TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG
INSTITUT FÜR BETRIEBSSYSTEME UND RECHNERVERBUND
GRUPPE VERTEILTE SYSTEME

Prof. Dr. Stefan Fischer

Seminar Verteilte Systeme
Wintersemester 2002/03

SICHERHEIT IN AD-HOC-NETZEN

Betreuer: Stefan Schmidt

Alexander Schulz
Matr.-Nr.: 2538831
Franz-Liszt-Str. 40
38106 Braunschweig
Al.Schulz@tu-bs.de

Inhaltsverzeichnis

1. EINLEITUNG

Ad-hoc-Netze sind die neueste Art viele mobile Geräte miteinander kommunizieren zu lassen. Dabei wird ein Kommunikationsnetz über die Funk. bzw. IR-Schnittstelle zwischen jedem benachbarten Gerät, d.h. sich im Empfangs- und Sendebereich zu befinden, aufgebaut. Durch die Etablierung eines solchen Ad-hoc-Netzes kann jedes Gerät mit jedem anderen Gerät kommunizieren. Geräte die sich genau zwischen zwei sich nicht empfangenden Geräten befinden die miteinander kommunizieren möchten, agieren als Router. Damit ist eine Kommunikation zwischen nicht direkt benachbarten Geräten möglich. Der Hauptanwendungsbereich für solche Art von spontaner Vernetzung ist derzeit noch im militärischen Bereich zu suchen, obwohl verstärkt ein Trend zur kommerziellen Nutzung zu beobachten ist. Der Grund für diese Entwicklung sind die wohl einzigartigen Eigenschaften. Ad-hoc-Netze können mit relativ niedrigen Kosten schnell installiert werden. Viele dieser Eigenschaften bergen jedoch auch Sicherheitsrisiken in sich und müssen gegen Angriffe geschützt werden. Diese Ausarbeitung stellt Sicherheitsprobleme für Ad-hoc-Netze dar und versucht verschiedene Lösungen darzustellen. Die hier aufgezeigten Lösungen sind zum aktuellen Zeitpunkt zumeist noch nicht über den Prototypenstatus hinaus.

2. SICHERHEITSPROBLEME

Sicherheitsprobleme entstehen in Ad-hoc-Netzen durch dessen allgemeine Eigenheiten. Ad-hoc-Netze beruhen auf keiner externen Infrastruktur wie es Festnetze oder Wireless-LANs tun. Die Luft wird als geteiltes Medium genutzt und das Netz muss sich selbst organisieren. Die Selbstorganisation wird durch die Mobilität der Teilnehmer noch erschwert. So kann es durch Mobilität eines Teilnehmers zur Nichterreichbarkeit eines Teilnehmers oder sogar zur Partitionierung des Netzes kommen. Weiterhin existiert keine zertifizierte Autorität die Rechte im Netz gewährt bzw. verweigert. Durch den nur limitierten physikalischen Schutz jedes einzelnen Gerätes, ist blindes Vertrauen nicht möglich. Ein Gerät könnte gestohlen und manipuliert werden um Daten abzufangen, zu manipulieren oder den Datentransfer zu stören. Solche Attacken machen deutlich, dass Sicherheit einen sehr hohen Stellenwert bei Ad-hoc-Netzen bekommen muss. Als

grundlegende Attribute eines sicheren Ad-hoc-Netzes können genannt werden: Verfügbarkeit, Vertraulichkeit, Integrität, Authentizität und Nicht-Anfechtbarkeit.[1] Diese Attribute können je nach Anforderung des Netzes unterschiedlich stark gewichtet werden und führen zu unterschiedlichen Sicherheitsrichtlinien.

Im militärischen Anwendungsbereich erscheint dies außer Frage zu stehen. Keine militärische Einheit möchte im Ernstfall von anderen Einheiten abgeschnitten sein, oder gar seine Position über das Ad-hoc-Netz verraten. Militärische mobile Ad-hoc-Netze werden deshalb sehr strikte Anforderungen an Vertraulichkeit sowie Widerstandsfähigkeit gegen *denial of service*[2] Attacken haben.[3] Für einen Verband von Laptops in einer Konferenz ist die Positionsbestimmung möglicherweise als Sicherheitsrisiko irrelevant. Wünschenswert ist jedoch eine Kooperationsbereitschaft der teilnehmenden Geräte bspw. beim Routen von fremden Paketen. Ein *spontaeous networking* sollte möglich sein, bei dem Sicherheitsrichtlinien von den Vertrauensbeziehungen der teilnehmenden Personen abhängig ist. Dazu muss eine Plattform vorhanden sein, welche es Geräten ermöglicht sich gegenseitig zu authentifizieren, *zero configuration*.[4] Ein Netz von Sensoren hat wiederum ganz andere Sicherheitsanforderungen. Dies zeigt, dass Anforderungen an ein Ad-hoc-Netz sehr unterschiedlich ausfallen können. Dies bedeutet auch, dass mit ganz unterschiedlichen Attacken auf unterschiedliche Ad-hoc-Netze mit unterschiedlichen Zielen gerechnet werden muss. Grundlegend unterscheidet man zwischen Attacken gegen Basismechanismen wie Routing, und Attacken gegen Sicherheitsmechanismen wie Schlüsselverwaltung.

