



Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Verteilte und Ubiquitäre Systeme

Seminar WiSe 2009/2010

Security Issues and Countermeasures in MANETs and P2P Networks

Dominik Schürmann

betreut durch
Dr. Stephan Sigg

21. Dezember 2009

Zusammenfassung

Peer-to-Peer (P2P) Systeme und Mobile Ad Hoc Networks (MANETs) werden durch die aktuellen Entwicklungen im Bereich Smartphones, verteilten Sensornetzen und Filesharing eine immer größere Bedeutung zugeschrieben. Die Systeme stellen durch ihre Vielfalt unterschiedliche Ansprüche an die Sicherheit. Moderne Filesharingdienste zum Beispiel stellen sie an die Anonymität und dezentrale Verteilung der Dateien unter den Nutzern, Sensornetze an die Authentizität der einzelnen Knoten ohne die Aufsicht einer zentrale Autorität. Um diesen Anforderungen gerecht zu werden basieren heutige Sicherheitskonzepte auf standardisierten Verschlüsselungsprotokollen und Techniken, die eine Kompromittierung des Netzes und ein Abfangen von sensiblen Informationen stark erschweren bis unmöglich machen.

In dieser Ausarbeitung wird auf gebräuchliche Sicherheitskonzepte eingegangen, grundlegende Schwachstellen in dezentralen Netzen aufgezeigt und Konzepte behandelt wie mit diesen umzugehen ist.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufbau und empfohlene Literatur	1
2	Grundlagen der Sicherheit in MANETs und P2P Netzen	2
2.1	Zugrundeliegende Konzepte	2
2.2	Verfügbarkeit	3
2.2.1	Denial-of-Service Attacken	3
2.2.2	Sybil Attacke	7
2.3	Integrität und Authentizität	7
2.4	Anonymität	8
2.4.1	Probleme im Tor Netzwerk	8
2.5	Resultierende Sicherheitsanforderungen	10
3	Kryptografische Grundlagen	10
3.1	Public-Key Kryptosysteme	11
3.2	Erweiterung durch eine Certification Authority als TTP	12
3.3	Threshold Signatur Schema	13
3.4	Byzantine Agreement	14
4	Ergebnis	17

1 Einleitung

Im Zuge neuer Smartphone- und Sensornetzgenerationen steht weit mehr Rechenleistung, RAM und Speicherkapazität auf jedem einzelnen Gerät zur Verfügung als jemals zuvor. Moderne Peer-to-Peer (P2P) Systeme und Mobile Ad Hoc Networks (MANETs) verwalten sich aufgrund dieser Hardwarevoraussetzungen meist dynamisch selbst und sind dezentral angelegt. So entsteht ein Knotenverbund, der ohne eine zentrale Zertifizierungsstelle und ohne Aufsicht auskommen muss. Da ein solcher Knotenverbund meist nicht fest definiert ist, also sich in einer stetigen Anpassungsphase durch das Beitreten und Entfernen von Knoten befindet, kann ein solches System leicht durch Einschleusen eines bösartigen Knotens unterlaufen werden. Dieser Knoten könnte zum Beispiel im Falle eines Sensornetzes sensible Informationen nach außen übertragen oder Implementierungsfehler ausnutzen, um das Netzwerk gezielt zu überlasten. Bei Filesharingdiensten entstehen Probleme, wenn gefälschte Dateien in den Umlauf gebracht werden oder die Anonymität der Benutzer gefährdet wird.

Um diese Schwachstellen zu beheben stehen Verfahren zur Verschlüsselung, Authentisierung und Sicherung der Integrität zur Verfügung, sowie grundlegende Algorithmen um das Kompromittieren solcher Netze erheblich zu erschweren.

1.1 Aufbau und empfohlene Literatur

In der Seminararbeit wird auf gebräuchliche Sicherheitskonzepte und deren Terminologie in MANETs und P2P Netzwerken eingegangen. Hierbei werden ausgewählte sicherheitskritische Problematiken beschrieben und mögliche Angriffsszenarien sowie deren Vermeidung aufgezeigt. Dies wird zu Beginn dargestellt um danach die daraus resultierenden Anforderungen an die Sicherheit in modernen dezentralen Netzwerken zu formalisieren. Um diesen Anforderungen gerecht zu werden ist ein Verständnis der kryptografischen Methoden von Nöten. Unter anderem wird die Umsetzung mithilfe Public-Key Kryptografie, Threshold Signatur Schema und Byzantine Agreement erklärt. Die vorliegende Ausarbeitung kann aber wegen ihres Umfangs nur einen oberflächlichen Einblick in die Thematik bilden, weshalb folgende weiterführende Literatur als sinnvolle erachtet wird:

- Grundlage dieser Seminararbeit bildet Kapitel 17 „Cooperative Security in Peer-to-Peer and Mobile Ad Hoc Networks“ von Esther Palomar, Juan M.E. Tapiador, Julio C. Hernández-Castro und Arturo Ribagorda aus dem Buch „Cooperative Wireless Communications“ [1, Kapitel 17].
- Einen guten Einstieg in die unterschiedlichen Denial-of-Service Angriffsszenarien bietet „Denial of Service in Sensor Networks“ [2] von Anthony D. Wood und John A. Stankovic, das 2002 beim IEEE Issue „Internet Computing“ erschienen ist.
- Einen detaillierten Einblick in die Funktionsweise der Threshold Kryptografie in Bezug auf moderne MANETs bietet der relativ neue Artikel „Threshold cryptography in mobile ad hoc networks under minimal topology and setup assumptions“ [3] von Giovanni Di Crescenzo, Renwei Ge und Gonzalo R. Arce.

- Dietmar Wätjens Buch „Kryptographie. Grundlagen, Algorithmen, Protokolle“ [4] bietet einen Überblick über die mathematischen Hintergründe der Kryptografie und dient an der TU Braunschweig als unterstützendes Lehrbuch in den Vorlesungen Kryptologie I-III.

2 Grundlagen der Sicherheit in MANETs und P2P Netzen

Es bedarf einer kurzen Einführung in MANETs und P2P Netze um danach auf die Sicherheitsprobleme eingehen zu können. Die Schwachstellen und deren Lösungen werden anhand verschiedener Angriffszenarien demonstriert und mögliche Abwehrmethoden erläutert. Hierbei werden verschiedene Artikel als Referenz aufgeführt um unterschiedliche Beispiele in aktuellen Implementierungen aufzeigen zu können.

2.1 Zugrundeliegende Konzepte

Um auf die sicherheitsrelevanten Aspekte eingehen zu können muss die zugrundeliegende Terminologie von P2P Netzen und MANETs erklärt werden.

Bei P2P Netzen unterscheidet man im Normalfall zwischen den Knoten, der dem P2P Netz zugrundeliegenden Architektur und den auszutauschenden Informationen [1, Kapitel 17].

Die Knoten stellen in diesem Zusammenhang die Benutzer des Netzes dar, die in einem Verbund in der Ausgangssituation gleichberechtigt nebeneinander existieren.

Dieser Verbund kann auf unterschiedliche Arten realisiert sein.

Eine *dezentrale Architektur* bildet ein Netz aus Knoten, die Server und Client gleichzeitig sind und sich ohne eine zentrale Koordination steuern. Eine *partiell zentralisierte Architektur* hingegen ernennt besonders leistungsfähige Knoten zu Supernodes, die das Routing sowie das Balancing des Netzes übernehmen. *Hybrid dezentralisierte Architekturen* setzen zusätzlich zu den Netzwerkknoten einen zentralen Server ein, der die Knoten, nicht aber deren Daten, verwaltet.

