Ein Sicherheitskonzept für clusterbasierte Ad-hoc-Netze

Marc Bechler, Achim Hauck, Daniel Müller, Frank Pählke, Lars Wolf

Institut für Telematik
Universität Karlsruhe (TH)
Zirkel 2
D-76128 Karlsruhe
[bechler | hauck | dmueller | paehlke | wolf]@tm.uka.de

Abstract: Die Authentisierung ist eine der wesentlichen Grundlagen für die Sicherheit in Kommunikationsnetzen. In Ad-hoc-Netzen stellt die Authentisierung jedoch eines großes Problem dar, da sich eine klassische Public-Key-Infrastruktur mit zentralen Zertifizierungsstellen nur sehr schwer implementieren lässt. In diesem Beitrag stellen wir ein Sicherheitskonzept für clusterbasierte Ad-hoc-Netze vor, das auf der proaktiven Geheimnisteilung basiert. Zur gesicherten Kommunikation wird ein vierstufiges Sicherheitsmodell entworfen, das ohne zentrale Instanz zur Schlüsselverwaltung auskommt. Des Weiteren wird auch gezeigt, welche Protokollerweiterungen für die übrigen Kommunikationsprotokolle notwendig sind, damit diese in das Sicherheitskonzept integriert werden können.

1 Motivation

Ad-hoc-Netze sind Angriffen derzeit meist schutzlos ausgeliefert. Zum einen kann die drahtlose Kommunikation durch passive Angriffe leicht abgehört werden, zum anderen sind die Kommunikationsprotokolle auch durch gezielte Angriffe verwundbar. Angriffe sind dabei nicht auf eine einzelne Kommunikationsschicht beschränkt, sondern können auf allen Schichten durchgeführt werden. Beispielsweise sind Denial-of-Service-Angriffe auf folgenden Schichten möglich: physikalische Schicht (z.B. durch Störung der Übertragungsfrequenzen), Sicherungsschicht (z.B. durch permanente Belegung des Mediums), Netzwerkschicht (z.B. durch gezieltes Einschleusen von falschen Routing-Informationen) und Anwendungsschicht (z.B. durch verteilte Denial-of-Service-Angriffe). Im Gegensatz zu stationären Netzen sind bekannte Angriffe wie Maskerade, Man-in-the-Middle, Replay oder das Einschleusen von Daten in Ad-hoc-Netzen meist einfacher durchzuführen.

Der Einsatz von entsprechenden Sicherheitsmechanismen wird durch die spezifischen Eigenschaften von Ad-hoc-Netzen zusätzlich sehr erschwert. Zu diesen Eigenschaften zählen hohe Dynamik, begrenzte Bandbreite, störanfällige und unter Umständen asymmetrische Verbindungen sowie begrenzte Ressourcen der Endgeräte. Die Implementie-

rung einer zentralen und vertrauenswürdigen Instanz zur Verwaltung von (öffentlichen) Schlüsseln der Teilnehmer ist somit unmöglich, falls kein Zugriff auf das ortsfeste Internet existiert.

In diesem Beitrag wird ein Sicherheitskonzept vorgestellt, mit dem die Kommunikation in mobilen Ad-hoc-Netzen geschützt werden kann. Dieses Sicherheitskonzept basiert ähnlich wie das in [ZH99] vorgestellte Konzept zur verteilten Realisierung einer Zertifizierungsinstanz auf der proaktiven Geheimnisteilung und ist an die besonderen Eigenschaften von Ad-hoc-Netzen angepasst. Unser Konzept setzt ein clusterbasiertes Netz voraus, bei dem das Ad-hoc-Netz in einzelne Zellen partitioniert ist. Verglichen mit [ZH99], wo sehr detailliert die Frage der verteilten Schlüsselverwaltung erörtert wird, geht unser Beitrag in verstärktem Maße auf das Problemfeld der Autorisierung und Rechtevergabe ein. Zudem wird ein vierstufiges Sicherheitsmodell eingeführt, das eine an die Fähigkeiten der Endgeräte anpassbare Komplexität ermöglicht. Ein wesentlicher Aspekt des Sicherheitskonzepts ist die konsequente Vermeidung von zentralen Instanzen, da diese zentrale Angriffspunkte darstellen und beim Ausfall keine sichere Kommunikation mehr möglich ist. Ein solches Sicherheitskonzept erfordert eine starke Interaktion mit den übrigen Protokollen, die zur Kommunikation in clusterbasierten Ad-hoc-Netzen notwendig sind. Daher zeigt dieser Beitrag auch Lösungskonzepte, wie diese Kommunikationsprotokolle in unser Sicherheitskonzept integriert werden können.

Der Beitrag ist wie folgt aufgebaut: Im nachfolgenden Abschnitt werden zunächst die Grundlagen von clusterbasierten Ad-hoc-Netzen sowie grundlegende Begriffe aus dem Bereich der Netzwerksicherheit erläutert. Kapitel 3 widmet sich unserem Sicherheitskonzept; es werden auch Mechanismen und Protokollerweiterungen aufgezeigt, die für eine Umsetzung notwendig sind. Abschließend wird das Sicherheitskonzept sowie die wesentlichen Ergebnisse zusammengefasst und ein Ausblick auf weitere anstehende Forschungsarbeiten gegeben.

2 Grundlagen zur Sicherheit in Ad-hoc-Netzen

2.1 Clusterbasierte Ad-hoc-Netze

Als Grundlage für unser Sicherheitskonzept setzen wir ein clusterbasiertes Ad-hoc-Netz voraus. Dabei ist das Netz in einzelne Cluster partitioniert, denen die Knoten logisch (meist abhängig von ihrem Aufenthaltsort) zugeteilt sind (vgl. Abb. 2.1). In jedem Cluster gibt es einen ausgezeichneten Knoten, den *Clusterhead* (CH), der einen Cluster aufbaut und ihn organisiert. Weiterhin gibt es Gateways (GW), die Kontakt zu benachbarten Clustern herstellen.

Die Clusterheads senden in bestimmten Intervallen *CH-Beacons*. Dies sind Nachrichten, die die notwendigen organisatorischen Informationen für die Cluster-Mitglieder enthalten und als Broadcasts gesendet werden. Die CH-Beacons enthalten – neben einer Sequenznummer – eine Liste der zum Cluster gehörenden Knoten. Auch eine Liste der

GWs im Cluster ist in den CH-Beacons enthalten. Zusätzlich senden die GWs in bestimmten Zeitabständen GW-Beacons, um die benachbarten Cluster mitzuteilen.

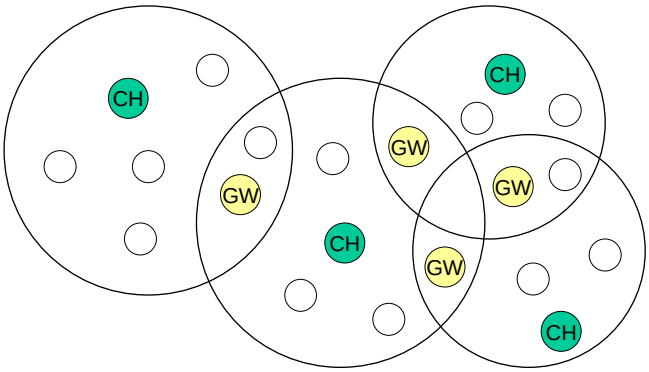

Abbildung 2.1: Clusterbasiertes Ad-hoc-Netz

Durch die Organisation in Cluster zerfällt das Routing-Protokoll in zwei Teile: Intra-Cluster-Routing und Inter-Cluster-Routing. Beim *Zone Routing Protocol* [HPS01a, HPS01b] wird beispielsweise eine Kombination aus proaktivem (*table driven*) und reaktivem (*on-demand*) Routing eingesetzt: Für die Wegewahl innerhalb eines Clusters wird ein proaktives Verfahren verwendet, bei der Wegewahl über die Cluster-Grenzen hinweg kommt ein reaktives Verfahren zum Einsatz. In [Pe01] werden die Grundlagen zu clusterbasierten Netzen erklärt sowie weitere Routing-Algorithmen und Routing-Optimierungen speziell für diese Organisationsstruktur behandelt.