2.1 Attacken gegen Basismechanismen

Das wohl auffälligste Risiko bei wireless Geräten ist das Risiko gestohlen zu werden. Damit unterliegen die Geräte auch gleichfalls dem Risiko manipuliert und für nachfolgende Attacken missbraucht zu werden.

[1] vgl. [1], S. 1
[2] Ziel dieser Attacke ist es dem Opfer den Zugang zu bestimmten Diensten nicht zu ermöglichen
[3] vgl. [2], S. 1
[4] vgl. [1], S. 1

Ein ebenfalls sehr augenscheinliches Risiko existiert durch die Nutzung der Luft als geteiltes Kommunikationsmedium. Durch aktives Aussenden von Interferenzen kann die Kommunikation zwischen zwei Teilnehmer massiv gestört werden. Durch die Eigenschaft des geteilt seins, ist es Angreifern zudem möglich Datenübertragungen zwischen einzelnen mobilen Geräten mitzuhören.

Ein weiterer Angriffspunkt ist der Medienzugriff. Der MAC-layer steuert die Zugriffe auf die Luftschnittstelle zum Transport von Daten zu anderen Geräten. Diese Zugriffe geschehen in Vereinbarung mit allen davon betroffenen Geräten in der Umgebung. Es wird demnach eine Kooperation aller Geräte vorausgesetzt. Eine Attacke auf den Medienzugriff ist dadurch möglich, dass die dafür definierten Regel nicht eingehalten werden und dadurch die Belegung der Luftschnittstelle unfair ist. Dadurch kann die Netzwerkleistung stark beeinträchtigt werden.

Um Daten zu entfernten Geräten, die nicht direkt vom Sender erreichbar sind, sondern nur über andere Geräte, zu transportieren, muss anders als in konventionellen Netzen jedes Gerät auf der Route vom Sender zum Empfänger Daten weiterleiten (routen). Ein Unterbinden des punktuellen Weiterleiten von Datenpaketen bringt den Datenstrom an einer Stelle zum Erliegen und kann durch evtl. vorhandene andere Routen im Netz ausgeglichen werden. Ein egoistisches Gerät könnte sich aus verschiedenen Gründen weigern Daten für andere Geräte weiterzuleiten. Dies wird als *node selfishness* bezeichnet.[5] Ein Stören des gesamten Routingmechanismus durch das Aussenden falscher Routinginformationen durch einen Angreifer kann hingegen das komplette Netz stören, da Datenpakete falsche Wege zu Empfängern nehmen und womöglich nie ankommen.

Eine weitere Attacke bezieht sich auf das böswillige Auskundschaften der Umgebung um den Standpunkt bestimmter Geräte zu lokalisieren.[6]

2.2 Attacken gegen Sicherheitsmechanismen

[5] vgl. [2], S. 2
[6] vgl. [2], S. 2

Gegen Sicherheitsmechanismen gibt es eine Vielzahl von Attacken. Der böswillige Austausch von Public Keys, das Ausspionieren von Keys oder das Eindringen in Trust bzw. Key Server sind Beispiele für Angriffe. Diese Attacken sind nicht auf Ad-hoc-Netze beschränkt, sondern ein Problem aller Netzen in denen kryptografische Schlüssel vorgeschrieben sind. In Ad-hoc-Netzen müssen jedoch die Lösungen an die Besonderheiten wie Selbstorganisation, Teilnehmermobilität und Fehlen einer Zentralen Autorität angepasst werden.

3. SCHUTZ DER BASIS-MECHANISMEN

Basis-Mechanismen stellen die Grundlage für Kommunikation in Ad-hoc-Netzen dar. Sie ermöglichen bspw. den Zugriff verschiedener Geräte auf das Kommunikationsmedium, das Auffinden eines Gerätes um mit ihm zu kommunizieren sowie das Weiterleiten von Datenpaketen zu entfernten Geräten über bestimmte Geräte bis zum Zielgerät. Wie in 2.1. erläutert gibt es eine Vielzahl von Attacken gegen diese Mechanismen. Ein ungeschütztes Netz würde durch solche Attacken stark beeinträchtig oder sogar außer Funktion gesetzt werden. Im nachfolgenden werden Schutzmaßnahmen gegen diese Attacken beschrieben.

