ein Angreifer im ersten und zweiten Szenario Medienströme passiv abhören. Erst der Einsatz von SRTP mit einem abgesicherten Ende-zu-Ende-Schlüsseltausch garantiert die Vertraulichkeit der Gesprächsinhalte. Auch die Signalisierung bietet im zweiten Szenario, in dem die Authentifizierung zwischen Endpunkt und Proxy mit HTTP-Digest erfolgt, Angriffspunkte, die durch Packet Injection in einen SIP-Dialog ausgenutzt werden können. Hier verschafft erst die vollständige Absicherung des Signalisierungspfads mit TLS unter der Voraussetzung, dass alle Proxys vertrauenswürdig sind, Abhilfe.

Die Bewertung des Attack Trees für Integrität ist in Tabelle 8.4 abgebildet. Hier wird wieder ersichtlich, dass die Gefahr bei stärkerer Absicherung auch von menschlichen Faktoren, wie der Vergabe von schwachen Passwörtern, abhängt. Ein weiteres Ergebnis ist, dass Accounting-Daten auch noch in Szenario 2 durch Beeinflussung der Signalisierung mit dem Angriff aus Abschnitt 4.4.2 möglich ist.

Die Attack Trees wurden mit dem vorliegendem Kostenmaß in den meisten untersuchten Szenarien mit 2 bewertet. Das bedeutet, dass das Angriffsziel durch Angriffe, die höchstens allgemeine Kenntnisse wie z.B. Benutzernamen erfordern, erreicht werden kann. Betrachtet man die Bewertung für den Attack Tree für Vertraulichkeit in Tabelle 8.3 genauer, stellt man fest, dass diese Gesamtbewertung in Szenario 3 einzig durch Angriffe auf schwache und Standardpasswörter zurückzuführen ist. Da ein Benutzer VoIP-Hardware wie ein normales Telefon ohne vorherige Autorisierung verwendet und die Login-Daten von Administratoren vergeben werden können, ist es relativ einfach, diese Art von Angriffen zu verhindern. Für das besagte Szenario bedeutet das, dass die Gesamtbewertung des Attack Trees auf 5 steigt und der Angriff nur noch mit physischem Zugriff durch das Austauschen der Hardwarekomponenten möglich ist.

Angriffsziel/Angriff Szenario ⟶	1	2	3	S
Gesamtbewertung des Attack-Trees	2	2	2	2
1. Verfügbarkeit der Hardwarekomponenten	2	5	5	5
1.1. Unterbrechen der Stromversorgung	5	5	5	5
1.2. Physisches Zerstören von Hardware	5	5	5	5
1.2.1. Proxys und Gateways	5	5	5	5-6
1.2.2. Endgeräte	5	5	5	5
1.2.3. Trennen von Netzwerkverbindungen	5	5	5	5
1.3. Einschleusen korrupter Firmware	3	6	6	6
1.3.1. Packet Injection in einen TFTP-Download	3	6	6	6
1.3.2. Benutzern korrupte Firmware zusenden	5	6	6	6
1.4. DoS auf Netzwerkebene	2	6	6	6
2. Verfügbarkeit von Softwarekomponenten	2	2	2	2
2.1. Fluten mit Anfragen.	2	2	2	2
2.1.1. Auf Netzwerkebene: TCP SYN-Flood	2	6	6	6
2.1.2. Auf Anwendungsebene: Fluten mit SIP-Anfragen	2	2	2	6
2.1.3. Auslasten der Netzwerkanbindung	2	2	2	2
2.2. Senden von fehlerhaften Paketen	2	6	6	6
2.2.1. Fehlerhafte Signalisierungspakete	2	6	6	6
2.2.1.1. Überlange Header	2	6	6	
2.2.1.2. Fehlerhafte Längenangaben	2	6	6	
2.2.1.3. Unzulässige Zeichen	2	6	6	
2.2.1.4. Ungeprüfte Verarbeitung von Formatstrings	2	6	6	
2.2.1.5. Unzulässige Formatierung des Pakets	2	6	6	
2.2.2. Fehlerhafte Medienübertragungspakete	3	6	6	6
2.2.2.1. Überlange Pakete	3	6	6	
2.2.2.2. Unzulässige Zeitstempel und Sequenznummern	3	6	6	
2.2.2.3. Falsche Identifier	3	6	6	
2.2.2.4. Fehlerhafte Mediendaten	3	6	6	
2.2.3. Fuzzing	2	6	6	6
2.2.4. Angriffe auf den IP-Stack	2	6	6	6
2.2.5. Ausnutzen von Softwarefehlern	2	6	6	6
3. Angriffe auf Signalisierungsprotokolle	2	2	2	4
3.1. Filtern von Paketen	4	4	4	4
3.2. Deregistrierung	2	6	6	6
3.3. DoS-Angriffe auf Proxys	2	2	2	
3.3.1. Via-Spoofing	2	2	6	
3.3.2. Massives Forking	2	2	2	
3.4. Angriffe auf die Signalisierung von Telefonaten	2	3	6	6
3.4.1. Beantworten von Invite-Anfragen	2	3	6	
3.4.2. Senden einer Cancel-Anfrage während des Rufaufbaus	2	3	6	
3.4.3. Senden einer Bye-Anfrage nach abgeschlossenem Rufaufbau	2	3	6	
3.4.4.Senden eines RTCP-Bye-Pakets in einen RTP-Medienstrom	2	3	6	
4. Bestechung von Systemadministratoren und Verantwortlichen	5	5	5	5

