Sicherheit und Datenschutz in mobilen Netzwerken

Daniel Schall

Technische Universität Kaiserslautern
Dept. of Computer Science
67653 Kaiserslautern, GERMANY,
d_schall@informatik.uni-kl.de

Betreuer:
Jürgen Göres

Zusammenfassung Mobile Netzwerke sind aufgrund ihrer dynamischen, spontanen Natur zahlreichen Gefahren ausgesetzt und erfordern daher besondere Vorkehrungen, um Sicherheit und des Datenschutzes auch im mobilen Kontext gewährleisten zu können. Diese Arbeit stellt einige sicherheits- und datenschutzbezogenen Herausforderungen in mobilen Netzwerken vor und zeigt einige Techniken, die Lösungsansätze für die beschriebenen Problemstellungen anbieten.

Inhaltsverzeichnis

Sicherheit und Datenschutz in mobilen Netzwerken 1
Daniel Schall
1 Motivation 3
2 Sicherheit 4
2.1 Was bedeutet Sicherheit? 4
2.2 Szenarien 5
2.3 Verschlüsselung und deren Grenzen in mobilen Systemen 8
2.3.1 Symmetrische Verfahren 8
2.3.2 Asymmetrische Verfahren 8
2.3.3 Hybride Verfahren 9
2.3.4 Hierarchische Public-Key- Infrastrukturen 9
2.3.5 PGP und das Web-of-Trust 11
2.4 Erweiterte Public-Key-Infrastrukturen 13
2.4.1 Polynomial Secret Sharing 13
2.4.2 Threshold Cryptography 14
2.4.3 Erweiterte Threshold Cryptography 15
2.4.4 Fazit 18
3 Privatsphäre, Datenschutz und Privacy 19
3.1 Begriffsklärung 19
3.2 Warum schützen? 20
3.3 Gesetzliche Regelung 21
3.4 Probleme 22
3.5 Erweiterte Problematik in mobilen Szenarien 23
3.5.1 Sicheres Auffinden von Diensten 24
3.5.2 Drahtlose, spontane Kommunikation 25
3.6 Lösungsideen und -wege 25
3.6.1 k-Anonymität 25
3.6.2 Das MIX-Modell von Chaum 26
3.6.3 Proxies (JAP, Rewebber) 26
3.6.4 Crowds und The Onion Router 27
3.6.5 Spezifikation von Datenschutzanforderungen und -zusicherungen mittels P3P 29
3.6.6 Conditionally Anonymous Digital Signatures 29
3.6.7 Privacy Awareness System (pawS) 30
3.6.8 Data Confidentiality and Secure Computation 32
4 Fazit 35
5 Literatur 36

1 Motivation

In einer Welt, in der Computer immer kleiner, leistungsfähiger und damit auch mobiler und allgegenwärtig werden, erfreuen sich die Menschen an der wachsenden Vernetzung und sehen die Möglichkeiten, die ihnen diese Technologie eröffnet. Vielen werden beim Gedanken an ubiquitäres Computing die Vision sehen, immer erreichbar zu sein, alles quasi „auf dem Weg“ erledigen zu können, ständig die aktuellsten Informationen zu bekommen und nichts mehr zu vergessen, da sich der PDA alles merkt. Betrachten wir ein kleines Beispielszenario, das so vielleicht schon heute Realität geworden ist:

Es ist Montagmorgen und Max ist auf dem Weg zum Flughafen. Unterwegs synchronisiert er sein Smartphone noch schnell mit GMail, indem er sich in das offene Stadtnetz einloggt. Anschließend steigt er in die S-Bahn, bezahlt sein Ticket mittels Handy-Payment und setzt sich ans Fenster, um auf seinem Laptop den RSS-Feed einer Lokalzeitung zu lesen. Er hat leider vergessen, ihn zuhause herunterzuladen, aber glücklicherweise sind genügend viele Leute im Bus, dass ihre PDAs, Laptops und Smartphones spontan ein Ad-Hoc-Netzwerk bilden. Max klinkt sich ins Netz ein und lädt sich den Feed bequem als ausführbare Datei von einem Mitfahrer herunter.