3.1 Hardwareschutz

Eine Schutzmaßnahme gegen Diebstahl von mobilen Geräten ist das Integrieren von kryptografischer Hardware in Form von Smart-Cards in das mobile Gerät. Diese Smart-Cards können schnell und einfach hinein und herausgenommen werden. Eine Kommunikation im Ad-hoc-Netz ist demnach nur mit dieser Smart-Card möglich. Diese Hardware personifiziert den Benutzer im Netz und erlaubt zudem den Gebrauch von anderen mobilen Geräten durch Einstecken der Card in ein anderes mobiles Gerät. Ein gutes bekanntes Beispiel hierfür ist die SIM-CARD der GSM-Netzbetreiber.[7]

Zum Schutz des Netzwerkmechanismus wäre eine ähnliche Lösung denkbar. Um Mithören von Datenpaketen und Attacken auf den Medienzugriff nicht zu ermöglichen

[7] vgl. [2], S. 2

bzw. zu erschweren wird die Software dafür in eine Smart-Card oder einen Sicherheitschip integriert, welcher im mobilen Gerät vorhanden sein muss um am Ad-hoc-Netz teilnehmen zu können.[8] Fehlt diese Card oder der Chip so ist eine Kommunikation mit anderen Geräten schon auf dieser Schicht unterbunden. Ein Mithören von Paketen ist ebenfalls nicht möglich. Ein Problem dieser Lösung ist jedoch das Upgraden der Software. Um einer böswilligen Manipulation zu entgehen muss das System feststellen können ob es sich um ein legitimes Upgrade oder eine Attacke handelt.

3.2 Routing

Ein großes Problem in Ad-hoc-Netzen ist dass unter 2.1 angesprochene *node selfishness*. Dies ist kein eigentlicher Angriff auf das Netz sondern ein Verweigern der Kooperation wodurch ein Ad-hoc-Netz erst funktionieren kann. Ein böswilliger Angreifer hingegen würde einem Weiterleiten von Paketen zustimmen, diese dann aber verwerfen. Um solche Geräte (Nodes) zu identifizieren und gegebenenfalls zu isolieren bzw. aus dem Netz auszuschließen existieren zwei Mechanismen. Ein *watchdog* versucht solche Geräte im Netz zu finden und ein *pathrater* versucht eine alternative Route, ohne über den böswilligen Node zu routen, für Datenpakete zu finden.[9] Jeder Node hat einen *watchdog* der im *promiscous mode* durch Abhören des Mediums erkennen kann ob ein Node die empfangene Datenpakete weiterleitet. Der *watchdog* verwaltet für jeden Nachbarnode einen Zähler über nicht weitergeleitete Pakete von ihm. Überschreitet dieser Zähler eine gewisse Schranke, so informiert er den Sender der Pakete die über diesen Node geleitet werden. Ein *pathrater* bewertet bekannte Knoten und errechnet dadurch eine Bewertung für eine gesamte Route. Er wählt stets die Route mit der besten Bewertung aus. Die Bewertung der Route entspricht der durchschnittlichen Bewertungen aller enthaltenen Nodes. Die Bewertung der Nodes startet bei 0,5. Durch fehlerfreie Weiterleitungen steigt die Bewertung für den Node alle 200ms um 0,01 bis auf maximal 0,8. Ein Verbindungsabbruch erniedrigt den Wert um 0,5 bis minimal 0. Ein vom *watchdog* gemeldeter Knoten wird auf –100 gesetzt.[10] Nun besteht die Frage was „das Netz“ mit dem egoistischen Node machen soll. Durch die

[8] vgl. [2], S. 2f
[9] vgl. [2], S. 3
[10] vgl [6]

eben genannten Mechanismen wird der fremde Datentransport um den Node herumgeführt aber der Node selbst ist immer noch in der Lage seine eigenen Daten zu versenden und für ihn bestimmte Daten zu empfangen. Dies würde bedeuten, dass er keinen Anreiz hat seinen Egoismus aufzugeben. Eine Möglichkeit *node selfishness* zu unterbinden wäre das Trennen aller Verbindungen der Nachbarknoten zu diesem böswilligen Node. Dieser ist dann vom Ad-hoc-Netz ausgeschlossen. Die Versuche einer erneuten Verbindung des ausgeschlossenen Nodes könnten nun in denial-of-service Attacken enden, oder er könnte durch Wechsels seine Standortes in einen Bereich des Netzes wo sein Verhalten nicht bekannt ist, erneut eine Verbindung mit dem Ad-hoc-Netz aufzunehmen.[11]