2.2 Allgemeine Ziele der Sicherheit

Gesicherte Kommunikation basiert oft auf der Verwendung von Verschlüsselungsverfahren. Jedoch reicht reine Verschlüsselung für eine Sicherheitsarchitektur allein nicht aus. Hierfür müssen – sowohl im Festnetz als auch in Ad-hoc-Netzen – verschiedene Sicherheitsziele verfolgt werden:

- *Authentizität*: Hierunter versteht man den sicheren Nachweis der Identität der beteiligten Kommunikationspartner bzw. der Herkunft der übertragenen Daten.
- *Integrität*: Ein sicheres System muss die Integrität der übertragenen Daten erhalten, d.h. es muss dafür sorgen, dass die Daten nicht verfälscht werden – weder durch zufällige Übertragungsfehler, noch durch mutwillige Angriffe.
- *Vertraulichkeit*: Hiermit ist gemeint, dass die übertragenen Daten vor nicht-autorisierten Personen oder Geräten verborgen bleiben.
- *Nichtabstreitbarkeit*: Dieses Ziel bezeichnet die sichere Nachweisbarkeit von Aktionen gegenüber Dritten. Es wird zwischen der Nichtabstreitbarkeit des Sendens, des Empfangs und der Übermittlung unterschieden.
- *Autorisierung*: Hierunter versteht man die Vergabe von Rechten, z.B. zum Zugriff auf Ressourcen oder zum Ausführen von Aktionen. Als Voraussetzung ist zunächst eine Authentisierung nötig, die eigentliche Autorisierung erfolgt dann z.B. über Zugriffskontrolllisten.

- *Verfügbarkeit*: Dies bezeichnet die ständige Präsenz von Diensten und Ressourcen in einem Netz sowie ihren Schutz gegen Ausfälle und Angriffe jeder Art.

Der Authentizität kommt eine besondere Bedeutung zu, da sie die Grundlage für das Erreichen aller anderen Sicherheitsziele bildet: So macht z.B. eine Verschlüsselung der Daten nur dann Sinn, wenn sich die Kommunikationspartner zuvor gegenseitig von der Korrektheit ihrer Identität überzeugt haben. In Ad-hoc-Netzen ist das Kommunikationsmedium – die Luft – Angriffen weitgehend schutzlos ausgesetzt, sodass alle Sicherheitsziele nur mit kryptographischen Verfahren erreicht werden können. Die verlässliche Authentisierung wird daher zu einem zentralen Problem.

2.3 Gängige Authentisierungsverfahren

Es gibt zwei große Klassen kryptographischer Sicherungsmechanismen [Sc96]: Bei *symmetrischen Verfahren* müssen die Kommunikationspartner zunächst über einen sicheren Kanal ein gemeinsames Geheimnis (*Shared Secret*) vereinbaren. Bei *asymmetrischen Verfahren* (*Public-Key-Verfahren*) besitzt jede Instanz ein Paar aus einem öffentlichen und einem geheimen Schlüssel; der öffentliche Schlüssel kann ungesichert verbreitet werden.

Asymmetrische Verfahren eignen sich sehr gut, um das Problem der Authentisierung und des initialen Schlüsselaustauschs für eine gesicherte Kommunikation zu lösen. Allerdings benötigen sie deutlich mehr Rechenzeit als symmetrische Verfahren. Zum sicheren Austausch von Schlüsseln können zum einen asymmetrische Verschlüsselungsverfahren wie ElGamal oder RSA verwendet werden. Zum anderen existieren Algorithmen, mit denen zwei oder mehr Kommunikationspartner ein gemeinsames Geheimnis berechnen können, ohne dass dieses Geheimnis aus den übertragenen Nachrichten gewonnen werden kann. Das bekannteste Verfahren für zwei Kommunikationspartner ist der Diffie-Hellman-Algorithmus [Sc96]. Zur Authentisierung können digitale Signaturalgorithmen wie RSA oder DSA (*Digital Signature Algorithm*) verwendet werden. In die Berechnung einer Signatur gehen die signierte Nachricht (bzw. ein kryptographischer Hash-Wert davon) und der private Signaturschlüssel ein; mit dem öffentlichen Schlüssel kann der Empfänger die Authentizität und Integrität der Nachricht verifizieren: Um die Nachricht zu verfälschen, ohne dass die Signatur ungültig wird, müsste ein Angreifer im Besitz des privaten Schlüssels sein.

Ein Nachteil asymmetrischer Verfahren ist ihr relativ hoher Rechenaufwand. Zur Verschlüsselung, Authentisierung und Integritätssicherung großer Datenmengen sowie zur Implementierung auf ressourcenschwachen Endgeräten sind daher symmetrische Algorithmen besser geeignet. Gängige Algorithmen zur symmetrischen Verschlüsselung sind Blockchiffren wie DES, Triple-DES oder AES und Stromchiffren wie RC4 [Sc96]. Zur symmetrischen Authentisierung und Integritätssicherung übertragener Daten werden schlüsselabhängige Hash-Funktionen, sogenannte *Message Authentication Codes* (MAC), eingesetzt, z.B. HMAC-MD5 oder HMAC-SHA1. Um die benötigten gemeinsamen Schlüssel sicher auszutauschen oder zu generieren, können die symmetrischen Verfahren mit asymmetrischen Verfahren kombiniert werden.

Soll für die Authentisierung mit symmetrischen Schlüsseln über einen unsicheren Kanal überprüft werden, ob ein Client im Besitz eines Geheimnisses (z.B. eines Passwortes) ist, ohne dieses Geheimnis über den unsicheren Kanal zu übertragen, kann ein *Challenge-Response-Verfahren* eingesetzt werden. Dabei sendet der Server dem Client eine Herausforderung (*Challenge*), die dieser mit dem Geheimnis in einer wohldefinierten Weise verknüpft und als Antwort wieder zurücksendet (*Response*). Der Server führt dieselbe Berechnung durch und vergleicht die beiden Ergebnisse miteinander. Somit kann er herausfinden, ob der Antragsteller im Besitz des korrekten Passwortes ist.

Ein Voraussetzung bei der asymmetrischen Authentisierung ist die gesicherte Zuordnung vom öffentlichen Schlüssel zur Identität seines Besitzers. Hierzu wird in stationären Netzen häufig eine *Public Key Infrastructure* (PKI) herangezogen, welche die Identität der Schlüsselinhaber durch digital signierte Zertifikate bestätigt. Mit Hilfe einer PKI und einer geeigneten Kombination von asymmetrischen und symmetrischen Verfahren können die Ziele Authentizität, Vertraulichkeit, Integrität und in gewissen Grenzen auch Nichtabstreitbarkeit erreicht werden. Zur Wahrung der Verfügbarkeit sind weitergehende Mechanismen notwendig. Der Betrieb einer PKI muss nach vorgegebenen Richtlinien (*Policies*) erfolgen. In [Ho99] wird eine PKI definiert: Jeder Anwender muss sich bei einer Registrierungsstelle eindeutig ausweisen und bekommt von einer vertrauenswürdigen *Certification Authority* (CA) ein digital signiertes Zertifikat ausgestellt. Dieses Zertifikat bescheinigt ihm, rechtmäßiger Besitzer seines öffentlichen Schlüssels zu sein.