Außerdem loggt sich Max ins Firmennetz ein und legt einige Neukunden in der Datenbank an. In der Firma angekommen, wundert sich Max, weshalb sein Laptop laut Virenscanner mit einem Trojaner infiziert wurde und seine neuen Kunden bereits Angebote der Konkurrenz erhalten...

Aus der Sicherheits- und Datenschutzperspektive betrachtet, verletzt Max in dieser Situation grundlegende Vorsichtsprinzipien. Es ist erkennbar, wie wichtig es ist, sich über die Sicherheit und den Schutz der persönlichen Daten in allgegenwärtig verfügbaren Computernetzen Gedanken zu machen. Im Folgenden werden zuerst einige grundlegende Probleme aufgezeigt, die sich in mobilen Umgebungen, betreffend der Sicherheit der Benutzer und deren Daten, ergeben, um anschließend einige neue Ansätze vorzustellen, wie diese Probleme zumindest teilweise gelöst werden können. Im Weiteren werden dann den Datenschutz betreffenden Problem erörtert und neue Lösungsverfahren vorgestellt werden.

Zuerst müssen, um ein einheitliches Verständniss zu erreichen, einige Begriffe definiert werden. Im folgenden Text werden die Begriffe „mobiler Agent“, „Knoten“ und „PDA“ synonym verwendet und bezeichnen mobile, tragbare Computer, die in der Lage sind, über Ad-Hoc-Netzwerke zu kommunizieren und eine vergleichbare Leistung wie ein PDA besitzen.

Mit *Ad-Hoc-Netzwerken* werden hier Basistechnologien wie Bluetooth, WLAN oder ZigBee bezeichnet, die spontan zwischen mehreren Knoten etabliert werden können und zusätzlic einen Routingmechanismus implementieren. Traditionell erfolgt die Kommunikation über das TCP/IP-Protokoll. In dieser Arbeit wird davon ausgegangen, dass Basisdienste wie das angesprochene Routing und Netzwerkmanagement auf jedem Netzwerkteilnehmer verfügbar sind, daher wird auf diese Dienste nicht speziell eingegangen.

2 Sicherheit

Der durchschnittliche mobile Computer wird immer leistungsfähiger und verfügt über mehr Rechenleistung, so dass System- oder Architekturunterschiede zwischen PDA und Desktop-Computer entfallen. Standardsoftware, die ursprünglich für den Desktop-PC-Markt entwickelt wurde, kann auf einer breiten Masse von mobilen Geräten eingesetzt werden. Dadurch vergrößern sich auch die möglichen Angriffsvektoren um aus dem PC-Bereich altbekannte Bedrohungen, wie etwa Viren, Würmer oder trojanische Pferde durch die weite Verbreitung standardisierter Hard- und Softwareplattformen. Gleichzeitig entstehen durch neue Verwendungsmöglichkeiten und verbesserte Eigenschaften wie längere Akkulaufzeiten und schnellere Kommunikationstechnologien neue, bisher unbekannte Bedrohungen [1]. Doch bevor einige Bedrohungsszenarien skizziert werden, soll die Bedeutung des Begriffs Sicherheit, mit besonderer Beachtung der in einem mobilen Kontext relevanten Aspekte, vorgestellt werden.

2.1 Was bedeutet Sicherheit?

Die grundlegenden Sicherheitsaspekte sind nach [2] im Vergleich zu drahtgebundenen Netzen auch in mobilen Netzwerken gleich geblieben: Als erstes Sicherheitsmerkmal ist die *Verfügbarkeit* zu nennen. Diese beschreibt das Maß für die Fähigkeit von Diensten auf Anfragen mobiler Agenten erwartungsgemäß zu reagieren. Da Angriffe (speziell Denial-of-Service) gegen Dienste von überall aus dem mobilen Netz kommen können, ist die Sicherstellung der Verfügbarkeit ein wichtiges Anliegen. Im weiteren Sinne zählt zu Verfügbarkeit auch die Robustheit des Netzes gegenüber Angriffen.