Die übertragenden Informationen sind auf den Knoten verteilt und werden auf Anfrage externer Systeme repliziert [1, Seite 393-394]. Diese Art von Selbstverwaltung in P2P Netzen führt zu Problemen hinsichtlich des Routings dieser Knoten, deren Kommunikation untereinander und der Sicherheit innerhalb eines Knotenverbundes. Sicherheitssysteme in P2P Netzen sind meist ineffizient und benötigen die Kooperation der Knoten untereinander wenn eine zentrale Autorität fehlt. Ein sogenanntes Trust Management muss peripher aufgebaut werden und kann somit keine vollständig sichere Authentisierung gewährleisten wenn ein gewisser Prozentsatz des Netzes kompromittiert wurde. Die verfügbaren Verfahren sind meist im Bezug auf Verfügbarkeit, Integrität und Anonymität der Knoten optimiert. Diese drei grundlegenden Systemvoraussetzungen sind oft Ziel unterschiedlicher Angriffe, die im Folgenden dargestellt sind.

2.2 Verfügbarkeit

Als ausschlaggebender Faktor für die Benutzbarkeit von außen spielt die Verfügbarkeit eines Verteilten Systems eine große Rolle. Ist ein System auch nur kurzzeitig nicht erreichbar bedeutet dies bei Handynetzen oder Online-Plattformen wie Twitter wirtschaftlichen Schaden [5]. Bei dezentralen Sensornetzen im Gesundheitssektor mit direktem Einfluss auf die Realwelt sogar menschlichen Schaden. Es muss sichergestellt sein, dass das Netzwerk zwischen den Knoten nicht durch Denial-of-Service (DoS) oder Sybil Attacken, die auf Fälschung von Identitäten basieren, gestört werden kann. Um sinnvolle Konzepte entwickeln zu können und die Verfügbarkeit zu gewährleisten müssen mögliche Angriffe evaluiert und untersucht werden.

2.2.1 Denial-of-Service Attacken

DoS Attacken zielen darauf ab ein Netzwerk durch gezielt hervorgerufene Überlastung nicht nutzbar zu machen. Es werden zwar im Normalfall keine sensiblen Informationen preisgegeben, aber dafür das Netzwerk für einen bestimmten Zeitraum teilweise bis vollständig außer Kraft gesetzt.

DoS Attacken lassen sich nur durch das Einbeziehen möglicher Angriffsszenarien schon während des Designprozesses eines solchen Netzwerkes vermeiden. Somit macht es Sinn vor der Implementierung die Szenarien zu betrachten und daraus Konsequenzen zu ziehen.

Sensornetzwerke bestehen oft aus kleinen Knoten, die meist wenig Energiereserven haben, drahtlos kommunizieren, nicht unbedingt eindeutige Identifikationsnummern besitzen und dynamisch Ad-hoc Netze mit anderen Knoten aufbauen müssen. Gerade diese Eigenschaften machen solche Knoten zum einfachen Ziel für DoS Attacken.

Hardwareausfälle, Softwarefehler und Überlastung von einzelnen Ressourcen können einen DoS auslösen. So sollten Sensornetze so entworfen werden, dass trotz eines Ausfalls einzelner Knoten, bei Fehlern in der Übertragung oder bei Überlastung einzelner Netzbereiche, die funktionierenden Teilbereiche weiterhin ausgenutzt werden und dadurch eine gewisse Ausfalltoleranz gewährleistet wird. Ein DoS Angriff kann auf den unterschiedlichen Ebenen der Netzwerkarchitektur erfolgen und ist somit nicht ausschließlich auf einer Schicht zu vermeiden. Im Folgenden wird auf die unterschiedlichen Angriffsszenarien eingegangen und eventuelle Lösungsansätze aufgezeigt. Die vorliegenden Informationen basieren auf „Denial of Service in Sensor Networks“ [2] und der darin referenzierten Literatur.

Physikalische Ebene Auf der physikalischen Ebene müssen die Übertragungswege in der Realwelt betrachtet werden und deren Anfälligkeit bezüglich gezielter Überlastung.

Jamming Unter Jamming versteht man eine Attacke auf die Drahtlosübertragung an sich. Ein Attackierender kann mit einer kleinen Anzahl an eingeschleusten Knoten die eigentlich im Netz befindlichen Knoten außer Gefecht setzen, indem Netzwerktraffic verursacht wird, der die zur Übertragung genutzte Frequenz überlastet und somit keinen Verkehr der eigentlichen Knoten durchlässt. Ein solches Jamming

wird meist über eine längere Zeit betrieben, kann aber auch kurzzeitig genutzt werden um ein Netzwerk zu stören.

Eine denkbare Lösungsstrategie ist Frequency Hopping, also das Wechseln des Frequenzkanals, wenn eine Überlastung auf dem aktuellen Kanal eintritt. Eine vorher festgelegte Sprungreihenfolge sollte nicht von außen ersichtlich sein um das „Hinterherspringen“ des Angreifers auszuschließen. Frequency Hopping wird zum Beispiel im Global System for Mobile Communications (GSM) Standard umgesetzt, ist aber mit einem erheblichen Mehraufwand in der Implementierung verbunden. Gerade Sensornetzknoten verwenden aufgrund ihrer eingeschränkten Hardwarefähigkeiten meist nur einen Frequenzkanal und müssen damit eine andere Lösungsstrategie einsetzen. Um das Aufrechterhalten des Netzes zu gewährleisten sollten betroffene Sensorknoten bei einer Attacke kurze, dafür aber starke, Signale zu Supernode oder einer Basisstation senden und sich danach in eine Art Ruhemodus versetzen um Energie zu sparen. Die koordinierende Basisstation kann dann alternative Routingwege um den betroffenen Bereich herum finden und diese neue Routingtabelle an die anderen Knoten senden, die nun fortan weiterhin erreichbar und funktionsfähig bleiben, wie in Abbildung 1 zu sehen ist. Die betroffenen Knoten können nun in kurzen „Wachphasen“ überprüfen ob ihre Frequenz immer noch überlastet ist und sich gegebenenfalls wieder in die Routingtabelle eintragen lassen.

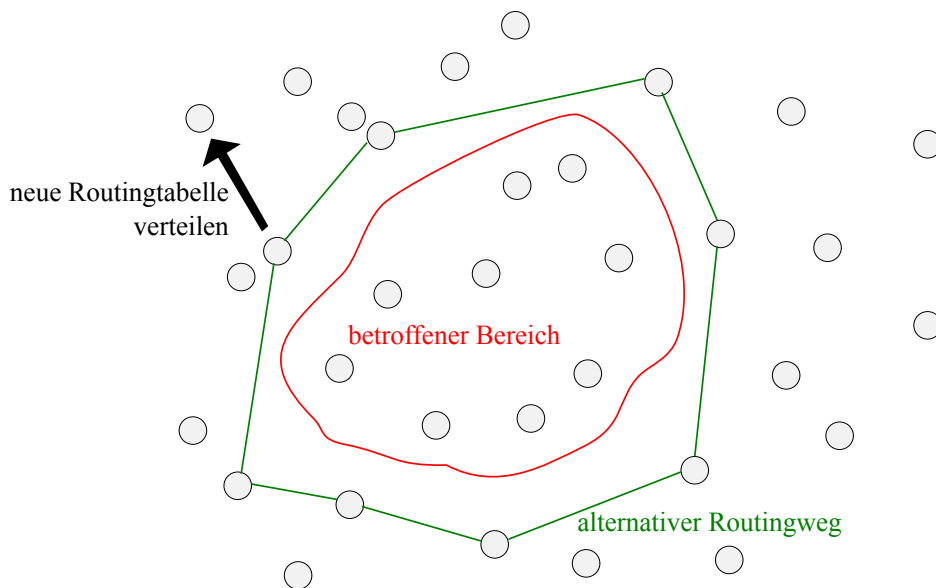


Abbildung 1: Bildung alternativer Routingwege

Tampering Tampering: von *to tamper*, an etwas herumfuschen, bezeichnet eine Attacke, bei dem der Knotenverbund durch das Austauschen von Knoten durch kompromittierte Knoten unterlaufen wird und dieser kompromittierte Knoten dann das Netz durch fehlerauslösende Übertragungen verlangsamt. Dieser Angriff ist nur möglich wenn ein Sensorknoten leicht nachzubauen ist und an die Schlüssel der

Kryptografie gelangt werden konnte. Somit kann einem solchen Angriff im Vorhinein durch eine sichere Hardwareimplementierung und guter Informationspolitik vorgebeugt werden.