Das derzeit gebräuchlichste (identitätsbasierte) Zertifikatformat ist ITU-T X.509. Ein solches Zertifikat sagt aus, dass einem bestimmten „Namen“ (Person oder Organisation) ein bestimmter öffentlicher Schlüssel gehört. In Ad-hoc-Netzen ohne Zugriff auf das Internet müssen die Teilnehmer aber nicht notwendigerweise einen gemeinsamen globalen Namensraum teilen. Auch erschwert die Dynamik der Netztopologie die Realisierung einer zentralen Instanz; daher ist eine vertrauenswürdige CA nicht implementierbar. Die Vertrauenswürdigkeit eines digitalen Zertifikats hängt maßgeblich davon ab, wer es ausgestellt hat – schließlich bürgt der Aussteller für die Identität des Besitzers. In Ad-hoc-Netzen ergibt sich somit das Problem, die Vertrauenswürdigkeit benutzter Schlüssel zu überprüfen.

2.4 Geheimnisteilung

Schemata zur Geheimnisteilung schützen Vertraulichkeit und Integrität von Information, indem sie diese Information auf verschiedene Orte verteilen. Die Geheimnisteilung hat für Ad-hoc-Szenarien Vorteile gegenüber den in Abschnitt 2.3 vorgestellten, bislang gängigen Authentisierungsverfahren. Zum Beispiel wird das Geheimnis nicht von einer einzigen zentralen Stelle verwahrt, die gezielter Angriffspunkt sein kann oder die unerwartet ausgeschaltet bzw. unerreichbar werden kann, wie dies in Ad-hoc-Netzen häufig der Fall ist. Eine Möglichkeit für die Realisierung der Geheimnisteilung ist die Schwellwert-Kryptographie [Sh79]. Dabei wird ein Geheimnis D in n Teile D_1, ..., D_n so zerlegt, dass das Wissen über mindestens k Teile ($k \leq n$) es möglich macht, das Geheimnis zu rekonstruieren. Besitzt ein Angreifer max. k–1 Teile, so ist die Rekonstruktion von D für ihn unmöglich. Ein solches Schema wird (k,n)-Schwellwert-Schema genannt. Realisie-

ren lässt es sich beispielsweise durch eine Langrange-Interpolation mit Polynom-Funktionen im $\mathbf{R}^2$: Für k fixe Punkte $(x_1, y_1),..., (x_k, y_k)$ mit unterschiedlichen x_i gibt es genau ein Polynom $f(x)$ vom Grad k–1, so dass für alle i gilt: $f(x_i) = y_i$. Soll D in n Teile zerlegt werden, wird von einem *Trusted Dealer* ein Polynom $f(x) = a_0 + a_1x + ... + a_{k-1}x_{k-1}$ mit $a_0 = D$ gewählt, $D_1 = f(1)$, ..., $D_n = f(n)$ berechnet und auf n Teilnehmer verteilt. Das Polynom muss selbstverständlich geheim bleiben bzw. wieder gelöscht werden. Aus jeder Teilmenge der D_i mit mindestens k Elementen können nun die Koeffizienten a_i des Polynoms $f(x)$ berechnet werden, um schließlich $D = f(0)$ zu berechnen.

Der *Trusted Dealer* stellt jedoch wieder einen zentralen Angriffspunkt dar, der für Ad-hoc-Netze zu vermeiden ist. Dafür bietet es sich an, das Geheimnis von den Teilnehmern in einem Prozess gemeinsam konstruieren zu lassen. So kennt keine einzelne Stelle zu irgend einem Zeitpunkt das komplette Geheimnis. Allerdings muss bei diesem Verfahren sichergestellt werden, dass die Teilnehmer nur korrekte Werte übermitteln und sowohl das Geheimnis als auch die Teilgeheimnisse von den einzelnen Teilnehmern verifiziert werden können (auch *verifizierbare Geheimnisteilung* genannt). In [Pe91] wird ein Verfahren vorgestellt, das keine zentrale Stelle benötigt, die das Geheimnis erstellt und verteilt. Außerdem kann das Geheimnis, das von den Teilnehmern selbst konstruiert wird, so verteilt werden, dass jedes Mitglied der Gruppe dessen Korrektheit verifizieren kann. Diese Eigenschaft ist wichtig, da das Geheimnis nicht von einer vertrauenswürdigen Stelle kommt und die Knoten deshalb nicht generell davon ausgehen können, dass es korrekt berechnet wurde.

Die Grundlage für das hier vorgestellte Sicherheitskonzept ist jedoch die proaktive Geheimnisteilung, bei der ein Angreifer nicht die gesamte Lebensdauer eines Geheimnisses Zeit hat, die notwendigen k Stellen zu kompromittieren. Geheimnisse, die über einen langen Zeitraum benötigt werden (z.B. Signaturschlüssel) können somit besser geschützt werden. Bei der proaktiven Geheimnisteilung werden die Teile des Geheimnisses periodisch geändert, ohne dass das Geheimnis geändert werden muss. Ein Angreifer hat somit nur wenig Zeit zur Verfügung, mindestens k Orte anzugreifen. Nach einer Auffrischperiode sind alle Informationen wertlos, die ein Angreifer bisher über das Geheimnis gewonnen hat. Die proaktive Geheimnisteilung bietet sich vor allem für Einsätze an, bei denen ständig auf das Geheimnis zurückgegriffen werden muss (z.B. auf einen Signaturschlüssel). In [He97] wird ein allgemeines Verfahren beschrieben, wie Schwellwert-Public-Key-Signaturschemata über diskrete Logarithmen in proaktive Signaturschemata transformiert werden können.

Bei der *proaktiven Geheimnisteilung* wird zwischen zwei Zuständen des Systems unterschieden. Im ersten Zustand werden die Teilschlüssel erneuert oder rekonstruiert, falls sie verloren gegangen sind. Im zweiten Zustand werden die angebotenen Funktionen angewandt. Normalerweise wird in einem proaktiven Geheimnisteilungsverfahren die Erneuerung der Teilgeheimnisse immer auf dieselben Teilnehmer im Netz angewandt. In mobilen Ad-hoc-Netzen ändert sich diese Zusammensetzung jedoch häufig. In [DJ97] werden Protokolle beschrieben, wie die bisherigen Inhaber der Teilschlüssel den Schlüssel auf eine Menge an Teilnehmern verteilen können. Die Teilgeheimnisse werden entweder ereignisgesteuert oder periodisch aufgefrischt. Eine Kombination von beiden Arten ist ebenfalls möglich.

Eine geeignete Anwendung für eine proaktive Geheimnisteilung ist die proaktive digitale Signatur. In [Ja95] wird beschrieben, wie eine proaktive CA mittels proaktiver Geheimnisteilung aufgebaut werden kann. [TMT00] stellt ein digitales Signaturschema nach dem ElGamal-Verfahren vor, das (k,n)-Schwellwertkryptographie benutzt und weder einen *Trusted Dealer* noch verschlüsselte Kommunikation während der Signaturausstellung benötigt. Nur während der Schlüsselgenerierung ist ein sicherer Kommunikationskanal notwendig. Ein speziell an den Fall von Ad-hoc-Netzen angepasster Ansatz zur Realisierung einer verteilten CA mittels Geheimnisteilung wird in [ZH99] vorgestellt.

3 Sicherheitskonzept für Ad-hoc-Netze

Die Verwendung einer zentralen PKI ist in einem Ad-hoc-Netz unmöglich: Zum einen stellt sie einen zentralen Angriffspunkt dar, zum anderen kann das Gerät, auf dem sie läuft, spontan ausgeschaltet werden. Dieser Abschnitt widmet sich einem Sicherheitskonzept, das keine zentrale Instanzen für die Verwaltung von Schlüsseln benötigt. Nach einer Zusammenfassung der Anforderungen und Ziele im ersten Teil widmet sich Abschnitt 3.2 dem Konzept selbst. Abschnitt 3.3 beschreibt die notwendigen Erweiterungen der übrigen Kommunikationsprotokolle, damit diese in das Sicherheitskonzept integriert werden können.

3.1 Anforderungen und Ziele

Die Anforderungen an ein Sicherheitskonzept für Ad-hoc-Netze (die teilweise bereits weiter oben genannt wurden) sind die folgenden:

- *Sicherheit:* Authentizität, Integrität und Vertraulichkeit übertragener Nachrichten bzw. Kommunikationspartner müssen gewährleistet werden können.