Tabelle 8.2: Bewertung des Attack Trees für Verfügbarkeit

Angriffsziel/Angriff Szenario ⟶	1	2	3	S
Gesamtbewertung des Attack-Trees	1	1	5	5
1. Direktes Abhören der Pakete	1	1	6	6
1.1. Abhören auf Netzwerkleitungen	1	1	6	6
1.2. Abhören von Funkverbindungen	1	1	6	6
2. Pakete auf Netzwerkebene umleiten	2	6	6	6
3. Manipulation der SIP-Signalisierung	2	3	6	6
3.1. Anrufe umleiten	2	3	6	6
3.1.1. Registration Hijacking	2	6	6	6
3.1.2. Einen Anruf umleiten: „302 Moved Temporarily“ senden.	2	3	6	
3.1.3. Alle folgenden Anrufe umleiten: „301 Moved Permanently“ senden.	2	3	6	
3.2. Sprachdatenstrom umleiten	2	3	6	6
3.2.1. SDP-Contact manipulieren	4	4	6	
3.2.2. Re-Invite mit neuen RTP-Kontaktdaten senden	2	3	6	
4. Kompromittieren von VoIP-Komponenten	2	6	6	6
4.1. Ausnutzen von Softwarefehlern	2	6	6	6
4.2. Verschaffen von Zugriff durch Login mit Passwörtern (Und)	2	6	6	6
4.2.1. Beschaffen von Passwörtern	1	6	6	6
4.2.1.1. Standardpasswörter verwenden	2	6	6	6
4.2.1.2. Schwache Passwörter erraten	2	6	6	6
4.2.1.3. Abhören von Passwörtern	1	6	6	6
4.2.2.Einloggen	2	2	2	2
4.2.3. Diebstahl	5	5	5	5
5. Physisches Austauschen der Hardwarekomponenten	5	5	5	5
6. Bestechung von Systemadministratoren und Verantwortlichen	5	5	5	5

Tabelle 8.3: Bewertung des Attack Trees für Vertraulichkeit

Angriffsziel/Angriff — Szenario ⟶	1	2	3	S
Gesamtbewertung des Attack-Trees	2	2	2	2
1. Absenderkennung fälschen	2	2	2	2
1.1. From-Header manipulieren	2	6	6	
1.2. Übernahme eines fremden Accounts	2	2	2	2
1.2.1. Stehlen von Endgeräten	5	5	5	5
1.2.2. Verschaffen von Zugriff durch Login mit Passwörtern	2	2	2	2
1.2.2.1. Standardpasswörter verwenden	2	6	6	6
1.2.2.2. Schwache Passwörter erraten	2	2	2	2
1.2.2.3. Abhören von Passwörtern	1	6	6	6
1.2.2.5. Einloggen (Und-Verknüpft mit einem der Vorgänger)	2	2	2	2
1.2.3. Kompromittierung des Registrars	2	6	6	6
1.2.4. IP-Spoofing	2	6	6	6
1.3. Kompromittierung von Proxys	2	6	6	6
2. Manipulation von Accounting-Daten	2	2	6	6
2.1. Direkte Manipulation	2	6	6	6
2.1.1. Kompromittierung eines Servers, auf dem Accounting-Daten abgelegt sind.	2	6	6	6
2.1.2. Manipulation während einer Übertragung	4	6	6	6
2.2. Manipulation des Signalisierungsablaufs	2	2	6	6
2.2.1. Vortäuschen eines beendeten Gesprächs auf Teilen des Signalisierungspfads.	2	2	6	6
2.2.1.1. Senden von Bye-Paketen mit niedrigen Max-Forwards-Werten.	2	6	6	
2.2.1.2. Spoofen von Bye-Paketen, um der Gegenseite ein Gesprächsende vorzutäuschen.	2	2	6	
2.2.1.3. Gefälschte Bye-Transaktion	2	6	6	
2.2.2. DoS-Angriff auf Teile des Signalisierungspfades	2	2	6	6
2.2.3. Direct-IP-Calls	2	2	6	6
3. Integrität des Medienstroms	2	4	6	6
3.1. Replay-Angriff	2	4	6	6
3.2. Manipulation des Paketinhalts	4	4	6	6
3.3. Fälschen der Zieladresse des Medienstroms im SDP-Body.	4	4	6	6
4. Bestechung von Systemadministratoren und Verantwortlichen	5	5	5	5