Ein weiteres Problem für Routing Protokolle sind alte bzw. falsche Routinginformationen. Alte Routinginformationen entstehen durch Mobilität der Teilnehmer, welches einen dynamischen Wechsel der Topology zur Folge hat. Damit werden alte Informationen auch zu falschen Informationen. Angreifer wie sie in 2.1 beschrieben wurden senden falsche Routinginformationen in das Netz um gezielt oder ungezielt Daten fehlzuleiten bzw. einen Datentransport zu unterbinden. Dabei muss zwischen externen Attacken und internen Attacken unterschieden werden. Externe Attacken kommen von nicht im Ad-hoc-Netz befindlichen Nodes. Eine Kommunikation mit diesen Nodes kann durch in 3.1 beschriebene Verfahren und Benutzung von kryptografischen Verfahren zur Übertragung von Routinginformationen unterbunden werden.[12] Interne Attacken kommen von im Ad-hoc-Netz etablierten Nodes. Solch böswilliges Verhalten zu entdecken ist sehr schwierig. Es kann meist nicht mit Sicherheit festgestellt werden ob es sich nur um Änderungen in der Topology, also alte Informationen handelt, oder ob es sich tatsächlich um mutwillig falsche Informationen handelt. Routingprotokolle müssen deshalb in der Lage sein alternative Routen zu bestimmen wenn sich eine Route als fehlerhaft herausstellt. Diese alternativen Routen umgehen dann böswillige Nodes.

Das Senden von Daten über mehrere Routen führt jedoch auch zu einer Mehrbelastung des Netzes. *Diversity coding* ist eine Möglichkeit Übertragungswiederholungen zu

[11] vgl. [2] S. 3
[12] vgl. [1] S. 3f

vermeiden.[13] Die Idee ist, redundante Informationen zur Feststellung von Fehlern und zur Fehlerkorrektur über alternative Routen zu übertragen. Dadurch sollte der Empfänger in der Lage sein, Übertragungsfehler auszugleichen.

Eine ganz andere Möglichkeit der Übertragung von Routinginformationen und damit auch der Möglichkeit der Manipulation dieser Informationen aus dem Weg zu gehen ist *geographic routing*.[14] Dies bedeutet, dass Pakete mit Hilfe der jeweiligen geografischen Postitionen der einzelnen Nodes weitergeleitet werden. Es brauchen keine kompletten Routen zum Transport festgelegt werden sondern es muss jeweils nur der nächste Knoten ermittelt werden. Dadurch gibt es auch keine Routingtabellen. *Geographic routing* setzt natürlich voraus, dass jeder Node seine geografische Position ermitteln kann und diese auch preisgeben möchte.

3.3 Kooperation

Ein weiteres Problem in Ad-hoc-Netzen stellt die Verfügbarkeit von Diensten dar. Aufgrund der Selbstorganisation sind Garantien nicht möglich. Es können aber zwei Verhaltensgrundsätze für Nodes festgelegt werden. Der erste Grundsatz lautet: Nodes müssen einen Anreiz zum Kooperieren bekommen. Die Kooperation betrifft hier auch die Weiterleitung von fremden Datenpaketen. Der zweite Grundsatz lautet: Nodes müssen davon abgehalten werden das Netz zu überladen.[15] Eine Umsetzung dieser beiden Grundsätze in Ad-hoc-Netzen führt eine virtuelle Währung „nuglet“ ein. Diese Währung wird von Nodes untereinander als Gegenleistung für Dienste ausgetauscht. Dadurch kann ein Node nur Dienste nutzen wenn er auch Dienste anbietet, also kooperiert.

[13] vgl. [1] S. 4
[14] vgl. [2] S. 3
[15] vgl. [2] S. 3f

Zwei spezielle Umsetzungen der *nuglet*-Währung zur Vergütung von Paketweiterleitungen sind das *packet purse m*odel und das *packet trade model*.[16] Beim *packet purse model* bezahlt der Sender für den Transport und beim *packet trade model* bezahlt der Empfänger.

Abbildung 1 illustriert das *Packet Purse Model*. Natürlich muss solch ein System auch gegen Attacken und falsche Vergütungen geschützt werden. Eine Möglichkeit ist eine hardwareseitige Implementierung dieses Vergütungsmodels. Dieses Hardwaremodul. würde die *nuglets* verwalten und auch die Verschlüsselung der *pocket purse* übernehmen.