- *Offenheit:* Das Netz soll offen für neue Knoten sein, d.h. es soll auch für Knoten, die dem Netz bisher nicht bekannt sind, möglich sein, dem Netz beizutreten.

- *Selektive Rechtevergabe:* Nutzungs- bzw. Zugriffsrechte auf Dienste und Ressourcen, die von einzelnen Knoten innerhalb des Ad-hoc-Netzes oder auch vom Netz als Ganzes angeboten werden, sollen feingranular an einzelne Knoten vergeben werden können.

- *Verfügbarkeit:* Die Funktionsfähigkeit des Netzes und insbesondere der Sicherheitsinfrastruktur sollte möglichst schwer angreifbar sein. Besonders anfällig in dieser Hinsicht sind zentrale Komponenten, die aus diesem Grund vermieden werden sollten.

- *Unterstützung von Dynamik und Skalierbarkeit:* Ad-hoc-Netze weisen u.U. eine hohe Dynamik der Mitgliederzusammensetzung auf. Alle Verfahren und Komponenten eines Sicherheitskonzepts müssen so ausgelegt sein, dass sie sowohl schnelle Änderungen als auch schwer vorhersehbare Teilnehmerzahlen gut verkraften.

3.2 Konzepte und Überblick

Das vorgestellte Sicherheitskonzept für Ad-hoc-Netze besteht aus mehreren Komponenten, die in den folgenden Unterabschnitten vorgestellt werden: Neben einer clusterübergreifenden Zertifizierungsinfrastruktur, welche die Grundlage für die Ende-zu-Ende-Sicherung übertragener Daten mittels Public-Key-Verfahren liefert, ist davon unabhängig eine Sicherung einzelner Übertragungsstrecken mittels eines clusterinternen symmetrischen Schlüssels vorgesehen. Die Rechtevergabe erfolgt über Autorisierungszertifikate.

3.2.1 Clusterübergreifende Zertifizierungsinfrastruktur

Grundlage des Sicherheitskonzepts ist die Verwendung asymmetrischer kryptographischer Verfahren (Public-Key-Verfahren) zur Sicherung von Authentizität, Integrität und Vertraulichkeit von Nachrichten bzw. Kommunikationspartnern. Jeder am Netz beteiligte Knoten besitzt dazu ein (selbstgeneriertes) Schlüsselpaar, mit dessen Hilfe eine Ende-zu-Ende-Sicherung zwischen beliebigen Knoten erfolgen kann. Öffentliche Schlüssel werden dabei in Zertifikaten weitergegeben.

Zertifikate werden durch eine vertrauenswürdige Zertifizierungsinstanz erstellt. Diese wird hier jedoch im Unterschied beispielsweise zu gebräuchlichen PKIs im Internet nicht als zentrale Instanz (bzw. als Hierarchie zentraler Instanzen), sondern – wie auch in [ZH99] – verteilt realisiert: Eine ausgezeichnete Teilmenge der Netzknoten bildet gemeinsam die Zertifizierungsinstanz. An der Erstellung von Zertifikaten muss jeweils ein bestimmter Anteil (z.B. die Mehrheit) dieser ausgezeichneten Knoten aktiv teilnehmen. Dieses Vorgehen bietet zwei Vorteile: Einerseits verbessert sich die Verfügbarkeit der Sicherheitsinfrastruktur, da auch noch Zertifikate erstellt werden können, wenn zeitweise nicht alle Knoten der Zertifizierungsinstanz verfügbar sind, andererseits erhöht sich auch die Resistenz der Sicherheitsinfrastruktur gegen Angriffe, da die Kompromittierung einzelner Knoten toleriert werden kann.

In den hier betrachteten clusterbasierten Netzen bietet es sich an, die Rolle der Zertifizierungsinstanz den Clusterheads (CH) zuzuweisen. Auch wenn dies nicht zwingend erforderlich ist, wird im Folgenden von dieser Zuordnung ausgegangen; die Zertifizierungsinstanz wird also gemeinsam durch alle CHs des Netzes gebildet. Bezüglich der zur Erzeugung, Verwaltung und Verwendung des gemeinsamen Zertifizierungsschlüssels erforderlichen besonderen Protokolle bilden die CHs damit eine Art logisches Netzwerk, welches im Folgenden auch als Clusterhead-Netz (CH-Netz) bezeichnet wird.

Der private Schlüssel der Zertifizierungsinstanz ist über alle CHs verteilt, d.h. jeder CH besitzt ein Fragment des Gesamtschlüssels. Die Schlüsselfragmente werden mit Hilfe des in Abschnitt 2.4 beschriebenen Verfahrens der proaktiven Geheimnisteilung nach dem Digital Signature Scheme [Ge96] erzeugt. Abbildung 3.1 zeigt die Geheimnisteilung mittels eines (2,4)-Schwellwertverfahrens, bei dem Knoten N ein Zertifikat ausgestellt bekommt. Der private Signaturschlüssel D ist in vier Teile D_1, ..., D_4 über die CHs aufgeteilt. In diesem Beispiel genügen zwei CHs, um ein Zertifikat auszustellen. Die einzelnen Teile des Zertifikats werden von N mit der bekannten Funktion zusammenge-

setzt. Der zugehörige öffentliche Schlüssel ist jedem CH vollständig bekannt und wird mit Hilfe der CH-Beacons an alle Netzteilnehmer verteilt.

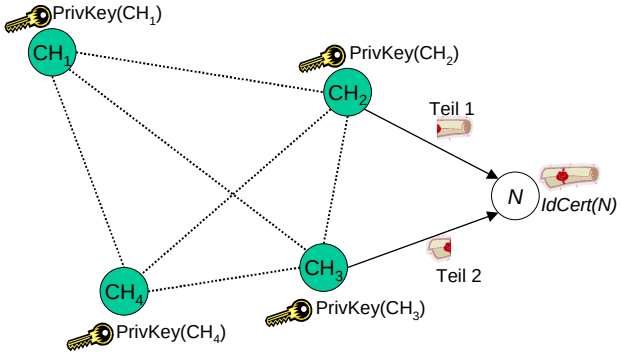

Abbildung 3.1: Beispiel eines CH-Netzes mit (2,4)-Schwellwertverfahren

3.2.2 Clusterinterne Sicherung

Unabhängig von den Möglichkeiten zur Ende-zu-Ende-Sicherung, die sich durch die Schlüsselpaare der einzelnen Knoten und die zugehörigen (vom CH-Netz erstellten) Zertifikate ergeben, ist außerdem eine clusterinterne Sicherung mit Hilfe eines symmetrischen Schlüssels vorgesehen, der allen Mitgliedern eines Clusters bekannt ist. Beispielsweise kann dieser Schlüssel zur Verschlüsselung sämtlicher auf den Strecken zwischen einzelnen Knoten übertragenen Daten verwendet werden („Link-zu-Link-Verschlüsselung“). Damit werden beispielsweise – zusätzlich zum Schutz der eigentlichen Nutzdaten – auch Informationen über Quelle, Ziel und Weg transportierter Pakete gegen Einsichtnahme durch Außenstehende und auch durch Mitglieder anderer Cluster im gleichen Ad-hoc-Netz geschützt. Die clusterinterne symmetrische Verschlüsselung entspricht bezüglich ihres Nutzens in etwa der beispielsweise bei IEEE 802.11oder Bluetooth eingesetzten Verschlüsselung und kann diese entweder ersetzen (um beispielsweise Sicherheitslücken durch zu geringe standardisierte Schlüssellängen zu schließen) oder mit ihr integriert werden (etwa zur Verteilung der dort eingesetzten Schlüssel).

Die Verwaltung und Verwendung des clusterinternen Schlüssels wird durch Sicherheitsrichtlinien geregelt, die vom jeweiligen CH vorgegeben werden.