Weiter ist *Vertraulichkeit* ein wichtiges Merkmal sicherer Kommunikation. Darunter versteht man, dass Informationen nur vom beabsichtigten Empfänger gelesen werden dürfen und für nicht autorisierte Knoten kein Zugriff auf den Inhalt der Nachricht besteht. Da in Ad-Hoc-Netzwerken Routing über jeden Knoten möglich ist und jeder Teilnehmer in Reichweite die Funkstrecke abhören kann, müssen besondere Vorkehrungen wie Verschlüsselung getroffen werden. Im Gegensatz zu drahtgebundenen Netzwerken benötigt ein potentieller Angreifer keinen physischen Zugang zur Infrastruktur mehr, es genügt, sich in der Nähe des Netzwerkes aufzuhalten, um die Kommunikation belauschen zu können.

Die *Integrität* der Daten ist ein weiteres zentrales Sicherheitsmerkmal. Dieses soll verhindern, dass Daten nicht von einem Dritten, sogenannten *Man-in-the-Middle*, abgefangen, verändert oder eingeschleust werden können wie in Abb. 1 auf Seite 5 verdeutlicht. Alice und Bob kommunizieren in Fall (A) direkt miteinander, in Fall (B) hat sich Mallory dazwischengeschaltet, indem er die Daten von beiden Seiten über sich umlenkt, ohne dass die beiden etwas davon bemerken. Es ist erheblich einfacher, drahtlose Kommunikation zu belauschen und Störsignale ins Netz einzuschleusen, als dies bei drahtgebundenen Netzen der Fall ist.

Das vierte Sicherheitsmerkmal ist die *Authentifizierung* von Knoten, also die sichere Identifikation der Kommunikationspartners. Diese Maßnahme ist dringend erforderlich, um beispielsweise vor Identitätsdiebstahl zu schützen. Außer-

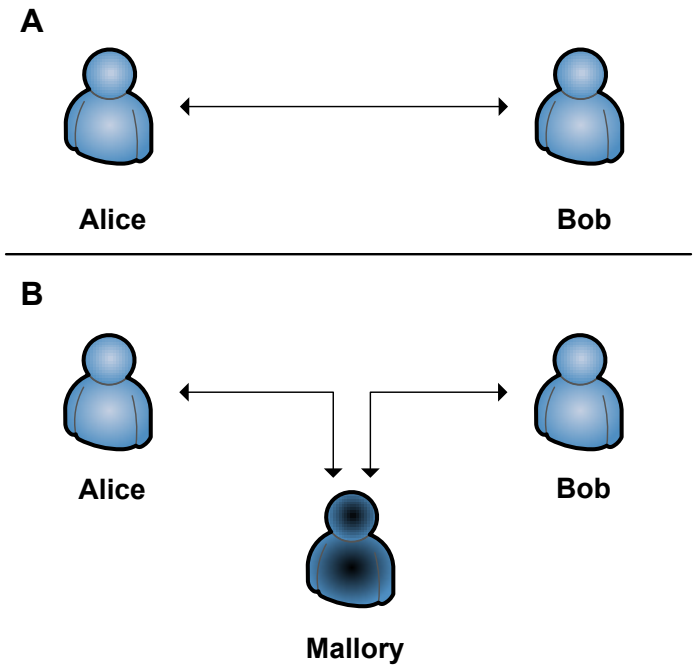

Abbildung 1. Man-in-the-Middle-Angriff

dem ermöglicht nur eine sichere Identifikation den Aufbau von Vertrauen, da ohne Authentifizierung keine Aussagen darüber getroffen werden können, ob der Kommunikationspartner tatsächlich der ist, für den er sich ausgibt.