Tabelle 8.4: Bewertung des Attack Trees für Integrität

Das zeigt, dass bei einem bewerteten Attack Tree nicht nur das Gesamtergebnis zur Beurteilung der Sicherheit herangezogen werden sollte, weil insbesondere bei boolschen oder diskreten, endlichen Kostenmaßen wenige Angriffe zu einem deutlich schlechteren Ergebnis führen können, selbst wenn der Schutz vor den Angriffen einfach zu bewerkstelligen ist. Anderseits können Annahmen, wie z.B. die Fehlerfreiheit von Software, zu einer starken Verbesserung der Gesamtbewertung führen. Diese Annahmen wurden in dieser Arbeit trotzdem getroffen, um die Bewertung auf VoIP-spezifische Faktoren zu fokussieren.

Bei Angriffszielen ist es zwar möglich, einen vollständigen Attack Tree auszuarbeiten, bei der weiteren Verfeinerung kann von einem Attack Tree kein Anspruch auf Vollständigkeit erhoben werden, weil bis dahin unbekannte Angriffe nicht erfasst werden.

Insgesamt zeigt die Bewertung aller Attack Trees, dass sich das Sicherheitsproblem nach der Absicherung der VoIP-spezifischen Komponenten mit den in Kapitel 5 beschriebenen Protokollen auf die Infrastruktur, Softwarefehler und menschliches Versagen verlagert.

Skype scheint in der Bewertung in einigen Punkten besser abzuschneiden als VoIP über SIP. Es muss jedoch beachtet werden, dass die Bewertung von Skype unter der Annahme durchgeführt wurde, dass die Verschlüsselung des Datenverkehrs die gleiche Sicherheit wie TLS bietet. Skype ist proprietär, und die verwendeten kryptographischen Protokolle wurden bisher nur in einer Studie [Ber05] untersucht. Implementierungen von kryptographischen Algorithmen, die nicht als frei verfügbarer Quellcode vorliegen, der von einer großen Anzahl von Experten untersucht werden kann, gelten im Allgemeinen als nicht vertrauenswürdig, weil die Gefahr von eingebauten Hintertüren und Implementierungsfehlern besteht, die den Schutz der behandelten Daten beeinträchtigen können.

9 Zusammenfassung und Ausblick

In diesem Kapitel werden die Ergebnisse der Arbeit zusammengefasst und ein Ausblick auf weitere Entwicklungen und Ergebnisse gegeben.

9.1 Zusammenfassung

Diese Arbeit beschäftigte sich mit der Sicherheit von VoIP. Nachdem am Anfang ein Überblick über die Themenbereiche der VoIP-Sicherheit gegeben und die Grundlagen der VoIP-Protokolle SIP/SDP, RTP und Skype erläutert wurden, wurde zunächst eine technisch detaillierte Übersicht über konkrete Angriffe auf die Signalisierung mit SIP/SDP gegeben. Es wurden sowohl DoS-Angriffe behandelt, die ausgewählte Gespräche beenden, als auch solche, die einen SIP-Proxy belasten und zu dessen Ausfall führen können. Auch Angriffe auf die Vertraulichkeit, mit denen Gespräche oder RTP-Medienströme umgeleitet und damit abgehört werden können, wurden vorgestellt. Zum dritten Schutzziel, der Integrität, wurde ebenfalls ein bekannter Angriff vorgestellt und ein neuer Angriff entwickelt und implementiert. Mit Hilfe dieser Angriffe könnte es unter bestimmten Umständen möglich sein, die Abrechnung von Gesprächsgebühren zu manipulieren.