3.2.3 Aufnahme neuer Knoten und Verwaltung von Rechten

Wird ein Knoten in einen Cluster aufgenommen, so hat er zunächst den Status eines Gastknotens ohne jegliche Rechte. Erst wenn der öffentliche Schlüssel des Knotens vom Netz signiert wurde (also die CH ihm nach Abschluss einer ausreichenden Authentisierung ein Zertifikat ausgestellt haben), ist er volles Cluster-Mitglied und kann dann weitergehende Rechte über so genannte Autorisierungszertifikate zugewiesen bekommen.

Autorisierungszertifikate können – im Unterschied zu den vom CH-Netz ausgestellten identitätsbezogenen Schlüsselzertifikaten – von beliebigen Knoten des Gesamtnetzes ausgestellt werden. Sie autorisieren den Zertifikatinhaber jeweils dazu, auf bestimmte Dienste oder Ressourcen (z.B. Übergang in das Internet, Drucker, etc.) zuzugreifen, die vom Zertifikataussteller verwaltet werden. Diese durch Zertifikate verbrieften Zugriffsrechte können transitiv weitergegeben werden, sofern der weitergebende Knoten das Recht hierzu erhalten hat.

Sowohl zur Durchführung der initialen Authentisierung des Knotens gegenüber dem Netz als auch zur Einschätzung seiner Vertrauenswürdigkeit bei der Vergabe von weiteren Rechten werden (sofern der Knoten nicht beispielsweise Zertifikate externer, dem Netz bekannter Zertifizierungsstellen vorweisen kann) Vertrauensbeziehungen benötigt, die für neue, dem Netz nicht bekannte Knoten a priori nicht zur Verfügung stehen. Der neue Knoten kann und muss sich deshalb zunächst so genannte Bürgschafts-Zertifikate von anderen Knoten des Netzes beschaffen. Dies können z.B. unmittelbar benachbarte Knoten sein, bei denen ein direkter Austausch zwischen deren menschlichen Benutzern und damit eine Authentisierung auf anderer Ebene möglich ist. Eine gewisse Anzahl an Bürgschaften wird vom Netz schließlich als ausreichende Authentisierung anerkannt. Die Bürgen üben damit die Funktion einer Registrierungsstelle (bei herkömmlichen PKIs) aus.

3.3 Verfahren und Konzepte

Die Umsetzung des vorgestellten Sicherheitskonzepts erfordert zusätzliche Mechanismen und Erweiterungen in den übrigen Protokollen, die zur Kommunikation in einem clusterbasierten Ad-hoc-Netz notwendig sind. In den nachfolgenden Abschnitten werden Lösungskonzepte aufgezeigt, die eine Integration dieser Protokolle in unser Sicherheitskonzept möglich machen.

3.3.1 Schlüsselverteilung und Schlüsselauffrischung

Das von den CHs aufgespannte CH-Netz besitzt ein eigenes Public-Key-Schlüsselpaar. Der private Netzschlüssel wird dabei im CH-Netz mittels proaktiver Geheimnisteilung nach dem *Digital Signature Scheme* (DSS) [Ge96] erzeugt (vgl. Abbildung 3.1). Da sich die Zusammensetzung des CH-Netzes ständig dynamisch verändert, müssen auch die Teilgeheimnisse ständig neu angepasst und verteilt werden. Dieser Aufwand fällt bei jedem Eintritt bzw. Austritt eines CHs an. Im vorgestellten Sicherheitskonzept wird dieser Vorgang deshalb mit der Schlüsselauffrischung kombiniert: Ändert sich die Struktur des CH-Netzes, so werden auch die Teilgeheimnisse aufgefrischt. Zur Auffrischung der Teilgeheimnisse können beispielsweise die in [ZH99] vorgestellte Techniken verwendet werden.

Der öffentliche CH-Netzwerkschlüssel muss jedem Knoten im Ad-hoc-Netz bekannt sein und wird über die CH-Beacons propagiert. Ein CH-Beacon von CH_a besteht daher aus den öffentlichen Schlüsseln von CH und CH-Netz, einer Liste der Knoten im Cluster samt deren Status, und eine Liste von Gateways zu benachbarten Clustern.

CH-Beacon(CH_a): PubKey(CH_a), PubKey(CH-Network), Stat(N_1),..,Stat(N_i), Gw_1,..,Gw_j

Auch die Gateways müssen in bestimmten Abständen Beacons verschicken, damit sich die Knoten im Cluster ein Bild vom Netz machen können um ggf. ihre Wegewahl-Entscheidungen zu treffen. Im GW-Beacon nennt Gateway Gw_b seinen öffentlichen Schlüssel, die benachbarten Cluster sowie seinen Status im aktuellen Cluster.

GW-Beacon(Gw_b): PubKey(GW_b), C_1,...,C_n, Stat(Gw_b)

3.3.2 Anmeldeprozedur

Wird ein Knoten in einen Cluster aufgenommen, so hat er zunächst den Status eines Gastknotens ohne jegliche Rechte. Erst wenn der öffentliche Schlüssel des Knotens vom CH-Netz signiert wurde (also die CHs ihm ein Zertifikat ausgestellt haben), ist der Knoten volles Cluster-Mitglied und kann weitergehende Rechte über Autorisierungszertifikate zugewiesen bekommen. Dieser Vorgang der Anmeldung läuft wie folgt ab: Kommt ein neuer Knoten *A* in das Netz, sucht er sich zunächst einen Cluster. Empfängt *A* CH-Beacons, so sendet er seine Bewerbung an den verantwortlichen CH. Daraufhin handelt *A* mit dem CH in einem Handshake-Verfahren die erforderlichen Sicherheitsrichtlinien aus und ist somit Gastknoten. Empfängt *A* hingegen keine CH-Beacons, bildet er einen eigenen Cluster und ernennt sich selbst zum CH. Er generiert einen geheimen symmetrischen Cluster-Schlüssel und sendet selbst CH-Beacons.

Die Authentisierung ist Grundlage für alle weiteren Komponenten. Es genügt im Allgemeinen nicht, dass sich ein Knoten am Netz authentisiert; das Netz muss sich auch beim Knoten authentisieren. Ein Angreifer könnte sonst ein anderes Netz simulieren und so eventuell versuchen, bei Anmeldeversuchen von Knoten Geheimnisse zu erkunden. In dem vorgestellten Sicherheitskonzept kommen Bürgschaftszertifikate (*WarrantCert*) zum Einsatz, die ein Knoten von einem Bürgen *S* erwerben kann (vgl. Abbildung 3.2). Mit diesem Zertifikat kann sich *A* beim CH-Netz für eine Signatur anmelden. Ein für *A* ausgestelltes *WarrantCert* beinhaltet auch einen Zeitraum, in dem das Zertifikat gültig ist. Das gesamte Zertifikat wird mit einer Signatur von *S* (*Sign(S)*) unterzeichnet, um dessen Echtheit zu gewährleisten. Somit umfasst es folgende Informationen:

WarrantCert(A): Node(A), PubKey(A), Validity(t), Fct("S warrants for A"), Sign(S)

Allerdings wird ein Bürge nur dann für einen Knoten bürgen, wenn er sich dessen wahrer Identität sicher ist. Dies kann z.B. mittels eines Zertifikates einer gemeinsam vertrauten Wurzelzertifizierungsstelle sein, mittels physikalischem Kontakt oder über einen sicheren Kanal. Die Authentisierung wird also dezentral auf die Knoten verteilt. Abbildung 3.2 zeigt die Vorgänge einer erfolgreichen Authentisierung. Je mehr Bürgen ein Knoten für sich finden kann (also je mehr *WarrantCert* er sammeln kann), desto sicherer ist seine Identität, d.h. das CH-Netz kann für mehrere Bürgschaften dem neuen Knoten mehr oder höhere Rechte einräumen.