Die unwiderlegbare *Nachweisbarkeit*, dass eine Nachricht von einem bestimmten Knoten stammt, ist vor Allem für kostenpflichtige Dienste und deren Abrechnung in mobilen Umgebungen interessant und kann außerdem zum Nachweis für kompromittierte Knoten und deren Ausschluss aus dem Netzwerk dienen. Daher stellt die Nachweisbarkeit einen weiteren, wichtigen Sicherheitsaspekt dar.

Sicherzustellen, dass der Zugriff auf Objekte (Informationen, Dienste, etc.) nur berechtigten Agenten offensteht, ist das Ziel des letzten der fünf Sicherheitsaspekte, der *Zugriffskontrolle*. Außerdem wird diese Kontrolle im drahtlosen Netz benötigt, um Agenten anhand von Regeln Zugang zum Netz zu gewähren oder zu verbieten.

2.2 Szenarien

Nach [1] können mögliche Angriffe auf mobile Agenten und deren Kommunikation in folgende Kategorien unterteilt werden:

– *A Angreifer mit Zugang zum mobilen Agenten*

Hat sich ein Angreifer Zugang zu einem mobilen Agenten verschafft, sei es durch physischen Zugriff auf das Gerät oder durch entfernten Zugang über das Netzwerk, sind zum einen die dort gespeicherten Daten gefährdet, andererseits kann der Angreifer nun mit Hilfe des gekaperten Agenten weitere Angriffe auf die Infrastruktur unter der Identität des ursprünglichen Besitzers durchführen. Besonders gefährdet sind hier mobile Agenten, die vom Besitzer sowohl dienstlich, als auch privat benutzt werden, da diese sensible, unternehmensinterne Daten gespeichert haben, aufgrund von Bequemlichkeit aber oft ohne Passwort oder Verschlüsselung verwendet werden.

1. Datendiebstahl
   Bei Zugang zur Gerätehardware durch einen Angreifer besteht die Gefahr des Diebstahls einer Vielzahl von Daten. Zum einen können alle gespeicherten Dateien, Kontakte, E-Mails etc. des Besitzers ausgelesen werden, sofern diese nicht verschlüsselt gespeichert sind.
   Andererseits kann aber auch das Gerät selbst viel über seinen Besitzer verraten. Abnutzungsspuren auf dem Touchscreen können dem Angreifer beispielsweise gute Hinweise auf Passwörter geben. Da mobile Geräte aufgrund relativ geringer Prozessorleistung oftmals nur mit einfachen Verschlüsselungsverfahren arbeiten, können die Daten meist vergleichsweise leicht entschlüsselt werden, indem ein Brute-Force-Angriff auf den Schlüssel auf einem um Größenordnungen schnelleren Desktop-PC gestartet wird.

2. Manipulation
   Nicht nur der einfache Diebstahl bereits vorhandener Daten auf dem mobilen Gerät stellt eine Gefahr dar, die Manipulation der Hard- und Software ist ferner geeignet, den Benutzer unerkannt abzuhören, während er sein Gerät benutzt. Auf Softwareebene existieren zahlreiche Backdoor-Programme, die den Benutzer unbemerkt ausspionieren können und beispielsweise Tastatureingaben oder Positionsdaten aufzeichnen und an einen Angreifer weiterversenden. Denkbar wäre der Einsatz von zusätzlichem Speicher zum Protokollieren aller Eingaben oder ein kleiner Funkchip, um die erbeuteten Daten direkt weitergeben zu können, ohne auf bestehende Gerätehardware zurückgreifen zu müssen.