Verbindungsebene Die Verbindungsebene stellt die Media Access Control (MAC) Ebene dar, also die Kommunikation der Knoten untereinander. Attacken auf dieser Ebene basieren auf der gezielten Herbeiführung von Ausfällen durch sich wiederholende Nachfragen eines eingeschleusten Knotens.

Kollision Eine herbeigeführte Kollision von Bytes in der Übertragung eines Datenpaketes führt dazu, dass ein gesamtes Paket unbrauchbar wird, da die Prüfsumme nun nicht mehr passend ist. Fehlerkorrigierende Codes können hier Abhilfe bei statistisch herausfindbaren Fehlern schaffen, aber das Problem der Datenkorruption nicht vollständig lösen. Eine aktive Erkennung der Knoten, die fehlerhafte Daten senden, macht in diesem Fall mehr Sinn, um sie vom Knotenverbund auszuschließen und weitere Kollisionen zu vermeiden.

Exhaustion Exhaustion: Englisch für Erschöpfung, steht für einen Angriff auf Verbindungsebene der darauf abzielt mit bestimmten Anfragen die Energiereserven der einzelnen Knoten schneller als im Normalbetrieb aufzubrauchen und sie damit zu erschöpfen. Dies kann zum Beispiel durch das Senden von Paketen mit Kollisionen geschehen um sich wiederholende erneute Übertragungen auszulösen, die den einzelnen Knoten voll auslasten. Ob solch eine Attacke gelingen kann liegt im Allgemeinen am Design und der Implementierung des Übertragungsprotokolls. So ist es in IEEE 802.11 basierenden Protokollen unter anderem möglich mit Request To Send mehrmals eine Anfrage zu starten, die eine Antwort des Knotens erwartet und aufgrund dessen jedes Mal die Hardware auslastet. Um dem entgegen zu wirken bietet es sich an eine maximale Anzahl an Anfragen pro Knoten einzuführen.

Netzwerk- und Routingebene Probleme auf der Netzwerk- und Routingebene entstehen dadurch, dass die Informationen, die ein Knoten aussendet, meist über mehrere andere geleitet werden bevor sie ihr eigentliches Ziel erreichen und somit auf dem Weg anfällig für Störungen sind. Jeder einzelne Knoten agiert hier als eigener Router, der die volle Funktionalität eines solchen umsetzt, das heißt nicht nur eingehenden Datenverkehr sondern auch ausgehenden Datenverkehr kontrolliert. Sobald ein Knoten in diesem Verbund durch einen kompromittierten ersetzt werden kann, lassen sich folgende Angriffe durchführen.

Routing unterlassen/überlasten Eine simple Form um ein Netz zu stören ist den, in einen eingeschleusten Knoten eingehenden, Datenverkehr komplett nicht oder nur teilweise zu seinem Ziel weiterzuleiten. In vielen dezentralen Netzwerken vertrauen die Knoten den anderen Daten an, die zu einem Zielknoten weitergeleitet werden sollen. Wenn dies nicht oder nur sporadisch geschieht, muss der Zielknoten die Daten nach einem Timeout neu anfordern und das Routing der Daten gerät somit

ins Stocken. Zusätzlich kann der eingeschleuste Knoten seinen Nachrichten eine hohe Priorität zuordnen und die anderen Knoten damit überlasten.

Basisstation/Supernodes ausschalten In größeren geografisch abhängigen Netzen nehmen bestimmte, meist besonders leistungsfähige, Knoten, die sogenannten Supernodes, bestimmte organisatorische Rollen ein. Dazu gehört die Aufgabe Routingtabellen zu erstellen und zu verteilen sowie im ungünstigsten Fall das Schlüsselmanagement. Wenn ein solcher Supernode leicht zu identifizieren ist, stellt er eine besondere Angriffsstelle im Netzwerk dar, da nach seinem Ausschalten entweder das gesamte Netz zusammenbricht oder mit nicht unerheblichen Aufwand ein neuer Supernode gefunden und eingestellt werden muss. Bei einem Schlüsselmanagement auf diesem Knoten können eventuell neue Schlüssel durch das unbemerkte Ersetzen im Netzwerk verteilt werden. Das Schlüsselmanagement sollte also über alle Knoten verteilt stattfinden, damit sicherheitsrelevante Funktionen nicht zentralisiert angeboten werden.

Falsches Routing Wenn eine falsche Routingtabelle erstellt wird und unter den anderen Knoten verteilt wurde kann somit der gesamte Datenfluss auf einen Knoten gelenkt werden um diesen zu überlasten. So kann mit Hilfe der eigentlich zum Netzwerk gehörenden Knoten ein wichtiger Supernode angegriffen werden ohne als Angreifer viele Knoten kompromittieren zu müssen. Eine einfachere Form des falschen Routings geschieht durch das Weiterleiten an falsche Zieladressen, beispielsweise durch die Änderung des Headers dieser Pakete [6].

Schwarze Löcher Wenn simple Protokolle auf dem Distanzvektoralgorithmus basieren, kann ein Knoten zu einem Schwarzen Loch werden wenn er alle von ihm zu den anderen Knoten gehenden Verbindungen $k_{1,\dots,n}$ mit den Kosten $c(k_i) = 0$ belegt.

Lösungen, den beschriebenen Angriffen auf der Routingebene zu begegnen, sind unter anderem durch Public-Key Kryptosysteme eine Authorisierung einzuführen, die nur bestimmten zum Netzwerk gehörenden Knoten die Erlaubnis zum Routen des Netzwerktraffics erteilt. Auch wenn solch ein Kryptosystem erfolgreich eingeführt wurde ist es weiterhin möglich einzelne Knoten intakt zu lassen und nur deren Funktionsweise zu ändern um dadurch falsches Routing durchzuführen. Die Lösung dazu liegt in Threshold Kryptografie. Die genaue Funktionsweise der genannten Kryptosysteme folgt in späteren Abschnitten der Ausarbeitung. Ein weiterer Lösungsansatz ist die Beobachtung der Nachbarknoten um deren Funktionsweise zu überprüfen und das darauf folgende eventuelle Ausschließen aus dem Knotenverbund. Eine zusätzlich redundante Übertragung von Nachrichten über unterschiedliche Kanäle, kann die Wahrscheinlichkeit erhöhen, dass der Zielknoten erreicht wird.

Transportebene Auf dieser Ebene findet die Ende-zu-Ende Kommunikation statt, die in Sensornetzen meist durch einfache Protokolle ohne Acknowledgments realisiert ist.