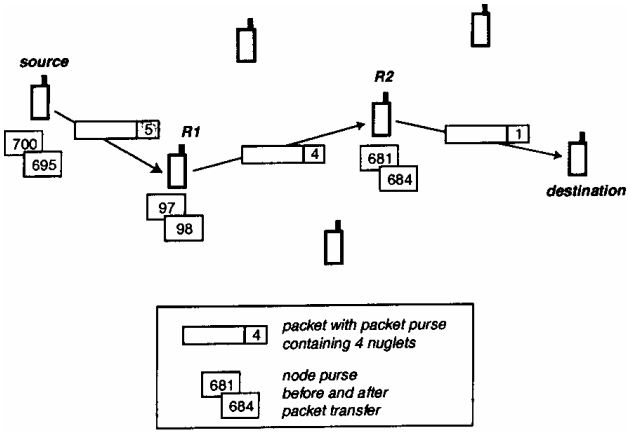

Abbildung 1: virtual currency: the nuglets- The source puts 5 nuglets in the packet purse. The first relaying node R1 charges 1 nuglet, wich is taken from the packet purse. The second relaying node R2 charges 3 nuglets. This higher greediness can be motivated, for instance, by the fact that R2 has to face a higher energy expenditure to relay the packet (typically if the distance to the next hop is higher). (Quelle: [2] S. 4)

4. SCHUTZ DER SICHERHEITS-MECHANISMEN

Dass Sicherheits-Mechanismen geschützt werden müssen steht hier wohl außer Frage. Erfolgreiche Angriffe, wie in 2.2 beschrieben, können dazu führen, dass ein böswilliger Node eine falsche Identität annimmt oder seine Identität wechselt. Für den Schutz von Datenübertragungen gibt es zwei grundlegende Verschlüsselungsverfahren. *Symmetric key cryptography* basiert auf einem geteilten Key mit dem Daten ver- und entschlüsselt werden. *Asymmetric key cryptography (4.1)* basiert auf einem persönlichen public/private key Paar wobei der *public key* veröffentlicht wird und damit für alle Nodes zugänglich ist. Mit Hilfe eines ausspionierten Secret-Keys ist es einem Angreifer möglich die für den Inhaber des Keys bestimmten Daten zu entschlüsseln und zu manipulieren. Ein wichtiger Teil des Sicherheit-Mechanismus ist die Etablierung von Keys. Dies kann auf zwei unterschiedliche Arten geschehen. Beim *key agreement (4.2)* wird ein geteilter Key von jedem Teilnehmer separat aus einer an alle Teilnehmer

[16] vgl. [2] S. 3f

gerichteten Information gewonnen. Beim *key transport* erzeugt ein Teilnehmer einen geheimen Wert und transportiert ihn auf sicherem Weg zu den Kommunikationspartnern.[17] Wie findet ein Node jetzt einen sicheren Weg zur Übertragung des Keys? Eine Lösung wird in 4.3 beschrieben. Eine andere Art der sicheren Übertragung des public keys durch eine geteilte Autorität im Ad-hoc-Netz stellt 4.4 dar.

4.1 Asymmetric key cryptography

Asymmetric key chryptography ist eine angemessene Sicherung, da es in Ad-hoc-Netzen keine feste Authorität und keine festen Server gibt. Bei diesem Kryptografieverfahren besitzt jeder Node ein public/private key Paar. Public keys werden an andere Nodes gesendet während private keys nur einem Node zugänglich sein sollten. Eine Nachricht an einen Empfänger wird mit des Empfängers public key verschlüsselt und an ihn gesandt. Der Empfänger mit dem zugehörigen private key ist als einziger in der Lage die Nachricht zu entschlüsseln. Eine weitere Möglichkeit des Systems besteht darin einen symmetrischen key mit Hilfe des public keys des Empfängers zu versenden. Damit wird *asymmetric key cryptography* nur zur Schlüsselübermittlung benutzt und die Kommunikation danach mit *symmetric key cryptography* verschlüsselt.[18] Geht man von einer Einbruchssicherheit des asymmetrischen Verfahrens aus, so kann auch die Übertragung des symmetrischen keys als sicher gelten und damit auch die darauffolgende Kommunikation mit dem symmetrischen key. Das Problem über die Richtigkeit des pulic keys bleibt jedoch bestehen. Aufgrund der fehlenden zertifizierenden Authorität ist es für den Empfänger eines public keys nicht ohne weiteres möglich zu entscheiden ob dieser vom erwünschten Kommunikationspartner gesendet wurde oder von einem Angreifer der als *relay*[19] fungiert. Eine Lösung für das Fehlen einer solchen festen Autorität sind emulierte Authoritäten wie sie im folgenden beschrieben werden. Drei grundlegende Ansätze zur Emulation in einem Ad-hoc-Netzwerk sind zu unterscheiden.[20] Der erste Anatz besteht im Emulieren einer normalen zertifizierenden Autorität die auf mehrere Nodes aufgeteilt ist. Dieser Ansatz wird in 4.4

[17] vgl. [2] S. 4
[18] vgl. [2] S. 4
[19] Node auf der Route vom Sender zum Empfänger, welcher Datenpakete weiterleitet (routet)
[20] vgl. [2] S. 4

beschrieben. Im zweiten Ansatz müssen sich Nodes gegenseitig authentifizieren um zu kommunizieren. Der dritte Ansatz basiert auf einer selbstorganisierenden *public key infrastuctur* und wird unter 4.3 beschrieben.