3. Verwendung des Agenten zum Angriff auf die Infrastruktur
   Ein erbeuteter oder manipulierter Agent kann dazu verwendet werden, weitere Agenten auszuspionieren, indem er beispielsweise die Kommunikation in einem Ad-Hoc-Netzwerk über sich umleiten lässt oder sich als legitimer Client ausgibt, um an vertrauliche Daten zu gelangen. Ein Angreifer kann auch die Vertrauensstellung des gekaperten Agenten dazu missbrauchen, schädliche Programme auf anderen Clients zu verteilen, so dass der Angrei-

fer in Besitz weiterer Agenten und Daten gerät.

B Angreifer hat Zugriff auf Kommunikationswege der Agenten
Ein Angreifer, der Zugriff auf Kommunikationswege hat, ist potentiell in der Lage, die Kommunikation von Agenten zu belauschen und eventuell neue Daten einzuschleusen. Dadurch ergeben sich zwei grundsätzliche Angriffsmöglichkeiten.

1. Angriff gegen Kommunikation
   Einmal kann der Angreifer die Kommunikation von Agenten belauschen und als sogenannter *Man-In-The-Middle* (vgl. Abb. 1) manipulieren, indem er sich zwischen die kommunizierenden Stationen schaltet und sich für beide Seiten als der beabsichtigte Kommunikationspartner ausgibt. So kann ein Angreifer Pakete einfügt oder versendetet Daten verändern, um Agenten zu manipulieren. Aufgrund von Protokollfehlern kann der Angreifer möglicherweise auch Pufferüberläufe durch manipulierte Pakete verursachen und so Schadsoftware auf den Clients ausführen. Hat ein Angreifer Zugriff auf die Kommunikation, kann er außerdem Denial-Of-Service-Angriffe gegen einzelne Agenten durchführen und diese vom Netzwerk abschneiden, sowie Passwörter und andere sensible Daten abhören, die unverschlüsselt übertragen werden.

2. Angriff gegen Infrastruktur
   Ein Angreifer kann aus dem Netz heraus die Infrastruktur der Kommunikation zwischen Clients angreifen. Indem er beispielsweise DoS-Angriffe, wie im vorigen Szenario beschrieben, gegen Verteilerknoten der Infrastruktur durchführt, kann er große Teile des Netzes stören. Hat ein Angreifer einen Clientzugriffspunkt erfolgreich ausgeschaltet, kann er sich selbst als dieser ausgeben, so dass Clients fortan Daten über den Angreifer schicken. Durch manipulierte Verwaltungsinformationen wird dem Angreifer das Ausspähen der Clients ermöglicht. Da solche Verteiler- oder Zugriffspunkte oftmals an ein anderes, möglicherweise internes Netzwerk mit niedrigeren Sicherheitsrichtlinien, angeschlossen sind, könnte ein Angreifer versuchen, über diese Punkte Zugriff auf weitere Dienste innerhalb eines Unternehmens zu bekommen.

Es wird ersichtlich, dass ein mobiler Agent umfangreichen Bedrohungen ausgesetzt ist und Maßnahmen erforderlich sind, um Angreifern den Zugriff auf die Daten und Kommunikationswege der Agenten zu verwehren. Im Folgenden werden einige Konzepte vorgestellt und diese einer kritischen Analyse unterzogen. Zuerst werden etablierte Sicherheitskonzepte beschrieben und deren Anwendbarkeit auf mobile Szenarien untersucht.

2.3 Verschlüsselung und deren Grenzen in mobilen Systemen

Beim Einsatz von Verschlüsselungsverfahren in mobiler Kommunikation müssen die speziellen Eigenschaften von mobilen Agenten beachtet werden, wie die (noch) geringe Systemleistung im Vergleich zu Desktop-Computern, die begrenzte Akkuleistung, sowie eingeschränkte Kommunikationsmöglichkeiten [2].