Flooding Übertragungsprotokolle, die Verbindungsstati speichern, sind durch Flooding Attacken angreifbar. So werden Verbindungsanfragen wiederholt an den zu überlastenden Knoten gesendet, der die Stati dieser Verbindungen speichert und daraufhin mit einem vollen Speicher blockiert. Ein Einschränken der Anzahl an Verbindungen würde zu dem Problem führen, dass andere Knoten dem Netzwerk nicht mehr beitreten könnten. Als Lösung bietet sich hier ein Verfahren an mit dem der Knoten ein Puzzle anbietet, also ein schwieriges Problem, das durch eine kurz andauernde Berechnung gelöst werden kann. Alle die sich nun mit diesem Knoten verbinden wollen, müssen erst dieses Problem lösen, dem Knoten die Antwort schicken und erhalten erst dann die Erlaubnis zur Verbindung. Während andere Knoten das Puzzle lösen, müssen keine Daten gespeichert werden und ein Flooding durch viele Anfragen wird durch die Berechnungsdauer des Puzzles unterbunden. Eine viel referenzierte typische Flooding Attacke ist das TCP SYN Flooding, bei der viele TCP SYN Pakete, also Verbindungsanfragen, mit einer gefälschten Quelladresse im Header verschickt werden [7].

Desynchronisation Als Desynchronisation wird das Einschleusen von fremden Paketen in den Datenverkehr zweier Knoten bezeichnet, welches ein immerwährendes erneutes Synchronisieren der beiden Knoten erzwingt. Auf diese Weise können gezielt eingeschleuste Pakete den sinnvollen Datenaustausch der beiden Knoten unterbinden.

2.2.2 Sybil Attacke

In einem verteilten System ohne eine zentrale Autorität kann ein Knoten mehrere Identitäten annehmen. Um Integrität und Datensicherheit zu gewährleisten werden Aufgaben über mehrere Knoten verteilt oder Daten fragmentiert, bei denen dadurch nur alle Datenfragmente gemeinsam das ganze Datenpaket ergeben. Eine Sybil Attacke nutzt diese Schwachstelle des Systems um viele Identitäten zu erzeugen, diese anzunehmen und somit das Netzwerk zu infiltrieren [8]. Wenn ein eingeschleuster oder kompromittierter Knoten viele Identitäten vortäuscht, kann er beispielsweise alle nötigen Datenfragmente erhalten oder den eigenen Knoten zum Supernode ernennen.

Sybil Attacken vollständig zu unterbinden stellt eine Herausforderung für komplett dezentrale Netze dar, da in dem Fall keine Validierungs- und Authentifizierungssysteme mit einer Trusted Third Party (TTP) implementiert werden können. Solche Authentifizierungsverfahren werden von den Knoten selbst übernommen, die mit einem Mindestprozentsatz an wählenden Knoten entscheiden ob eine neue Identität in das Netz aufgenommen oder abgewiesen wird. Andere Sicherheitsmaßnahmen, wie das schon erläuterte bereitstellen eines Puzzles, verhindern das Überfluten eines Netzes durch neue Identitäten.

2.3 Integrität und Authentizität

In „Enhancing Data Authenticity and Integrity in P2P Systems“ [9] werden am Beispiel der Netzwerke Friend Troubleshooting Network, Gnutella und dem BitTorrent Vertei-

lungssystem zeigt welche Forschungsansätze existieren um die Integrität und Authentizität der einzelnen Knoten zu gewährleisten. Bei älteren Filesharingdiensten war es leicht möglich gefälschte Dateien in den Umlauf zu bringen indem sie einfach nach häufig angefragten Dateien benannt und durch Nutzer weiter verteilt wurden, die diese fälschlicherweise herunterluden. Im Fall von BitTorrent werden diese Dateien durch einen Tracker kontrolliert, der die einzelnen Dateifragmente verwaltet und mit einem SHA-1 Hashwert versieht, sodass diese eindeutig identifiziert und nicht ausgetauscht werden können. Das Servletprogramm des Nutzers kann aber bei der Überprüfung der Hashwerte nicht erkennen ob es sich bei einer Nichtübereinstimmung um einen Übertragungsfehler oder ein gefälschtes Datenfragment handelt.

Abhilfe der Integritäts- und Authentizitätsproblematik versprechen Trusted Platform Modules, die als Hardwareelement verbaut werden und Sicherheit in Datenverarbeitung und Übertragung herstellen, da sie eindeutige Identifikationsmerkmale besitzen, primitive Sicherheitsalgorithmen als Hardwareimplementierung beinhalten und bei Public-Key Kryptografie der private Schlüssel in der Hardware gekapselt vorliegt und damit nicht ausgelesen werden kann. Ein Challenge Response Protokoll wird im Zuge der Integritätsicherung genutzt um eine Identifikationsanfrage eines Knotens mit einem eindeutigen Profil zu beantworten, das über die Trusted Platform und den darin gespeicherten Zertifikaten generiert wird. So kann der andere Knoten das Profil auf dessen Authentizität überprüfen und nach Bestehen der Prüfung mit seiner eigenen Signatur unterschreiben.

2.4 Anonymität

In P2P Systemen spielt die Ende-zu-Ende Anonymität eine wichtige Rolle. Viele Menschen veröffentlichen Informationen im Internet und möchten dies eventuell anonym tun, entweder um private Daten zu schützen oder Zensur zu entgehen. Anonymität in P2P Netzen bedeutet die Identitäten der Kommunikationspartner zu verstecken damit keine Relationen zwischen Identitäten im P2P Netzwerk und Realpersonen hergestellt werden können. Das Tor Netzwerk, mit dem es möglich ist TCP Verbindungen zu anonymisieren, ist ein Beispiel des sinnvollen Einsatzes eines Knotennetzwerkes, das ein gutes Verhältnis zwischen Anonymität und Nutzbarkeit gewährleistet.

2.4.1 Probleme im Tor Netzwerk

Das moderne Tor Netzwerk basiert auf einem sogenannten Onion Routing Prinzip. In diesem Knotennetzwerk stellt ein anfragender Knoten eine Verbindung über mehrere andere Zwischenknoten her, bei denen diese Knoten nur ihren Vorgänger und Nachfolger, aber keine anderen Knoten im Netzverbund kennen. So wird der Datenverkehr, wie in Abbildung 2 zu sehen, über mehrere Knoten geleitet bevor er den Zielservers erreicht. Die Probleme hingehend der Anonymität liegen im Detail der Implementierung [10], die beispielhaft aufgezeigt werden.

Replay-Attacken Die ersten Onion Routing Implementierungen waren anfällig für Replay-Attacken, das heißt ein kompromittierter oder anderweitig kontrollierter Knoten

Verbindung via Tor

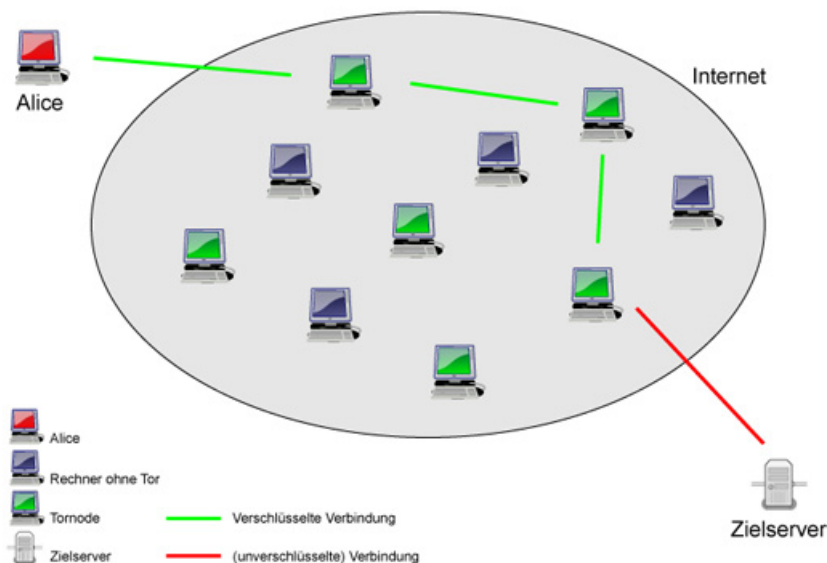


Abbildung 2: Datenfluss im Tor Netzwerk (Abbildung von der Webseite http://onro.de/index.php?seite=was_ist_tor [11])

hat den Datenverkehr aufgenommen, um zu einem späteren Zeitpunkt gezielt die auf der Datenroute liegenden Knoten zu kompromittieren. Diese wurden dazu genutzt den aufgenommenen Datenverkehr zu dechiffrieren. Um dem Problem entgegenzuwirken werden in der heutigen Implementierung des Tor Netzwerkes Session-Keys für jeden Teilnehmer in dem Routingpfad erstellt und später, nach dem Senden der Informationen über diesen Pfad, wieder gelöscht.