Das Gegenstück zu den Angriffen bilden die Sicherungsmaßnahmen, die ebenfalls vorgestellt wurden. Zusätzlich zu den allgemein bekannten Protokollen TLS und dessen Weiterentwicklung DTLS, S/MIME und MIKEY wurden Ansätze vorgestellt, die weniger bekannt sind, wie die beiden Erweiterungen zu HTTP-Digest. Diese Erweiterungen könnten in Geräten, die nicht die notwendigen Ressourcen für komplexe kryptographische Berechnungen haben, mit relativ kleinen zusätzlichen Kosten zu einem Sicherheitsgewinn führen. Bei dem Ansatz der Predictive Nonces wurde jedoch festgestellt, dass dieser keinen Schutz der Integrität bieten kann.

Im folgenden Teil wurden die vorgestellten Sicherheitsmechanismen darauf hin untersucht, welche Schutzziele sie erreichen und in welcher Qualität sie das tun. Die Untersuchung unterscheidet sich insofern von der in [AAGea05] vorgenommenen, dass diese Verfeinerung erfolgte. Für unterschiedliche Kombinationen der Sicherheitsmaßnahmen wurde unter Beachtung von Abhängigkeiten untersucht, welche Gesamtsicherheit sie bieten und dafür eine Obergrenze angegeben. Das Ergebnis ist, dass die Sicherheitsziele Vertraulichkeit und Integrität der Signalisierung mit SIP nicht vollständig erfüllt werden können, weil Proxys SIP-Nachrichten bearbeiten müssen. Außerdem wurde untersucht, vor welchen der in dieser Arbeit vorgestellten Angriffen die Sicherheitsmaßnahmen einen Schutz bieten.

Weiter wurden für die drei Sicherheitsziele Verfügbarkeit, Vertraulichkeit und Integrität Attack Trees konstruiert. Die Attack Trees wurden mit einem vorher hergeleiteten Kostenmaß, das die Schwierigkeit eines Angriffs kategorisiert bewertet. Es wurde eine Erweiterung der Bewertung von Attack Trees beschrieben, die eine strukturierte Kosten-Nutzen-Analyse für die Lösung von Sicherheitsproblemen ermöglicht. Als Ergebnis der Bewertung kann festgehalten werden, dass sich die Sicherheitsprobleme von VoIP bei richtiger Absicherung mit Digest-Auhtentifizierung, TLS-gesicherten Signalisierungsverbindungen und einem Schlüsseltausch über MIKEY auf die unteren Protokollschichten verlagern.

9.2 Ausblick

VoIP bietet bereits heute eine Auswahl von Sicherungsmöglichkeiten, die eine Ende-zu-Ende-Verschlüsselung der Sprachdaten erlauben und damit einen Schutz bieten, der in der klassischen Telefonie nur mit teurer Zusatzhardware gegeben werden kann. Beim Schutzziel der Verfügbarkeit und Integrität sieht es dagegen anders aus. Ohne zusätzliche Maßnahmen wie der Abwehr von DDoS-Angriffen und dem Einsatz von QoS auch auf der Seite des IP-Providers kann ein VoIP-System mit einfachsten Mitteln, z.B. dem Flooding mit IP-Paketen, gestört werden.

Angriffe wie die, die in Abschnitt 4.2 beschrieben wurden, zeigen, dass auch das Signalisierungsprotokoll SIP bei einer standardkonformen Implementierung Angriffspunkte bietet, bei denen ein Angreifer mit kleinem Aufwand eine starke Auslastung der für VoIP lebensnotwendigen Infrastrukturkomponenten provozieren kann. Beide Angriffe basieren darauf, dass die Erkennung von Routing-Schleifen nur mit dem Herunterzählen eines Hop-Zählers erfolgt. Hier ist eine Entwicklung von Lösungen gefragt, die auf der einen Seite die frühzeitige Erkennung solcher Schleifen erlauben, auf der anderen Seite aber auch längere Signalisierungspfade ermöglichen.

Ein weiteres Problem von SIP ist, dass den Proxys auf dem Signalisierungspfad in jedem Fall vertraut werden muss, weil bestimmte Header bei jeder Weiterleitung verändert werden können und deswegen keine Ende-zu-Ende-Absicherung erfolgen kann. Hier wäre die Erforschung alternativer Routingmechanismen interessant, die nicht diese Forderung stellen.

Insgesamt kann gesagt werden, dass ein sicherer VoIP-Betrieb möglich ist, die dafür notwendigen Features von Herstellern der VoIP-Hardware und VoIP-Providern jedoch nicht angeboten werden. Hier sollte eine Sensibilisierung der Unternehmen und Endbenutzer erfolgen um langfristig einen Sicherheitsstandard durchzusetzen, der über dem der klassischen Telefonie liegt.