4.2 Key agreement

Dieses Verfahren ist besonders geeignet für Gruppenkommunikationen. Ein Beispiel dafür wäre eine Konferenz auf der die teilnehmenden Personen, für die Dauer der Konferenz, ihre mobilen Geräte miteinander kommunizieren lassen wollen. Dafür wird ein neues Passwort gewählt und an alle teilnehmenden Personen verteilt. Dies kann durchaus mündlich oder schriftlich geschehen. Die Übermittlung in diesem Fall kann nur als sicher gelten, wenn sich die teilnehmenden Personen vertrauen und nur diese Personen auch Zugang zu dem Passwort haben. Dieses Passwort wird nun von den mobilen Nodes benutzt um mittels *password-authenticated key exchange* einen stärkeren geteilten Key zu erhalten. Es werden von Nodes demnach nur symmetrische Keys zur Kommunikationsetablierung akzeptiert die auch das Passwort übermitteln. Es wird also ein starker Key aus einem möglicherweise sehr schwachen Key (Passwort) erzeugt.[21]

[21] vgl. [2] S. 5

4.3 Self-organized public-key infrastructure

Self-organized public-key infrastructure stellt eine Lösung für das in 4.1. angesprochene Problem der public key Übertragung vor. Der Empfänger eines angeforderten public keys, um mit dem Besitzer zu kommunizieren, hat keine Garantie, dass der empfangene key auch der erwünschte key ist. Ein böswilliger Node könnte die Anfrage nach dem key abgefangen haben und seinen eigenen public key zurückschicken anstelle der Weiterleitung der Anfrage. Daraufhin würde die Kommunikation mit dem böswilligen Node anstelle des erwünschten Empfängers stattfinden. Die bekannteste Methode dieses Problem zu lösen basiert auf public key Zertifikaten. Ein public key Zertifikat ist eine Datenstruktur in welcher ein public key, durch die digitale Signatur des Besitzers des Zertifikates, an eine Identität gebunden ist. Will jetzt Node A den authentischen public key des Nodes B haben, so wird eine Kette von public key Zertifikaten so gebildet, dass das erste Zertifikat der Kette von Node A, mithilfe eines A besitzenden und vertrauenden public keys, geprüft werden kann.[22] Jedes verbleibende Zertifikat kann mit dem public key des vorherigen Zertifikates aus der Kette überprüft werden. Das letzte Zertifikat enthält den public key des Nodes B. Diese Bildung einer Kette aus Zertifikaten wird durch Vertrauensbeziehungen zwischen den Nodes ermöglicht. Node A vertraut allen Eigentümern der Zertifikate in der Kette. Die Zertifikate selbst können aufgrund der dynamischen Veränderung der Topology nicht in festen Verzeichnissen oder festen Servern abgelegt werden sondern werden bei den einzelnen Teilnehmern (Nodes) gespeichert. Jeder Node enthält eine begrenzte Anzahl von Zertifikaten die von

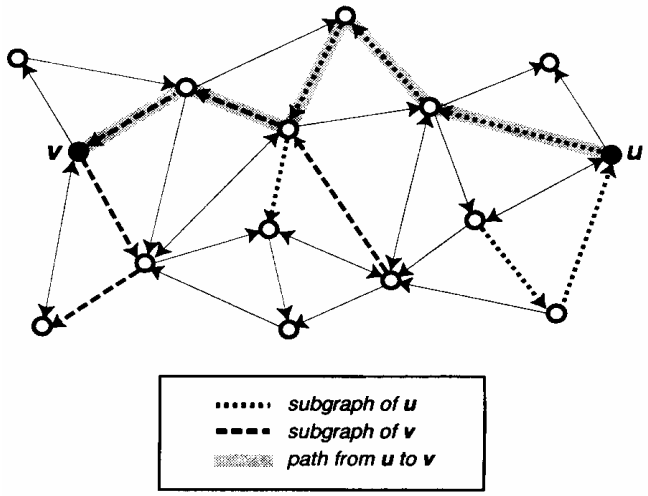

Abbildung 2: Merging subgraphs. When user u wants to verify the public key of user u, u and v merge their local certificate repositories, and u tries to find a certificate chain from u to v in the merged repository. In our trust graph model, local certificate repositories are represented by paths. Therefore, in the model, u and v merge their subgraphs, and u searches for a path from vertex v in the merged subgraph. (Quelle: [2] S. 6)