Überlastungen Überlastungen an bestimmten Bottlenecks wie zum Beispiel an Exit-Nodes, also Knoten die am Ende des Routingpfades stehen, wurden bei früheren Implementierungen nicht bedacht, da bei einer komplett dezentralen anonymen Architektur kein Gesamtbild über das Netzwerk erstellt werden kann, also auch keine Flusskontrolle stattfindet.

Verteilung der Routing- und Statusinformationen Statusinformationen und Routingtabellen wurden durch das gesamte dezentrale Netzwerk gebroadcastet, was einen deutlichen Overhead erzeugte, da diese Informationen regelmäßig auf allen Knoten aufgefrischt werden mussten. Später wurden Supernodes, also besonders vertrauenswürdige und performante Knoten, eingeführt, die eine Art Directory-Service innehatten, der bekannte Routen und Statusinformationen auf Anfrage freigab. Dadurch wurde das Netzwerk aber auch partiell zentralisiert.

Datenintegrität der gesendeten Daten Die Datenintegrität wurde an Exit-Nodes nicht nochmals überprüft. So konnte jeder Knoten, der Daten weitergeleitet hat, die Header der ankommenden Pakete ändern und verändert weiterschicken, um zum

Beispiel die Anfrage an einen Server bezüglich einer Webseite an einen anderen Server umzuleiten. Dies ist in aktuellen Implementierungen durch Verschlüsselung mit Session-Keys und einer Prüfung der Datenintegrität am Exit-Node ausgeschlossen.

2.5 Resultierende Sicherheitsanforderungen

Aus den evaluierten Angriffsszenarien lassen sich Sicherheitsanforderungen ableiten, die auf MANETs und P2P Netze im Allgemeinen zutreffen. Die zu dem Netzwerk gehörenden Knoten müssen sich authentisieren und unter einer Zugriffskontrolle stehen, um den Verbund gegen nicht autorisierte Zugriffe durch eingeschleuste Knoten zu schützen. Die Anforderungen an Authentifizierung, Vertrauenswürdigkeit und Integrität der Daten stehen oft im Gegensatz zu Performance und niedrigen Hardwarevoraussetzungen in Sensornetzwerken sowie Anonymität bei Filesharingdiensten.

Um eine gesicherte Datenübertragung zu gewährleisten müssen Schlüssel auf den Knoten verteilt werden um die Daten zu chiffrieren. Diese müssen ad hoc generiert und übertragen werden, wobei nicht jedem Knoten im Netzwerk automatisch getraut werden kann. Entweder besteht die Möglichkeit eine zentrale Auhorität als Certification Authority (CA) oder ein Web Of Trust Modell wie bei Pretty Good Privacy (PGP) zu nutzen. Eine weitere Anforderung ist, dass ein einzelner Knoten alleine keine wichtigen Aktionen ausführen können soll, sondern nur im Verbund mit seinen Nachbarn. Um dies zu gewährleisten bietet sich Threshold Kryptografie an, bei der beispielsweise die Aktion des Signierens auf mehrere Knoten verteilt wird. Dazu kommt, dass ein sicheres Routing gewährleistet werden soll, damit kein Knoten, der nur Daten weiterleitet, diese auch verändern kann.

Viele Eigenschaften stehen im Gegensatz zu Fehlertoleranz und Skalierbarkeit des Netzes, und so muss in jeder Implementierung neu abgewogen werden inwiefern diese Sicherheitsanforderungen Priorität im speziellen Einsatzgebiet haben.

Zur Umsetzung werden kryptografische und verfahrensspezifische Grundlagen benötigt, die im nächsten Abschnitt erklärt sind.

3 Kryptografische Grundlagen

In verteilten Systemen sind bekannte symmetrische Kryptografieverfahren ungeeignet, da der Schlüssel zur gesicherten Übertragung vorher ausgetauscht werden muss. In einem dynamischen Knotenverbund, in dem Knoten in unregelmäßigen Abständen hinzukommen und auch wieder entfernt werden, muss so jedes Mal der Schlüssel neu an die Teilnehmer ausgeteilt werden. Dies ist problematisch, da der auszutauschende Schlüssel nicht selbst über einen unsicheren, meist drahtlosen, Übertragungsweg geschickt werden kann. Somit ist dieses Verfahren nicht praktikabel. Was sich anbieten würde, sind Schlüsselvergabeverfahren über eine Trusted Third Party (TTP). Da es aber nicht immer möglich ist einen verwaltenden außenstehenden Server zu integrieren und auch bei einem solchen System ein privater Schlüssel ausgetauscht werden müsste um mit der TTP zu kommunizieren, ist auch dies in einem vollständig dezentralen Netzwerk nicht umsetzbar. Als Lösungsstrategie bietet sich ein asymmetrisches Public-Key Kryptosystem an,

da es auf diese Weise nicht notwendig ist private Schlüssel über das Netz auszutauschen.

3.1 Public-Key Kryptosysteme

Bei einem Public-Key Kryptosystem handelt es sich um ein asymmetrisches Verschlüsselungsverfahren. Das heißt zu jeder Identität existiert ein öffentlicher Schlüssel e_B und ein private Schlüssel d_B , der geheimgehalten und in unserem Fall auf dem Knoten gespeichert wird. Bei der Übertragung einer verschlüsselten Nachricht werden folgende Schritte durchlaufen (siehe dazu Abbildung 3):

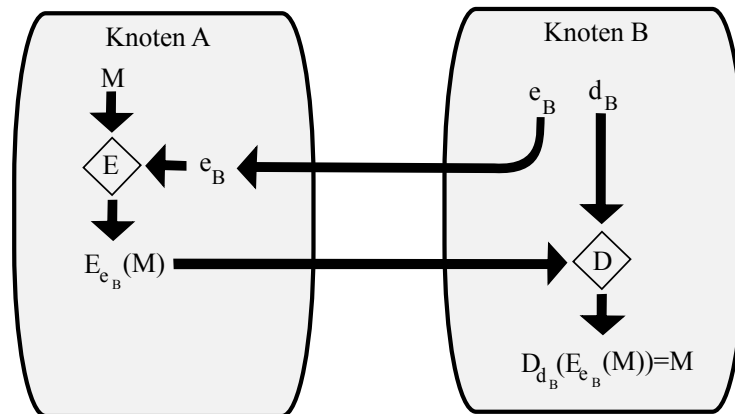


Abbildung 3: Public-Key Verschlüsselung

1. Ein öffentlicher Schlüssel e_B und der dazugehöriger private Schlüssel d_B werden mit einem Public-Key Schlüsselgenerator von Knoten B erstellt und der private sicher auf dem Knoten gespeichert.
2. Der öffentliche Schlüssel wird je nach Implementierung im Netzwerk freigegeben oder über die Knoten verteilt.
3. Eine Nachricht, die von Knoten A an Knoten B gesendet werden soll, wird mit Hilfe des öffentlichen Schlüssels chiffriert: $E_{e_B}(M)$.
4. Die chiffrierte Nachricht $E_{e_B}(M)$ wird an B übertragen.
5. Der empfangende Knoten B dechiffriert die Nachricht mit $D_{d_B}(E_{e_B}(M)) = M$