Literaturverzeichnis

[AAGea05] A. Adelsbach, A. Alkassar, K.-H. Garbe, and et al. *VoIPSEC - Studie zur Sicheheit von Voice over Internet Protocol.* BSI, 2005.

[ABW05] Flemming Andreasen, Mark Baugher, and Dan Wing. *Draft: Session Description Protocol Security Descriptions for Media Streams.* IETF Network Working Group, September 2005.

[ACB+05] Humberto Abdelnur, Vincent Cridlig, Jerome Bourdellon, Radu State, and Olivier Festor. VoIP Security Management. Technical report, Madynes, Juli 2005.

[ACL+04] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman. *RFC 3830: MIKEY: Multimedia Internet KEYing.* IETF Network Working Group, August 2004.

[ACL+05] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman. *Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP).* IETF Network Working Group, Juni 2005.

[AEHP99] Arango, Elliott, Huitema, and Pickett. *RFC 2705: Media Gateway Control Protocol.* IETF Network Working Group, Oktober 1999.

[AHM+05] J. Albers, B. Hahn, S. McGann, S. Park, and R. Zhu. An Analysis of Security Threats and Tools in SIP-based VoIP Systems. Technical report, University of Colorado, 2005.

[Aki02a] Ofir Akin. Security Threats to IP Telephony-Based Networks. Technical report, Sys-Security Group, Dezember 2002.

[Aki02b] Ofir Akin. The Trivial Cisco IP Phones Compromise - Security analysis of the implications of deploying Cisco Systems' SIP-based IP Phones model 7960. Technical report, Sys-Security Group, September 2002.

[Aki02c] Ofir Akin. Why E.T. Can't Phone Home? Security Risk Factors with IP Telephony based Networks. Technical report, Sys-Security Group, November 2002.

[ASRS01] Ralf Ackermann, Markus Schumacher, Utz Roedig, and Ralf Steinmetz. Vulnerabilities and Security Limitations of Current IP Telephony Systems. In *Proceedings of the IFIP TC6/TC11 International Conference on Communications and Multimedia Security Issues of the New Century*, 2001.

[AW05] Flemming Andreasen and Dan Wing. *Draft: Security Preconditions for Session Description Protocol Media Streams.* IETF Network Working Group, Oktober 2005.

[BBR02] Roberto Barbieri, Danilo Bruschi, and Emilia Rosti. Voice over IPsec: Analysis and Solutions. In *Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC02)*, page 261 ff., 2002.

[Bel89] S. M. Bellovin. Security Problems in the TCP/IP Protocol Suite. *Computer Communication Review*, 19(2):32–48, April 1989.

[BEOV05] Johan Bilien, Erik Eliasson, Joachim Orrblad, and Jon-Olov Vatn. Secure VoIP: call establishment and media protection. Technical report, KTH Royal Institute of Technology, Stockholm, Juni 2005.

[Ber] Daniel J. Bernstein. Syn cookies. `http://cr.yp.to/syncookies.html`.

[Ber05] Tom Berson. Skype Security Evaluation. Technical report, Anagram Laboratories, Oktober 2005.

[Bil03] Johan Bilien. Key agreement for secure voice over ip. Master's thesis, KTH Royal Institute of Technology, Stockholm, Dezember 2003.

[BL01] Borella and Lo. *RFC 3102: Realm Specific IP: Framework.* IETF Network Working Group, October 2001.

[BLGT01] Borella, Lo, Grabelsky, and Taniguchi. *RFC 3103: Realm Specific IP: Framework.* IETF Network Working Group, October 2001.

[BMN+04] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. *RFC 3711: The Secure Real-time Transport Protocol (SRTP).* IETF Network Working Group, März 2004.

[BS04] Salman Baset and Henning Schulzrinne. An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. Technical report, Department of Computer Science, Columbia University New York, September 2004.

[Cab03] Israel Caballero. Secure mobile voice over ip. Master's thesis, KTH Royal Institute of Technology, Stockholm, Juni 2003.

[CGR+00] Cuervo, Greene, Rayhan, Huitema, Rosen, and Segers. *RFC 3015: Megaco Protocol Version 1.0.* IETF Network Working Group, November 2000.

[Cis02] Cisco Systems. *Security in SIP-based Networks*, 2002.

[CJ99] S. Casner and V. Jacobson. *RFC 2508: Compressing IP/UDP/RTP Headers for Low-Speed Serial Links.* IETF Network Working Group, Februar 1999.