2.3.1 Symmetrische Verfahren *Symmetrische Verfahren* benutzen zum Verschlüsseln einer Nachricht den gleichen Schlüssel wie zum Entschlüsseln. Der gemeinsame Schlüssel muss daher zwischen Sender und Empfänger der Nachricht über einen *sicheren Kanal*, der möglichst schwer abgehört werden kann, ausgetauscht werden, bevor eine Verschlüsselung mit einem symmetrischen Verfahren stattfinden kann. Ein solcher sicherer Kanal kann zum Beispiel eine bestehende, verschlüsselte Verbindung sein, allerdings müsste für diese im Voraus schon einmal ein Schlüssel ausgetauscht worden sein. Daher wird der Schlüssel besser persönlich übergeben, oder mittels dem *Diffie-Hellman-Verfahren* übertragen, das es zwei Teilnehmern erlaubt, einen symmetrischen Schlüssel zu vereinbaren, ohne dass ein Angreifer mithören kann. Trotzdem ist ein Man-In-The-Middle-Angriff möglich, da ein Knoten M, der sich zwischen die Kommunikation von zwei Knoten A und B schaltet, den Diffie-Hellman-Schlüsselaustausch mit beiden Seiten durchführen kann und so eine gesicherte Verbindung zwischen A und M und eine weitere zwischen M und B herstellt, ohne dass die Knoten A oder B etwas davon bemerken würden. Da mit Anzahl der Teilnehmer im Kommunikationsnetz die Anzahl der zu verwaltenden Schlüssel quadratisch ansteigt (jeder muss den Schlüssel jedes Kommunikationspartners vereinbaren und speichern), ist der Verwaltungsaufwand in diesem Verfahren recht hoch.

2.3.2 Asymmetrische Verfahren Im Gegensatz zu den beschriebenen symmetrischen Verfahren existieren bei *asymmetrischen Verfahren* zwei Schlüssel für jeden Knoten. Einer wird als privater Schlüssel (Private Key) bezeichnet und ist nur dem jeweiligen Knoten bekannt, der andere, sogenannte öffentliche Schlüssel (Public Key), wird allgemein bekannt gemacht und ist für jeden Knoten frei zugänglich. Verschlüsselung kann wahlweise mit einem der beiden Schlüssel erfolgen, das Entschlüsseln der Nachricht ist dann nur mit Hilfe des anderen Schlüssels möglich. So kann eine Nachricht die an einen Knoten X geht, mit dem öffentlichen Schlüssel PK_X verschlüsselt werden und das Entschlüsseln ist nur noch mit dem privaten Schlüssel SK_X möglich. Zusätzlich ermöglichen asymmetrische Verfahren die Integritätssicherung einer Nachricht, indem vom Absender ein Hash-Wert H über die Originalnachricht berechnet wird und diesen Wert mit seinem privaten Schlüssel verschlüsselt $SK_X(H)$ und an die Nachricht anhängt. Dieser angehängte, verschlüsselte Hash-Wert wird auch *Signatur* genannt. Der Empfänger Y der Nachricht kann den verschlüsselten Hash-Wert $SK_X(H)$ nun mit dem öffentlichen Schlüssel PK_X entschlüsseln $PK_X(SK_X(H)) = H$ und den Wert H dann mit dem selbst berechneten Hash-Wert der Nachricht H' vergleichen. Sind beide identisch, kann der Empfänger sicher sein, dass die Nachricht

auf ihrem Weg nicht verändert worden ist, da nur der Knoten X in der Lage ist, den Hash-Wert mit seinem privaten Schlüssel SK_X zu verschlüsseln [3].

Der Aufwand bei asymmetrischen Verfahren, gemessen an den zu verteilenden Schlüsseln, steigt nur linear mit der Anzahl der im Netz vorhandenen Knoten, daher skaliert dieser Ansatz besser als symmetrische Verfahren.

2.3.3 Hybride Verfahren Leider sind asymmetrische Verfahren im Vergleich zu symmetrischen ca. um den Faktor 1000 langsamer, daher wird häufig *hybride Verschlüsselung*, eine Kombination aus symmetrischer und asymmetrischer Verschlüsselung eingesetzt, wie im Folgenden beschrieben.