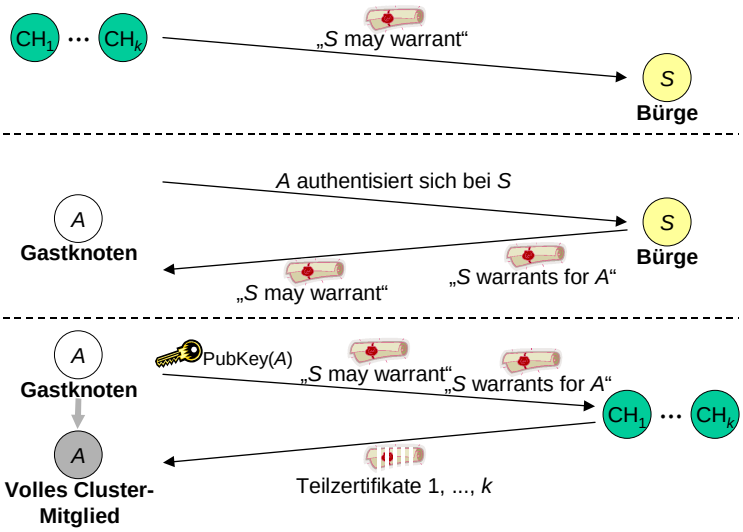

Abbildung 3.2: Authentisierung

Bevor die CHs ihre Teilzertifikate an *A* senden, müssen sie erst sicherstellen, dass der Bürge *S* auch zum Bürgen autorisiert war. Dies geschieht durch ein Autorisierungszertifikat *WarrantAuthCert*, das der Bürge vom CH-Netz bekommt. Von diesem Zertifikat sendet der Bürge eine Kopie an *A*. Mit beiden Zertifikate (*WarrantCert*, *WarrantAuthCert*) kann sich *A* beim CH-Netz bewerben und seine Schlüssel zertifizieren lassen. Ein *WarrantAuthCert* für Knoten *S* sieht dabei folgendermaßen aus:

WarrantAuthCert(S): Node(S), PubKey(S), Fct(“S may warrant”), Sign(CH-Network)

Die CHs überprüfen ihrerseits das vom bewerbenden Knoten *A* gesendete *WarrantAuthCert* und senden *A*, falls alles seine Richtigkeit hat, ihren Teil des Identifikationszertifikats *IdCert*. Erhält *A* genügend Teilzertifikate, so kann er das vollständige *IdCert* abschließend zusammensetzen. Das *IdCert* ist somit vom CH-Netz signiert und enthält folgende Informationen:

IdCert(A): Node(A), PubKey(A), Validity(t), Sign(CH-Network)

Damit ist er im Besitz seines eigenen signierten Schlüssels und somit volles Cluster-Mitglied. Der CH sendet *A* den symmetrischen Cluster-Schlüssel zu (mit dem signierten öffentlichen Schlüssel von *A* verschlüsselt).

Der Vorgang der Bürgschaft mit anschließender Schlüsselzertifizierung geht sparsam mit den beschränkten Ressourcen um. Es genügen nur wenige Nachrichten, um in den Besitz eines Schlüsselzertifikats zu kommen. Unter der Voraussetzung, dass die Authentisierung eines neuen Knotens bei dem Bürgen durch physikalischen Kontakt oder über einen sicheren Kanal stattfindet, werden bei dem verwendeten (k,n)-Schwellwert-Kryptosystem insgesamt $2k + 2$ Nachrichten ausgetauscht:

- Die (mit dem öffentlichen Schlüssel des Bürgen) verschlüsselte Anfrage des neuen Knotens beim Bürgen.

- Die (mit dem öffentlichen Schlüssel des neuen Knotens) verschlüsselte Antwort des Bürgen, welche die Zertifikate *WarrantCert* und *WarrantAuthCert* beinhaltet.

- Die (mit dem öffentlichen Schlüssel des jeweiligen CHs) verschlüsselte Anfrage bei den *k* Clusterheads, die neben den beiden Zertifikaten *WarrantCert* und *Warrant-AuthCert* auch den öffentlichen Schlüssel des neuen Knotens enthalten.

- Die *k* (mit dem öffentlichen Schlüssel des neuen Knotens) verschlüsselten Antworten der CHs, die einen Teil des *IDCert* und ggf. weitere Autorisierungszertifikate beinhalten.

3.3.3 Verknüpfung mit Routing und Gateways

In clusterbasierten Ad-hoc-Netzen wird zwischen zwei Arten des Routings unterschieden: Intra-Cluster- und Inter-Cluster-Routing. Bei ersterem findet die Kommunikation innerhalb eines Clusters statt, d.h. der Zielknoten befindet sich im gleichen Cluster. Wird hierfür ein proaktiver Routing-Algorithmus verwendet, kann der Sender über ein Flag im Paketkopf entscheiden, ob das Paket nur Wege über volle Cluster-Mitglieder nehmen darf, oder ob auch Gastknoten das Paket weiterleiten dürfen. Somit muss jeder Knoten unter Umständen zwei Tabellen verwalten: eine für Routen ausschließlich über volle Cluster-Mitglieder und eine für allgemeine Routen. Reaktive Verfahren haben den Vorteil, dass der Sender den Weg schon vor dem Versenden der Pakete kennt. Der Status der beteiligten Knoten entlang des Weges wird in den CH-Beacons mitgeteilt. Der Sender kann schon beim *RouteRequest* durch spezifizierte Sicherheitsansprüche die Antworten auf volle Cluster-Mitglieder einschränken. Die Sicherheitsrichtlinien beim Intra-Cluster-Routing werden von jedem Clusterhead individuell für seinen Cluster festgelegt. Dazu zählen z.B. verschlüsselte *LinkState Updates* bei proaktiven Routing-Algorithmen.

Beim Intercluster-Routing, also der Wegewahl zwischen Clustern, müssen ebenfalls Anpassungen vorgenommen werden. Sowohl bei reaktiven als auch bei proaktiven Routing-Verfahren muss der Sender über ein Flag im Paketkopf dem Gateway mitteilen, ob nur über volle Cluster-Mitglieder weitergeleitet werden darf. Da das Gateway auch Mitglied in benachbarten Clustern ist, kennt es den Status der dortigen Knoten.

Bekommt ein Knoten *N* erstmals Kontakt zu einem anderen Cluster, kann er die Rolle eines Gateways übernehmen. Hat sich der Knoten zuvor schon am entdeckten Netz authentisiert, ist er ggf. bereits im Besitz der Berechtigung, als Gateway zwischen den Clustern zu fungieren. Diese Berechtigung wird durch ein vom CH-Netz signiertes Gateway-Authentisierungszertifikat *GwAuthCert* realisiert.

GwAuthCert: Node(N), PubKey(N), Fct("Gateway"), Sign(CH-Network)

Der potentielle Gateway-Knoten sendet nun Informationen über den jeweils benachbarten Cluster an die zugehörigen CHs. Die Adresse des neuen CHs kann er von dem Kno-

ten erfahren, von dem er erstmals die neue Nachricht empfangen hat. Die CHs senden dann die Information über das neue Gateway über ihre CH-Beacons aus. Das Gateway sendet ab sofort eigene GW-Beacons, in denen es seinen Status in den jeweiligen Clustern (Gastknoten oder volles Cluster-Mitglied, Besitz eines *GwAuthCertificate*) angibt. Bei einem neuen Cluster, der noch nicht zum Netz gehört, ist der Knoten zunächst auf jeden Fall Gastknoten. Dies kann dazu führen, dass er in seinem bisherigen Cluster volles Cluster-Mitglied ist, im zweiten (rechten) aber nur Gastknoten (vgl. Abbildung 3.3).