[CM05a] Feng Cao and Saadat Malik. Requirement analysis for ip-based government emergency telephony service. Technical report, Critical Infrastructure Assurance Group, Cisco Systems, 2005.

[CM05b] Feng Cao and Saadat Malik. Security Analysis and Solutions for Deploying IP Telephony into the Critical Infrastructure. In *Proceedings of IEEE/Create-Net Workshop on Security and QoS in Communication Networks*, 2005.

[CMR02] G. Camarillo, W. Marshall, and J. Rosenberg. *RFC 3312: Integration of Resource Management and Session Initiation Protocol (SIP).* IETF Network Working Group, Oktober 2002.

[Col05] Mark Collier. Voice over IP Denial of Service. Technical report, SecureLogix Corporation, Mai 2005.

[Cve04] Nedeljko Cvejic. *Algorithms for Audio Watermarking and Steganography.* PhD thesis, University of Oulu, 2004.

[DA99] T. Dierks and C. Allen. *RFC 2246: The TLS Protocol, Version 1.0.* IETF Network Working Group, Januar 1999.

[Def04] Defense Information Systems Agency. *Voice over Internet Protocol Security Technical Implementation Guide*, Januar 2004.

[Des] Fabrice Desclaux. *Skype uncovered - Security study of Skype.* EADS CCR/STI/C.

[DvOW92] Whitfield Diffie, Paul C. van Oorschot, and Michael J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2(2):107–125, 1992.

[FHBH+97] Franks, Hallam-Baker, Hostetler, Sink, Leach, Luotonen, and Stewart. *RFC 2069: An Extension to HTTP : Digest Access Authentication.* IETF Network Working Group, Januar 1997.

[FHBH+99] Franks, Hallam-Baker, Hostetler, Lawrence, Leach, Luotonen, and Stewart. *RFC 2617: HTTP Authentication: Basic and Digest Access Authentication.* IETF Network Working Group, Juni 1999.

[FS00] P. Ferguson and D. Senie. *RFC 2827: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing.* IETF Network Working Group, Mai 2000.

[Fyo98] Fyodor. Remote OS detection via TCP/IP Stack FingerPrinting. In *Phrack Magazine*, volume 54. ANPASSEN, Oktober 1998.

[GS] Debbie Greenstreet and Sophia Scoggins. *Building Residential VoIP Gateways: A Tutorial, Part FOur: VoIP Security Implementation.*

[HJ98] Handley and Jacobsen. *RFC 2327: Session Description Protocol.* IETF Network Working Group, April 1998.

[HPFS02] R. Housley, W. Polk, W. Ford, and D. Solo. *RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.* IETF Network Working Group, April 2002.

[HRM03] Huitema, Rosenberg, and Mahy. *Traversal Using Relay NAT (TURN).* IETF Network Working Group, October 2003.

[HRWM00] Huitema, Rosenberg, Weinberger, and Mahy. *RFC 3489: Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs).* IETF Network Working Group, November 2000.

[HSSR99] Handley, Schulzrinne, Schooler, and Rosenberg. *RFC 2543: Session Initiation Protocol.* IETF Network Working Group, März 1999.

[HSSR02] Handley, Schulzrinne, Schooler, and Rosenberg. *RFC 3261: Session Initiation Protocol.* IETF Network Working Group, Juni 2002.

[Iac02] Luigi Lo Iacono. Rote Telefone. *iX*, 5:118, 2002.

[JP05] C. Jennings and J. Peterson. *Draft: Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP).* SIP WG, Oktober 2005.

[JS03] Wenyu Jiang and Henning Schulzrinne. Assessment of VoIP Service Availability in the Current Internet. In *Passive and Active Measurement Workshop*, 2003.

[KA98] S. Kent and R. Atkinson. *RFC 2401: Security Architecture for the Internet Protocol.* IETF Network Working Group, November 1998.

[Kle03] Alan Klein. Security Analysis: Traditional Telephony and IP Telephony. Technical report, SANS Institute, 2003.

[KM05] Infonetics Research Kevin Mitchell. Carrier voip equipment market up 40 `http://www.infonetics.com/resources/purple.shtml?ms05.ngv.1q.nr.shtml`, Mai 2005.

[Kul05] Johan Kultti. Secure text in sip based voip. Master's thesis, Lulea University of Technology, Mai 2005.

[KWF05] Richard Kuhn, Thomas Walsh, and Steffen Fries. *Security Considerations for Voice Over IP Systems.* National Institute of Standards and Technology, Januar 2005.