[22] vgl. [2] S. 5

verschiedenen Algorithmen ausgewählt werden.[23] Dadurch kann es passieren, dass ein Node alleine keine Kette von Zertifikaten zu einem speziellen anderen Nodes bilden kann. Die Lösung dafür ist ein Mischen der eigenen Zertifikate mit denen des zu erreichenden Nodes. Danach kann noch einmal nach einer Zertifikatkette gesucht werden.

4.4 Key management service nach Zhou & Haas

Um Daten und Routinginformationen durch ein Sicherheitskonzept zu schützen welches Kryptografie sowie digitale Signaturen benutzt, ist ein *key managment system* nötig. Dafür wurde eine public key Infrastruktur übernommen. Nodes die miteinander kommunizieren wollen benutzen diese um sich gegenseitig zu authentifizieren. Danach wird ein geteilter *secret session key* von beiden Nodes vereinbart und für die weitere Kommunikation benutzt.[24] In der *public key infrastructure* (PKI) hat jeder Node ein public/private key Paar und es gibt eine *trusted Certification Authority* (CA) für das Schlüsselmanagment. Nodes beziehen public keys also nicht mehr von unbekannten, möglicherweise nicht vertrauensvollen, Nodes sondern von der CA.[25] Die CA hat ebenfalls ein public/private key Paar um mit anderen Nodes zu kommunizieren. Die PKI ist dafür zuständig Zertifikate zu signieren die public keys enthalten. Da dieser Dienst permanent verfügbar sein muss, ist es notwendig, dass die CA auch permanent erreichbar ist. Nodes können ihre public keys bei der CA nach belieben aktualisieren. Nodes wollen zu jeder Zeit eine Möglichkeit zur Etablierung der Kommunikation mit anderen Nodes haben und benötigen dafür die CA. Durch die Dynamik in Ad-hoc-Netzen ist es nicht möglich eine einzige CA im Netz zu errichten. Der Node mit der CA könnte sich aus dem Netz herausbewegen oder durch andere Störungen auf der Luftschnittstelle vom Netz getrennt werden. Insbesondere ist eine CA böswilligen Attacken ganz allein ausgesetzt und könnte dadurch ebenfalls außer Funktion gesetzt werden. Ein erbeuteter CA private key würde zu einer Übernahme der Funktionen der CA führen und der Angreifer könnte durch Wiederrufen aller public keys oder durch das Signieren beliebiger Zertifikate, die Kommunikation zum Erliegen bringen. Mit Hilfe des private keys der CA kann sich der Angreifer ebenso gut als jeder beliebiger

[23] vgl. [2] S. 5f
[24] vgl. 4.1
[25] vgl. [1] S. 4

Node im Netz ausgeben. Deshalb wird hier das key managment auf mehrere Nodes verteilt. Jeder dieser Nodes (Server) enthält einen Teil des Keys der CA und speichert die public keys aller Nodes des Ad-hoc-Netzs.[26] Die Nodes zusammen bilden die CA. Jeder Server besitzt zudem auch ein eigenes public/private key Paar um sich mit anderen Servern verständigen zu können. Alle Nodes im Ad-hoc-Netz kennen den public key der CA und vertrauen jedem mit dem zugehörigen private key signierten Zertifikat. Nodes können an die CA *update* messages schicken um ihre public keys zu ändern und *query* Anfragen stellen um die public keys von anderen Nodes zu bekommen. Stellt nun ein Node A eine Anfrage an die CA um den public key eines Nodes B zu bekommen, so berechnet jeder Server eine partielle Signatur für seinen Teil des CA private keys. Diese Signaturteile werden zu einem Combiner[27] übermittelt. Nach einer Überprüfung wird dann eine komplette Signatur aus den Teilen generiert und versand. Dabei ist der Combiner in der Lage falsche Teile anhand des CA public keys zu erkennen und zurückweisen. Zur Erstellung der Signatur benötigt er nur eine vorher festgelegte Anzahl an Teilsignaturen. Liefert ein Server eine fehlerhafte oder gar keine Teilsignatur, so versucht er eine andere Menge an Teilsignaturen zur Generierung. Daraus folgt, dass bei einer Erwartung von t ausgefallenen oder „gehackten“ Servern, in einer bestimmten, der private key der CA in mindestens t+1 Teile aufgeteilt werden muss. Außerdem muss es im Netz mehrere Server geben, die den selben Teil des private keys besitzen und Teilsignaturen dafür generieren.