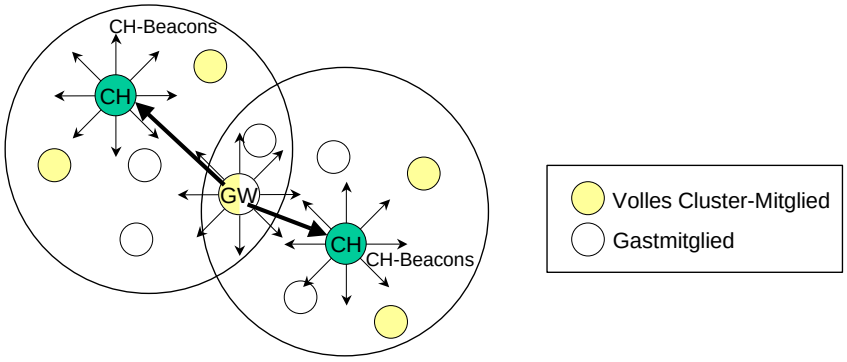

Abbildung 3.3: Gateway bei der Entdeckung eines neuen Clusters

3.3.4 Entdecken der Clusterheads und Wechseln von Clusterheads

Um seinen Schlüssel vom CH-Netz signieren zu lassen, muss ein Knoten bei dem (*k*,*n*)-Schwellwertverfahren mindestens *k* CHs kennen, von denen er Teilzertifikate anfordern kann. Er stellt dazu eine Anfrage an den eigenen CH. Da dieser in ständigem Kontakt zu den übrigen CHs im CH-Netz steht, kann er dem neuen Knoten eine Liste der übrigen CHs senden. Alternativ kann der neue Knoten über entsprechende *GW-Beacons* (vgl. Abschnitt 3.3.3), die er empfängt, weitere Cluster ausfindig machen.

Sollte ein CH nicht mehr in der Lage sein, seine Rolle auszuüben, muss er seine Funktionen an einen anderen vertrauten Knoten abgeben. Damit wird eine aufwändige Neukonfiguration des Clusters vermieden. Hat er einen Knoten gefunden, müssen sowohl die Knoten im Cluster als auch das CH-Netz davon in Kenntnis gesetzt werden. Der alte CH sendet einen Broadcast in den eigenen Cluster, in dem er den Knoten mitteilt, dass er seine Funktion aufgibt und anfragt, welcher Knoten diese Aufgabe zukünftig übernehmen wird. Abgesichert durch seine Signatur kann jeder Knoten im Cluster, der den Broadcast empfängt, sich auf den neuen CH einstellen, von dem er in Zukunft die CH-Beacons empfangen wird. Knoten, die den Broadcast nicht mitbekommen haben, werden in Zukunft fremde CH-Beacons empfangen. Dennoch bleiben sie volles Clustermitglied, da ihre Netzzertifikate weiterhin gültig sind.

Auch den Mitgliedern des CH-Netzes muss der alte CH den Wechsel mitteilen. Dies geschieht über einzelne, verschlüsselte Nachrichten an die Mitglieder, die sich daraufhin auf die neue Situation einstellen. Da der alte CH seinen geheimen Teilschlüssel an den

neuen weitergibt, ändert sich nicht die Verteilung des geheimen Cluster-Schlüssels. In weiteren Auffrischphasen für die Teilschlüssel kann mit dem neuen CH kommuniziert werden. Alternativ kann der ausscheidende Knoten als ausfallender Knoten betrachtet werden und der neue CH wird in das CH-Netz aufgenommen. Gerade dieser Aufwand soll aber vermieden werden, indem die Rolle des CHs explizit abgegeben wird.

3.3.5 Verschmelzung von Clustern

Ein Problem stellt das Aufeinandertreffen von zwei Cluster-Netzen dar. Im Falle einer Eingliederung muss ein bestehender Netzwerkschlüssel auf das andere Cluster-Netz neu verteilt werden, da eine Mischung nicht möglich ist (bzw. ein dritter neuer Netzwerk-Schlüssel zur Signatur entstehen würde). Im einfachsten Fall trifft ein einzelner Cluster auf ein Cluster-Netz. Bei etwa gleich großen Clustern besteht der geringste Aufwand darin, den einzelnen CH in das CH-Netz zu integrieren. Es kann aber natürlich auch vorkommen, dass zwei etwa gleich große Ad-hoc-Netze aufeinander treffen. In diesem Fall muss abgewägt werden, was weniger Aufwand bedeutet. Maßgeblich ist die Anzahl der neuen CHs, die Teile des privaten Netzschlüssels bekommen, sowie die Anzahl der Knoten, die ein neues Netz-Zertifikat für ihre Schlüssel beantragen müssen. Je nach Anzahl der Bürgschaften müssen auch weitere Zertifikate neu ausgestellt werden. Zunächst muss aber eine gegenseitige Einwilligung zur Migration stattgefunden haben: Zum einen muss das bestehen bleibende Ad-hoc-Netz dem neuen Netz vertrauen, zum anderen muss das neue Netz der Migration zustimmen, da seine Zertifikate danach ungültig sind.

Einem CH aus einem fremden Cluster muss von einer bestimmten Anzahl an Knoten aus dem eigenen Cluster getraut werden, damit der CH in das CH-Netz aufgenommen werden kann. Hat er genügend *WarrantCert* erhalten, bekommt er in der nächsten Auffrischphase seinen Teil des privaten Netzschlüssels. Bekommt er nicht genügend Zertifikate zusammen, muss er seine Rolle als CH an einen anderen Knoten aus seinem Cluster abgeben, der genügend Zertifikate besitzt. Ist dies auch nicht möglich, können die beiden Cluster-Netze nicht verschmolzen werden; die Knoten müssen sich dann einzeln neu am Cluster anmelden. Ist geklärt, welches Netz integriert werden soll, werden die Knoten des zu integrierenden Netzes dazu aufgefordert, ihre Schlüssel neu signieren zu lassen. Eventuell ist dann auch der Schwellwert neu anzupassen.

3.3.6 Rechtevergabe

Der Zugriff auf Dienste und Ressourcen geschieht in unserem Sicherheitskonzept über Autorisierungszertifikate. Dabei kann ein Dienste-/Ressourcenanbieter (DR-Anbieter) Autorisierungszertifikate an potentielle Nutzer (DR-Nutzer) ausgeben. Diese Zertifikate enthalten zusätzlich zum öffentlichen Schlüssel die Autorisierungsinformation. Eventuell können sie auch von den Knoten transitiv weitergegeben werden, wenn sie das Recht dazu besitzen. Daraus resultiert, dass sich DR-Anbieter und DR-Nutzer in einer vierstufigen Hierarchie gegenseitig vertrauen können (vgl. Abbildung 3.4). Diese Hierarchie kann von jedem Knoten unabhängig vom CH-Netz genutzt werden; jeder kann vollständig autark über die eigenen Dienste/Ressourcen verfügen.

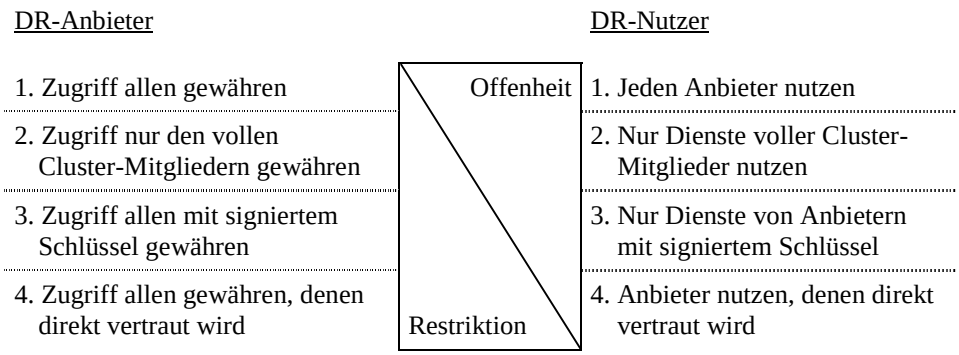

Abbildung 3.4: Vierstufige Hierarchie der Rechtevergabe

Durch die verschiedenen Schlüsselarten (symmetrischer Cluster-Schlüssel, asymmetrische öffentliche Schlüssel) gibt es auch mehrere Möglichkeiten, die anstehende Kommunikation zu verschlüsseln. In unserem Konzept wird ein vierstufiges Sicherheitsmodell verfolgt, das die verschiedenen Schlüsselarten und Zertifikate verwendet. Die Knoten entscheiden dann von Fall zu Fall, welche Sicherheit ihnen für die jeweilige Verbindung notwendig erscheint und benutzen die dafür vorgesehene Verschlüsselung:

1. *Keine Verschlüsselung:* Die Pakete werden unverschlüsselt übertragen.