[LHS05] S. Lawrence, A. Hawrylyshen, and R. Sparks. *Problems with Max-Forwards Processing (and Potential Solutions).* SIPPING WG, Oktober 2005.

[li1] Lawful interception overview. `http://www.newport-networks.com/cust-docs/87-Lawful-Intercept.pdf`.

[li2] Lawful interception of ip traffic: The european context. `http://www.blackhat.com/presentations/bh-usa-03/bh-us-03-baloo.pdf`.

[Lon02] Tom Long. Eavesdropping an IP Telephony Call. Technical report, SANS Institute, 2002.

[LRW00] Helger Lipmaa, Phillip Rogaway, and David Wagner. CTR-Mode Encryption. Technical report, NIST, 2000.

[MCA99] M. McBeth, R. Cole, and R. Adamson. Architecture for Secure Network Voice. In *IEEE MILCOM 99*, pages 1454–1457, 1999.

[MEL01] Andrew Moore, Robert Ellison, and Richard Linger. Attack Modeling for Information Security and Survivability. Technical report, Carnegie Mellon University, 2001.

[MHRSW05] M. Mintz-Habib, A. Rawat, H. Schulzrinne, and X. Wu. A VoIP Emergency Services Architecture and Prototype. In *14. International Conference on Computer Communications and Networks*, 2005.

[MK06] Wojciech Mazurczyk and Zbigniew Kotulski. New security and control protocol for VoIP based on steganography and digital watermarking. In *Informatyka - Badania i Zastosowania (IBIZA 2006)*, Februar 2006.

[Mob00] Frederik Moberg. Security analysis of an information system using an attack tree-based methodology. Master's thesis, Chalmers University of Technology, 2000.

[MR04] Nagendra Modadugu and Eric Rescorla. The design and implementation of datagram tls. In *Proceedings of ISOC NDSS 2004*, Februar 2004.

[MSR+03a] A. Milanovic, S. Srbljic, I. Raznjevic, D. Sladden, D. Skrobo, and I. Matosevic. Distributed System for Lawful Interception in VoIP Networks. In *IEEE EUROCON 2003*, volume 1, pages 203–207, 2003.

[MSR+03b] A. Milanovic, S. Srbljic, I. Raznjevic, D. Sladden, D. Skrobo, and I. Matosevic. Methods for Lawful Interception in IP Telephony based on H.323. In *IEEE EUROCON 2003*, volume 1, pages 198–202, 2003.

[NIS95] NIST. *Secure Hash Standard*, April 1995.

[NQBS] S. Niccolini, J. Quittek, M. Brunner, and M. Stiemerling. *VoIP Security Threat Analysis*. NEC.

[Orr05] Joachim Orrblad. Alternatives to mikey/srtp to secure voip. Master's thesis, KTH Royal Institute of Technology, Stockholm, März 2005.

[Osw05] Matthias Oswald. Voip security. ISACA After Hour Seminar, Mai 2005.

[Pos80] Jon Postel. *RFC 768: User Datagram Protocol*. Defense Advanced Research Projects Agency, August 1980.

[Pos81] Jon Postel. *RFC 792: Internet Control Message Protocol*. Defense Advanced Research Projects Agency, September 1981.

[PS] Joachim Posegga and Jan Seedorf. Projektseminar: Aktuelle probleme der it-sicherheit - ws 2004/05: Voice over ip-security. `http://www.informatik.uni-hamburg.de/SVS/teaching/ws2004-05/projseminar/VoIP/index.html`.

[PS05] J. Posegga and J. Seedorf. Voice over IP: Unsafe at any Bandwidth? In *Eurescom Summit 2005 Ubiquitous Services and Applications*, pages 305–314, 2005.

[Ram99] B. Ramsdell. *RFC 2633: S/MIME Version 3 Message Specification*. IETF Network Working Group, Juni 1999.

[RJP05] J. Rosenberg, C. Jennings, and J. Peterson. *Draft: The Session Initiation Protocol (SIP) and Spam*, Juli 2005.

[RM04] E. Rescorla and N. Modadugu. *Datagram Transport Layer Security*. IETF Network Working Group, Juni 2004.

[Rob04] Fernando Robles. The VoIP-Dilemma. Technical report, SANS Institute, 2004.

[Ros01] Rosenberg. *Request Header Integrity in SIP and HTTP Digest using Predictive Nonces*. IETF Network Working Group, Juni 2001.