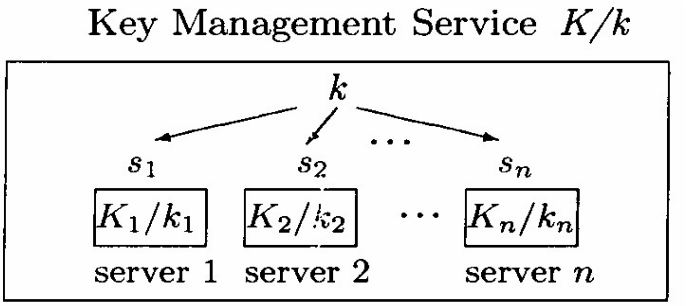

Abbildung 3: The configuration of a key managment service: the key manamgent service consists of n servers. The service, as whole, has a public/private key pair K/k. the public key K is known to all nodes in the network, whereas the private key k is divided into n shares s, one share for each server. Each server I also has a public/private key pair K/k and knows the public keys of al nodes. (Quelle: [1] S. 5)

Die CA ist weiterhin in der Lage den private key neu aufzuteilen, wobei die Aufteilung unabhängig von der alten Aufteilung ist.[28] Ein Aufteilung in mehr oder weniger Teile als zuvor ist ebenfalls möglich. Nach einer Neuaufteilung ist ein Mischen von alten und

[26] vgl. [1] S. 5
[27] Node der die Funktion des Zusammenfügens der Signaturteile übernimmt und die fertige Signatur verschickt; vgl.[1] S. 5f
[28] vgl. [1] S. 6f

neuen private key Teilen nicht möglich, was Angreifern eine zeitliche Barriere setzt. Ein stärkere Aufteilung des private keys ist zudem bei erhöhter Anzahl fehlerhafter Teilsignaturen oder anderen Attacken sinnvoll. Damit wird der private key auf mehr Server als zuvor verteilt. Bei Bedarf kann die Aufteilung auch wieder verringert werden.[29]

5. ZUSAMMENFASSUNG UND FAZIT

Ad-hoc-Netze sind ein recht neues Gebiet der Kommunikation im zivilen Bereich. In der Vergangenheit und auch in der Zukunft wird ein sehr großes Anwendungsgebiet im militärischen Bereich zu finden sein. Sicherheit in diesen Netzen ist sehr stark von dem Anwendungsgebiet abhängig. Ihr kann durchaus ein sehr hoher Stellenwert zugeschrieben werden. Sicherheitsprobleme vom Diebstahl über Medienzugriffsstörungen, Routingattacken, *node selfishness* bis hin zu Attacken gegen das Sicherheitssystem sind bekannt und wollen in einem sicheren Ad-hoc-Netz abgewert bzw. unterbunden sein. Attacken gegen das Sicherheitssystem sind besonders aufwendig zu verhindern. Gegen diese Attacken gibt es ein Reihe von Lösungen die oben dargestellt wurden. Viele dieser Lösungen sind jedoch noch im Prototypenstadium und funktionieren nur im Labor unter Testbedingungen. Wenn es aber auch im zivilen Bereich der Ad-hoc-Netze in der Luftschnittstelle voll wird, werden sich Hardwarefirmen auch um das Problem der Sicherheit und der Dienstgüte für diese Geräte und Netze kümmern.

[29] vgl. [1] S. 7

LITERATURVERZEICHNIS

[1] Lidong Zhou, Zygmunt J. Haas: Securing Ad Hoc Networks, Abstract, Cornell University, Ithaca

[2] Hubaux, Buttyan, Capkun: The Quest for Security in Mobile Ad Hoc Networks, Abstract, Institute for Computer Communications and Applications, Swiss Federal Institute of Technology - Lausanne, Switzerland

[3] Larry L. Peterson, Bruce S. Davie: Computernetze- Ein modernes Handbuch, dpunkt-verlag, Heidelberg, 2000

[4] Ralf Steinmetz:Multimedia-Technologie, Grundlagen, Komponenten und Systeme, 3. Auflage, Springer Verlag, Berlin, 2000

[5] Prof. Dr.-Ing Lars Wolf: Vorlesungsunterlagen der Vorlesung Multimedia-Systeme WS01/02, TU-Braunschweig, Institut für Betriebssysteme und Rechnerverbund

[6] David Wagner: Folien zum Seminarvortrag „Sicherheit in Ad-Hoc-Netzen“, Bonn, April 2002, http://web.informatik.uni-bonn.de/IV/martini/Lehre/Veranstaltungen/WS0102/Sem_Rechnernetze/Vortraege/David_Wagner.Folien.pdf