Um eine Nachricht zu unterschreiben und damit eindeutig einem Knoten zuordnen zu können, wird ein, durch das Public-Key System ermöglichtes, simples Signierverfahren genutzt (siehe dazu Abbildung 4):

1. Es wird mit $h(M)$ ein Hashwert der Nachricht gebildet.
2. Eine digitale Signatur, mit der der Knoten A seine Nachricht unterschreiben kann, wird durch das Chiffrieren dieses Hashwertes erzeugt: $s_A(M) = D_{d_A}(h(M))$

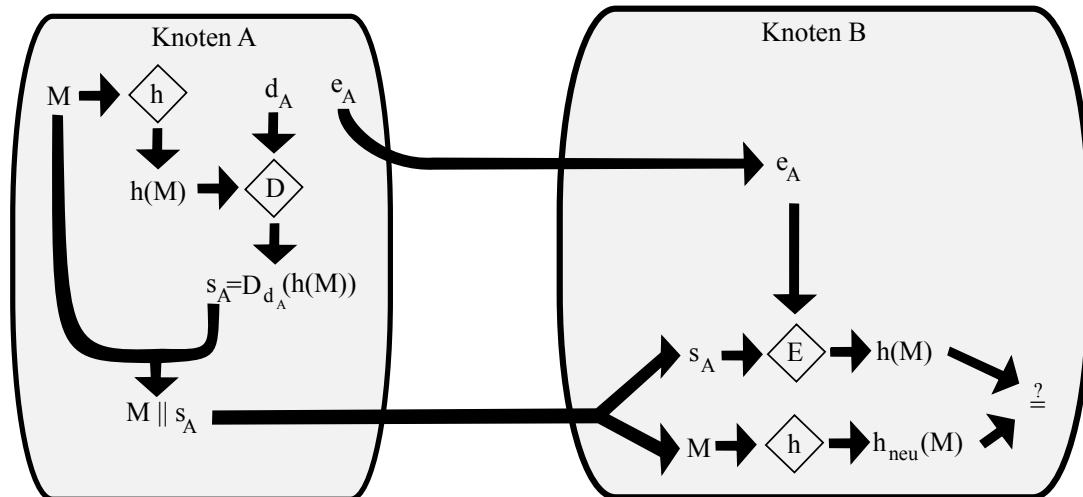


Abbildung 4: Public-Key Signatur

3. Die Signatur $s_A(M)$ wird nun mit der zu übertragenden Nachricht M , unabhängig davon ob diese wiederum verschlüsselt wurde oder nicht, concatenated $M || s_A(M)$ und als Datenpaket übertragen.
4. Der empfangende Knoten B teilt das Datenpaket wieder in Signatur $s_A(M)$ und Nachricht M .
5. Der Hashwert der empfangenen Nachricht $h_{neu}(M)$ wird berechnet.
6. Die empfangene Signatur wird entschlüsselt und mit dem neu berechneten Hashwert verglichen: $E_{e_A}(s_A(M) = E_{e_A}(D_{d_A}(h(M)))) \stackrel{?}{=} h_{neu}(M)$. Sind die Hashwerte gleich, ist gesichert, dass die Nachricht M auf ihrem Übertragungsweg nicht durch einen Dritten geändert wurde.

Probleme entstehen bei einem solchen Public-Key Kryptosystem dadurch, dass zwar die Datenpakete authentifiziert sind, aber nicht sichergestellt ist, dass der Public-Key selber authentifiziert wurde. Dies ist entweder durch eine TTP zu erreichen, der jeder Knoten voll vertraut und die wiederum die Public-Keys bereithält und signiert hat oder durch den Einsatz des Byzantine Agreements.

3.2 Erweiterung durch eine Certification Authority als TTP

Um sicherzustellen, dass es sich bei dem verwendeten Public-Key zur Verschlüsselung der Nachricht wirklich um den Public-Key handelt, der zu dem Knoten B gehört, wird eine TTP, in dem vorliegendem Fall eine Certification Authority (CA), benötigt, die eine Datenbank über alle Knoten und deren öffentlichen Schlüssel führt. Im Folgenden wird vorausgesetzt, dass die Verbindung zwischen dem Knoten und der CA selbst unabhängig

vom folgenden Verfahren gesichert ist. Im Beispiel kann dies wiederum durch eine asymmetrische Verschlüsselung realisiert sein, bei der jedem Knoten der öffentliche Schlüssel der CA fest inkodiert wurde und somit nicht erfragt werden muss.

1. Vor der Übertragung der signierten Nachricht $M||s_A(M)$, werden M , $h(M)$ und $s_A(M)$ über eine gesicherte Verbindung an die CA geschickt.
2. Die CA generiert ihre Signatur zur Nachricht M mit $s_{CA} = D_{d_{CA}}(h(M))$, falls die Signatur $s_A(M)$ validiert werden konnte.
3. Die CA sendet $M||s_{CA}$ über den verschlüsselten Kanal an Knoten A .
4. Knoten A vergleicht die entschlüsselte Signatur der CA mit einem neu generierten Hashwert: $E_{e_{CA}}(s_{CA}(M)) = E_{e_{CA}}(D_{d_{CA}}(h(M))) \stackrel{?}{=} h_{neu}(M)$

Ein solches Public-Key System benötigt viel Rechenleistung sofern jedes Paket auf diese Art und Weise verschlüsselt werden soll, wobei den einzelnen Knoten meist, abhängig vom Typ des Netzwerks, nicht genug Hardwareleistung zur Verfügung steht. So wird die asymmetrische Public-Key Kryptografie oft nur zum einmaligen Austausch eines Schlüssels für ein symmetrisches Verschlüsselungsverfahren genutzt.

3.3 Threshold Signatur Schema

Ein zu lösendes Problem stellt die Anwendung der Public-Key Kryptografie auf ein komplett dezentrales Netzwerk wie ein MANET ohne TTP dar. Solch ein vollständig dezentrales Netzwerk hat die Vorteile einer höheren Ausfallsicherheit, da das ganze System nicht von einem außenstehenden Server abhängt, der ein Angriffsziel für Denial-of-Service Attacken darstellt. Des Weiteren soll aber auch zukünftig sichergestellt sein, dass der Knotenverbund auch funktionsfähig bleibt, sollten einzelne Knoten kompromittiert worden sein. Ein Lösungsansatz bietet die Threshold Kryptologie, bei der mehrere Knoten ohne eine zentrale Autorität gemeinsam zum Beispiel Signaturen validieren oder Routing betreiben. Formal gesehen bietet sich konkret das Threshold Signatur Schema an, bei welchem ein Knoten die Anforderung sendet, dass eine Nachricht M unterschrieben werden soll und ein Teil der Gesamtknoten diese Operation, wie in Abbildung 5 zu sehen, gemeinsam ausführen. Im Artikel „Threshold cryptography in mobile ad hoc networks under minimal topology and setup assumptions“ [3] wird das Verfahren ausführlich erklärt.