[rou01] Routing protocol & tunneling attacks. `http://www.blackhat.com/presentations/bh-europe-01/fx/bh-europe-01-fx.pdf`, November 2001.

[RR05] James F. Ransome and John Rittinghouse. *Voice Over Internet Protocol (Voip) Security*. Elsevier, 2005.

[RS02] Rosenberg and Schulzrinne. *RFC 3264: An Offer/Answer Model with the Session Description Protocol (SDP)*. IETF Network Working Group, Juni 2002.

[SC03] Schulzrinne and Casner. *RFC 3551: RTP Profile for Audio and Video Conferences with Minimal Control*. IETF Network Working Group, Juli 2003.

[SCFJ96] Schulzrinne, Casner, Frederick, and Jacobson. *RFC 1889: RTP: A Transport Protocol for Real-Time Applications*. IETF Network Working Group, Januar 1996.

[SCFJ03] Schulzrinne, Casner, Frederick, and Jacobson. *RFC 3550: RTP: A Transport Protocol for Real-Time Applications*. IETF Network Working Group, Juli 2003.

[Sch99] Bruce Schneier. Attack Trees. *Dr. Dobb's Journal*, pages 21–29, Dezember 1999.

[Sch05] Hendrik Scholz. Voip phreaking - introduction to sip-hacking. `http://www.wormulon.net/files/pub/22C3_VoIP_Phreaking.pdf`, 12 2005.

[See04] Jan Seedorf. Security challenges in voip session establishment. `http://www.informatik.uni-hamburg.de/SVS/teaching/ws2004-05/oberseminar/Seedorf_SecurityChallengesinVoIPSessionEstablishment_13Dez2004.pdf`, Dezember 2004.

[She03] Sachin Shenoy. *Draft: Session Initiation Protocol Extension for Response Integrity Check using Validation Cookie*, Juni 2003.

[SKS04] Andread Steffen, Daniel Kaufmann, and Andreas Stricker. VoIP Security. In *E-Science und Grid, Ad-hoc-Netze, Medienintegration - 18. DFN-Arbeitstagung über Kommunikationsnetze*, pages 397–410, 2004.

[SL04] Douglas Sicker and Tom Lookabaugh. VoIP Security: Not an Afterthought. Technical report, ACM, September 2004.

[SM05] Mark Spencer and Franlin W. Miller. *InterAsterisk Exchange, Version 2*. IETF Network Working Group, Juli 2005.

[SP00] Schulzrinne and Petrack. *RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals*. IETF Network Working Group, May 2000.

[SR02] Schulzrinne and Rosenberg. *RFC 3262: Reliability of Provisional Responses in the Session Initiation Protocol (SIP)*. IETF Network Working Group, Juni 2002.

[SRHG02] Schulzrinne, Rosenberg, Huitema, and Gurle. *RFC 3428: Session Initiation Protocol (SIP) Extension for Instant Messaging*. IETF Network Working Group, Dezember 2002.

[SSN+02] Martin Steinebach, Frank Siebenhaar, Christian Neubauer, Jana Dittmann, Utz Roedig, and Ralf Ackermann. Intrusion Detection Systems for IP Telephony Networks. In *RTO IST Symposium on Real Time Intrusion Detection*, 2002.

[Tha01] Johann Thalhammer. Security in voip-telephony systems. Master's thesis, Graz University of Technology, 2001.

[Tuc04] Greg Tucker. VoIP and Security. Technical report, SANS Institute, 2004.

[US02] James Undery and Sanjoy Sen. *SIP Digest Authentication: Extensions to HTTP Digest Authentication*. IETF Network Working Group, Januar 2002.

[Vat05] Jon-Olov Vatn. *IP telephony: mobility and security*. PhD thesis, KTH Royal Institute of Technology, Stockholm, Mai 2005.

[VoI05] VoIP Security Alliance. *VoIP Security and Privacy Threat Taxonomy*, Oktober 2005.

[vom] voice over misconfigured internet telephones. `http://vomit.xtdnet.nl/`.

[WCJ05] Xinyuan Wang, Shiping Chen, and Sushil Jajodia. Tracking Anonymous Peer-to-Peer VoIP Calls on the Internet. In *Proceedings of the 12th ACM conference on Computer and communications security*, pages 81–91, 2005.

[Wei01] Eric Weiss. Security Concerns with VOIP. Technical report, SANS Institute, 2001.

[YH04] Song Yuan and Sorin A. Huss. Audio Watermarking Algorithm for Real-time Speech Integrity and Authentication. In *Proceedings of the 2004 workshop on Multimedia and Security*, pages 220–226, 2004.