2. *Geheimer Cluster-Schlüssel:* Pakete werden mit dem geheimen Cluster-Schlüssel verschlüsselt. Da dieser nur den Knoten im eigenen Cluster bekannt ist, kann diese Verschlüsselung nur bei der Kommunikation im Cluster verwendet werden.

3. *Eigenes Schlüsselpaar:* Die Knoten können sich selbst ein eigenes Public-Key-Schlüsselpaar generieren und dieses bei der Kommunikation benutzen. Der Sender besorgt sich den öffentlichen Schlüssel entweder direkt beim Empfänger oder von einer anderen Stelle und verschlüsselt damit seine Daten.

4. *Zertifizierung:* Knoten lassen sich ihren öffentlichen Schlüssel vom CH-Netz zertifizieren. Das Zertifikat bindet den öffentlichen Schlüssel nur an den Knoten; dessen wahre Identität ist jedoch unbekannt. Der Knoten muss sich erst beim Netz authentisieren und seine Identität feststellen lassen, bevor er ein solches *IdCert* erhält. Dies ist die höchste Stufe der Sicherheit.

Der wesentliche Vorteil des vierstufigen Sicherheitsmodells ist die anpassbare Komplexität, die es auch ressourcenschwachen Stationen ermöglicht, eine angepasste Verschlüsselung zu wählen. Allerdings kann dann auch eventuell keine Verbindung zustande kommen, wenn sich die Stationen nicht über einen minimalen Sicherheitsstandard einigen können.

4 Zusammenfassung und Ausblick

In dieser Arbeit wird ein besonders an die Gegebenheiten in Ad-hoc-Netzen angepasstes Konzept zur Realisierung einer verteilten Public-Key-Infrastruktur vorgestellt. Um die hohe Dynamik der Netztopologie und die stark schwankende Qualität der Verbindungen in Ad-hoc-Netzen zu berücksichtigen, kommt dieses Konzept ohne zentrale Infrastrukturkomponenten aus. Statt dessen werden die Aufgaben einer Zertifizierungsstelle auf alle Clusterheads im Netz verteilt, wobei zur Verteilung des geheimen Signaturschlüssels Mechanismen zur proaktiven Geheimnisteilung eingesetzt werden. Die Realisierung dieser verteilten Zertifizierungsstelle entspricht im Wesentlichen der aus [ZH99], wobei die Clusterheads als Schlüsselverwaltungsserver fungieren.

Anstelle von Registrierungsstellen können beliebige Netzknoten als Bürgen für die Identität von neu hinzugekommenen Knoten fungieren, sofern sie ein entsprechendes Bürgen-Autorisierungszertifikat vorweisen können. Aufbauend auf dieser Authentisierungsinfrastruktur wird eine vierstufige Hierarchie von Sicherungsmechanismen realisiert, durch welche die Sicherheitsziele Authentizität, Integrität und Vertraulichkeit erreicht werden können.

Der Prozess der Authentisierung wird in einem zweistufigen Authentisierungsverfahren realisiert, bei dem die Knoten zunächst den Status eines Gastknotens und später – nach ausreichender Authentisierung – den eines vollen Cluster-Mitglieds einnehmen können. Mittels dieser Differenzierung ist eine einfachere Autorisierung möglich, die durch Autorisierungszertifikate realisiert wird. Auch hierfür ist eine zentrale Verwaltung nicht notwendig.

Ein weiteres Merkmal ist die Möglichkeit zur Delegation der Clusterhead-Funktion an weitere Knoten im Cluster. Dadurch wird die weitere Existenz eines Clusters gewährleistet, falls ein Knoten seine Aufgaben als Clusterhead nicht mehr erfüllen kann oder will; die Robustheit des Ad-hoc-Netzes wird damit verbessert. Ein weiterer Anwendungsfall in diesem Zusammenhang ist die Eingliederung eines Clusters in ein bestehendes Ad-hoc-Netz.

Das Konzept der Bürgschaften mit anschließender Schlüsselzertifizierung geht sparsam mit den beschränkten Netzressourcen um. Es genügen wenige Nachrichten, um in den Besitz eines Schlüsselzertifikates zu gelangen: Für die Zertifikatsausstellung werden von dem Gastknoten beim (k,n)-Schwellwertverfahren insgesamt k Nachrichten für die minimal benötigte Anzahl an Schlüsselfragmenten generiert; nach positiver Überprüfung werden k Nachrichten versendet, diesmal von den Clusterheads an den neuen Knoten.

Für die Zukunft ist geplant, das in dieser Arbeit vorgestellte Sicherheitskonzept in der Praxis zu realisieren und zu evaluieren. Dabei ist auch zu untersuchen, wie der Parameter k in dem verwendeten (k,n)-Schwellwert-Kryptosystems in Abhängigkeit von n zu wählen ist, um hinreichende Sicherheit gewährleisten zu können. Zusätzlich untersuchen wir derzeit das zugrunde liegende Vertrauensmodell anhand typischer Szenarien, um daraus Strategien für die Besetzung der Schlüsselrollen (Clusterhead, Gateway) und die Vergabe von Rechten ableiten zu können. Weitere Arbeiten sind unter anderem Untersu-

chungen, wie die vorgestellten Sicherungsmechanismen mit ortsfesten Authentisierungsmechanismen (z.B. einer PKI im Internet) verknüpft werden können, falls das Ad-hoc-Netz über spezielle Gateway-Knoten zumindest vorübergehende Verbindung zum Internet hat.

5 Literaturverzeichnis

[DJ97] Y. Desmedt, S. Jajodia: Redistributing secret shares to new access structures and its applications. Technical Report ISSE TR-97-01, George Mason University, 1997

[Ge96] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin: Robust Threshold DSS Signatures. Advances in Cryptology, Proc. of Eurocrypt'96, 1996.

[HPS01a] Z. J. Haas, M. R. Pearlman, P. Samar: The Interzone Routing Protocol (IERP) for Ad Hoc Networks <draft-ietf-manet-zone-ierp-01.txt>. Internet Draft (expired), IETF, 2001.

[HPS01b] Z. J. Haas, M. R. Pearlman, P. Samar: The Intrazone Routing Protocol (IARP) for Ad Hoc Networks <draft-ietf-manet-zone-iarp-01.txt>. Internet Draft (expired), IETF, 2001.

[He97] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, M. Yung: Proactive Public Key and Signature Systems. ACM Conference on Computer and Communication Security, 1997.

[Ho99] R. Housley, W. Ford, W. Polk, D. Solo: Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC 2459, IETF, Januar 1999.

[Ja95] S. Jarecki: Proactive Secret Sharing and Public Key Cryptosystems. Advances of Cryptography, Proc. of the CRYPTO'95, 1995.

[Pe91] T. P. Pedersen: A Threshold Cryptosystem without a Trusted Party. Advances in Cryptology, Proc. of the Eurocrypt'91, 1991.

[Pe01] C. E. Perkins: Ad Hoc Networking. Addison-Wesley, 2001.

[Sc96] B. Schneier: Applied Cryptography. John Wiley, 1996.

[Sh79] A. Shamir: How to share a Secret. ACM Communications, Vol. 22, No. 11, 1979.

[TMT00] K. Takaragi, K. Miyazaki, M. Takahashi: A Threshold Digital Signature Issuing Scheme without Secret Communication. Submission to the IEEE P1363 Study Group for Future Public-Key Cryptography Standards, November 2000.

[ZH99] L. Zhou, Z. J. Haas: Securing Ad Hoc Networks. IEEE Network, November/December 1999.