1. In einem Netzwerk $G = (V, E)$ sendet ein Knoten $C \in V$ eine Anfrage die Nachricht M zu unterschreiben. Die Anfrage enthält die Nachricht M , die Identität des sendenden Knotens und die Parameter t und n . t sei die Anzahl der am Signierverfahren mindestens teilnehmenden Knoten; n die Anzahl aller vorhandenen Knoten
2. Alle verfügbaren Knoten $P_i \in V; i = 1, \dots, n$ nutzen das Key Generation Protocol Π_{kg} um jeweils einen String sk_i zu generieren.

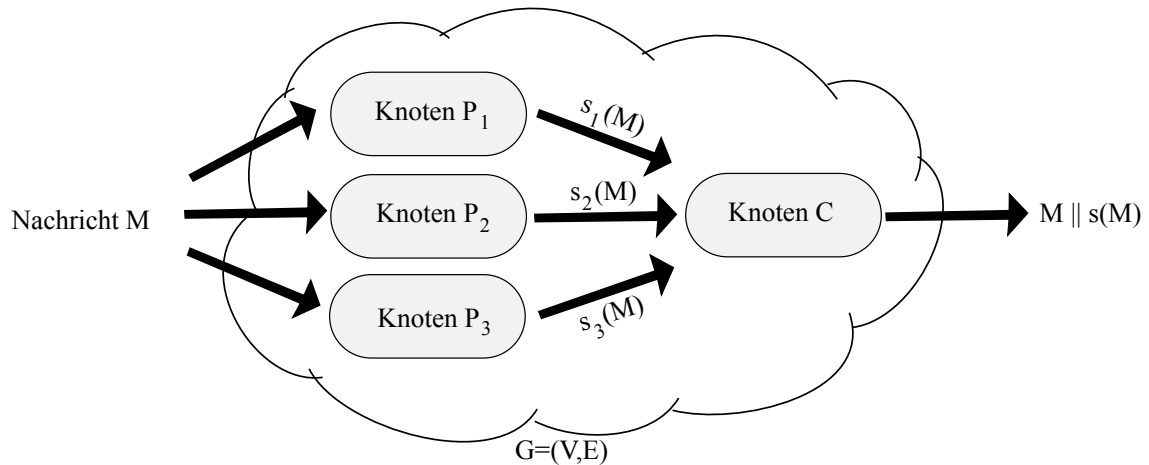


Abbildung 5: Threshold Signatur Schema

3. Die generierten Strings $sk_i; i = 1, \dots, n$ werden mit weiteren Parametern genutzt um Teilsignaturen der einzelnen Knoten zu erzeugen
4. Diese Teilsignaturen werden vom signierenden Knoten C mit dem Signaturprotokoll Π_{sgn} zusammengeführt, welches eine Signatur für die Nachricht M erzeugt.

In einem drahtlosen dezentralen MANET kann nicht davon ausgegangen werden, dass der Verbindungsgraph $G = (V, E)$ vollständig ist. Wenn weiterhin davon ausgegangen wird, dass ein Angreifer in der Lage ist τ Knoten zu kompromittieren oder auszuschalten, müssen in dem Threshold Modell mehr als $2\tau + 1$ Knoten im Netzwerk vorhanden sein: $|V(G)| \geq 2\tau + 1$, damit eine gültige Signatur erzeugt werden kann.

3.4 Byzantine Agreement

Bei dem Byzantine Agreement wird der Umgang mit fehlerhaften oder kompromittierten Knoten thematisiert, welche sich bösartig gegenüber anderen verhalten. Das Byzantine Generals Problem wurde erstmalig 1982 unter anderem von Leslie Lamport definiert:

„We imagine that several divisions of the Byzantine army are camped outside an enemy city, each division commanded by its own general. The generals can communicate with one another only by messenger. After observing the enemy, they must decide upon a common plan of action. However, some of the generals may be traitors, trying to prevent the loyal generals from reaching agreement. The generals must have an algorithm to guarantee that

A. All loyal generals decide upon the same plan of action.

[...]

B. A small number of traitors cannot cause the loyal generals to adopt a bad plan.“ [12, S. 382-383]

Untersuchungen haben gezeigt, dass bei einer mündlichen Übertragung $\frac{2}{3}$ aller Generale loyal sein müssen damit ein Agreement erfolgreich ausgehandelt werden kann, beziehungsweise unter $3m + 1$ Generalen maximal m Verräter sein dürfen. Korrekte Authentifizierung wird in einem dezentralen Netzwerk durch die Gruppe der „vertrauenswürdigen Knoten“ hergestellt (siehe Abbildung 6). Diese Gruppe übernimmt im Verbund die Aufgaben einer TTP und ist dafür zuständig, aktiv alle Knoten im Verbund, auch die in der Gruppe selber, periodisch zu überprüfen und neu zu authentisieren. Der Authentisierungsprozess erfolgt in vier Schritten:

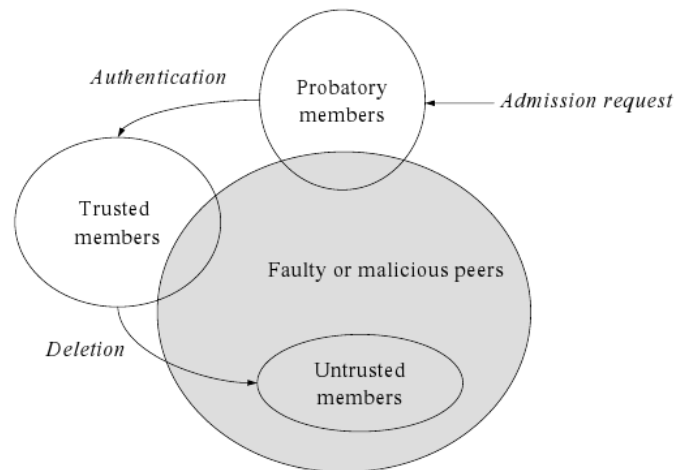


Abbildung 6: Authentisierung und Kategorisierung im Knotenverbund (Abbildung aus dem Artikel „A P2P Content Authentication Protocol Based on Byzantine Agreement“ [13, S. 63, Fig. 1])

1. **Verbindungsanfrage:** Knoten B entdeckt beim periodischen Überprüfen des Verbundes einen neuen Knoten A mit dem dazugehörigen öffentlichen Schlüssel e_A . Knoten B fragt eine Teilgruppe der „vertrauenswürdigen Knoten“ eine gemeinsame Validierung des öffentlichen Schlüssels e_A vorzunehmen (Abbildung 7, Nachricht 1). Alle an der Validierung teilnehmenden Knoten bekommen nun e_A zugesickt.
2. **Challenge Response Verfahren:** Jeder am Authentisierungsprozess teilnehmende Knoten sendet dem Knoten A eine zufällige Nonce, also eine einmalig verwendete Zeichenfolge, die mit dem öffentlichen Schlüssel e_A chiffriert wurde (Abbildung 7, Nachricht 2). Knoten A muss nun jede empfangene chiffrierte Nonce mit seinem privaten Schlüssel d_A entschlüsseln (siehe Abschnitt „Public-Key Kryptosysteme“) und die so entschlüsselte Nonce zu den jeweiligen Knoten zurücksenden (Abbildung 7, Nachricht 3). Jeder Knoten kann nun die empfangene Nonce mit der gesendeten vergleichen und erhält somit die Sicherheit, dass es sich bei dem Schlüssel e_A wirklich um den öffentlichen Schlüssel von Knoten A handelt.

3. **Verteilte Authentisierung:** Jeder am Prozess teilnehmende und Knoten A authentisierende Knoten sendet seine Teilauthentisierung an Knoten B (Abbildung 7, Nachricht 4). Wenn Knoten A von allen anderen authentisiert wurde, gehört der öffentliche Schlüssel e_A eindeutig zu Knoten A . Sind kompromittierte Knoten im Netzwerk oder wurden fehlerhafte Daten gesendet, wird die vierte Phase, das Byzantine Agreement, eingeleitet.
4. **Byzantine Agreement:** Um zu überprüfen ob es sich bei A um einen bösartigen Knoten handelt, schickt B eine Überprüfungsanfrage an A . Knoten A muss nun mit allen im Challenge Response Verfahren empfangenen chiffrierten Noncen und allen mit e_A dechiffrierten Noncen antworten (Abbildung 7, Nachricht 5). Wenn Knoten A vertrauenswürdig ist, wird er diese erwartete Antwort korrekt senden. Dadurch wird ersichtlich, dass mindestens ein anderer Knoten, der die zurückgesendete Nonce nicht an B bestätigt hat, bösartig oder fehlerhaft gewesen sein muss. Mit der 6. Nachricht sendet B ein Byzantine Fault an die teilnehmenden Knoten. Jeder Teilnehmer sendet nun eine Zustimmungsnachricht an die jeweils anderen, wodurch schlussendlich die bösartigen oder fehlerhaften Knoten erkannt werden können.

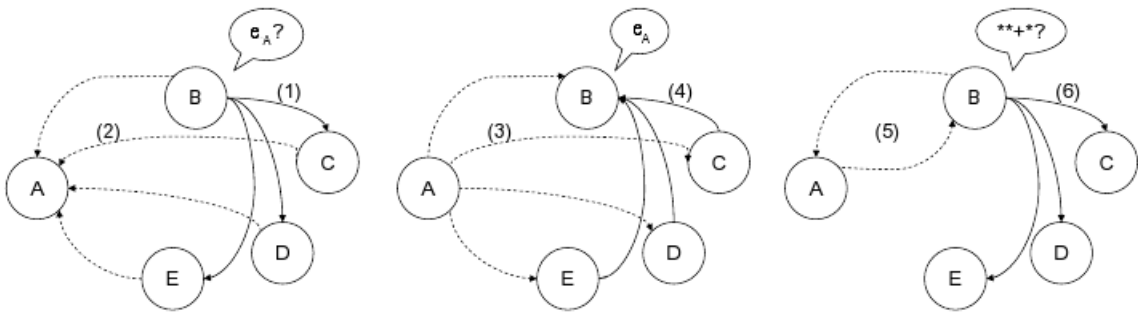


Abbildung 7: Authentisierungsprotokoll (Abbildung aus dem Artikel „A P2P Content Authentication Protocol Based on Byzantine Agreement“ [13, S. 64, Fig. 2])

Im Zuge der Authentisierung werden nach erfolgreichem Durchlauf des Algorithmus als vertrauenswürdig eingestufte Knoten in die entsprechende Kategorie eingeordnet und bösartig oder fehlerhafte entfernt. Das vorliegende Protokoll bietet also eine Authentisierung ohne TTP an und ist daher ein essentieller Bestandteil moderner dezentraler Netzwerke.

4 Ergebnis

Schon in der Design- und Implementierungsphase eines Protokolls und den dazugehörigen Applikationen für ein dezentrales Netzwerk müssen Sicherheitsanforderungen definiert und umgesetzt werden. Dazu gehört die Evaluation bekannter Angriffsszenarien und die Umsetzung erprobter Gegenmaßnahmen. Es muss abgewogen werden, welche Methoden einzusetzen sind um eine Sicherheit zu gewährleisten ohne die Benutzbarkeit der Anwendung einzuschränken. So ist eine absolute Anonymität in einem dezentralen Netzwerk nur unter Gefährdung der Performance und Authentifizierung möglich. Auch sich im Einsatz befindende Software muss regelmäßig auf Schwachstellen untersucht und daraufhin verbessert werden, um auf Dauer einen hohen Grad an Sicherheit gewährleisten zu können, wie dies zum Beispiel beim Tor Netzwerk durch stetige Forschung passiert [14].

Dadurch, dass in modernen Knotenverbunden immer mehr Hardwareleistung zur Verfügung steht, eignen sich auch aufwändigere Algorithmen wie die Threshold Kryptografie und verteilte Routingverfahren immer mehr für den realen Einsatz. In Anwendungsszenarien wie Sensornetzwerken, in denen keine Anonymität gefordert ist, macht es Sinn neben einer DoS sicheren Protokollimplementierung eine Kombination aus Public-Key Kryptografie mit Threshold Signatur zu nutzen, wobei dynamisch ernannte Supernodes Teilaufgaben einer TTP übernehmen. Dabei müssen sie durch Vertrauenswürdigkeit und Performance ausgezeichnet sein. So wird gerade bei stark dynamischen Netzwerken eine gewisse Grundperformance beim Routing durch TTPs sichergestellt ohne die Sicherheitsaspekte zu vernachlässigen. Die vorliegende Ausarbeitung hat einen Einblick in die Angriffsszenarien geboten um jeweils im Anschluss Konzeptionsmöglichkeiten für die Entwicklung sicherer Netzwerke vorzustellen.

Literatur

- [1] Yan Zhang, Hsiao-Hwa Chen, and Mohsen Guizani. *Cooperative Wireless Communications*. Auerbach Publications, 2009.
- [2] Anthony D. Wood, John A. Stankovic, D Anthony, and A John. Denial of service in sensor networks. In *Upper Saddle River*. Prentice –Hall, Inc, 2002.
- [3] Giovanni Di Crescenzo, Renwei Ge, and Gonzalo R. Arce. Threshold cryptography in mobile ad hoc networks under minimal topology and setup assumptions. *Ad Hoc Networks*, 5(1):63–75, 2007.
- [4] Dietmar Wätjen. *Kryptographie. Grundlagen, Algorithmen, Protokolle. (Spektrum Lehrbuch)*. Spektrum Akademischer Verlag, 2 edition, 2007.
- [5] Iranian hacker attack: What will it cost twitter? / the christian science monitor - CSMonitor.com. <http://www.csmonitor.com/Money/2009/1218/Iranian-hacker-attack-What-will-it-cost-Twitter>.
- [6] CERT advisory CA-1998-01 smurf IP Denial-of-Service attacks. <http://www.cert.org/advisories/CA-1998-01.html>.
- [7] Christoph L. Schuba, Ivan V. Krsul, Markus G. Kuhn, Eugene H. Spafford, Aurobindo Sundaram, and Diego Zamboni. Analysis of a denial of service attack on tcp. pages 208–223, 1997.
- [8] John Douceur and Judith S. Donath. The sybil attack. pages 251–260, 2002.
- [9] X. Zhang, S. Chen, and Ravi Sandhu. Enhancing data authenticity and integrity in p2p systems. *Internet Computing, IEEE*, 9(6):42–49, Nov.-Dec. 2005.
- [10] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *In Proceedings of the 13 th Usenix Security Symposium*, 2004.
- [11] Was ist tor. http://onro.de/index.php?seite=was_ist_tor.
- [12] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4:382–401, 1982.
- [13] Esther Palomar, Juan M. Estevez-Tapiador, Julio C. Hernandez Castro, and Arturo Ribagorda. A p2p content authentication protocol based on byzantine agreement. In Günter Müller, editor, *ETRICS*, volume 3995 of *Lecture Notes in Computer Science*, pages 60–72. Springer, 2006.
- [14] Tor: Volunteer. <https://www.torproject.org/volunteer.html.en#Research>.