Eine Nachricht N wird mit einem neu erstellten, symmetrischen Schlüssel K verschlüsselt $K(N)$ und der Schlüssel K wird dann mit dem öffentlichen Schlüssel PK_X des Empfängers X asymmetrisch verschlüsselt $PK_X(K)$ und an die Nachricht angehängt. So kann der Empfänger den symmetrischen Schlüssel mit Hilfe seines privaten Schlüssels SK_X wieder entschlüsseln $SK_X(PK_X(K)) = K$ und den symmetrischen Schlüssel K dann zum Entschlüsseln der eigentlichen Nachricht verwenden. Geht eine Nachricht an mehrere Empfänger $X_{1..n}$, ist es auch möglich, den symmetrischen Schlüssel für jeden einzelnen Empfänger $X_i, i \in \{1..n\}$ eigens mit dessen Public Key zu verschlüsseln und an die Nachricht anzuhängen [4]. Ein Problem beim Einsatz von diesen Verfahren besteht darin, dass zwar die Authentizität von Nachrichten gewährleistet wird, da nur der Absender in der Lage ist, diese mit seinem Private Key zu signieren, der Absender der Nachricht jedoch nicht sicher identifiziert wird. Jeder Knoten ist in der Lage, sich ein eigenes Public/Private-Key-Paar auszustellen und sich als beliebiger Kommunikationspartner ausgeben, ohne dass ein Gegenüber diese Angaben überprüfen kann, da es keine eindeutige Zuordnung zwischen Schlüsselpaar und Identitäten gibt.

Im Folgenden werden nun Ansätze präsentiert, die die vorgestellten (a-)symmetrischen Verschlüsselungsverfahren einsetzen und zusätzlich die Identifikation der Teilnehmer über eine Vertrauensinstanz ermöglichen.

2.3.4 Hierarchische Public-Key- Infrastrukturen *Public-Key-Infrastrukturen* wie der X.509-Standard der internationalen Fernmeldeunion (ITU) [5] setzen eine zentrale Zertifizierungsstelle (CA, Certificate Authority) voraus. Dieser Stelle müssen alle an der Zertifizierungshierarchie teilnehmenden Personen und Computer vertrauen, da sie die Gültigkeit anderer Zertifikate beglaubigt. Hierzu signiert die Stelle vertrauenswürdige Zertifikate anderer Personen oder untergeordneter Zertifizierungsstellen. Daher wird die oberste Stelle auch Root-CA oder *Stammzertifizierungsstelle* genannt. Untergeordnete Zertifizierungsstellen können wiederum Zertifikate signieren und über eine Verkettung von Vertrauensstellungen kann so die Echtheit eines von einer untergeordneten Stelle signierten Zertifikats bis zur Stammzertifizierungsstelle evaluiert werden.

Klassische Zertifikate wie der X.509-Standard bestehen aus einem Public-Key, einer Seriennummer und den Verfallsdaten des Zertifikats, oftmals enthalten

sie außerdem noch mehr Informationen über den Antragsteller (wie Identifikationsmerkmale) und den Zweck des Zertifikats (wie z.B. Client-/Serverauthentifizierung, Signatur, Verschlüsselung). Mit Hilfe der Identifikationsmerkmale, wie beispielsweise Name, E-Mail- oder Serveradresse ist ein Knoten nun in der Lage, die Identität des Kommunikationspartner zweifelsfrei feststellen zu können, da die zertifizierende Stelle für diese Angaben bürgt.

Die Zertifizierungsstelle beglaubigt die Gültigkeit des Zertifikats, indem sie dieses mit ihrem privaten Schlüssel signiert. So können andere Teilnehmer mit Hilfe des Zertifizierungsstellenzertifikats, also dem öffentlichen Schlüssel der CA, die Echtheit des Clientzertifikats überprüfen.

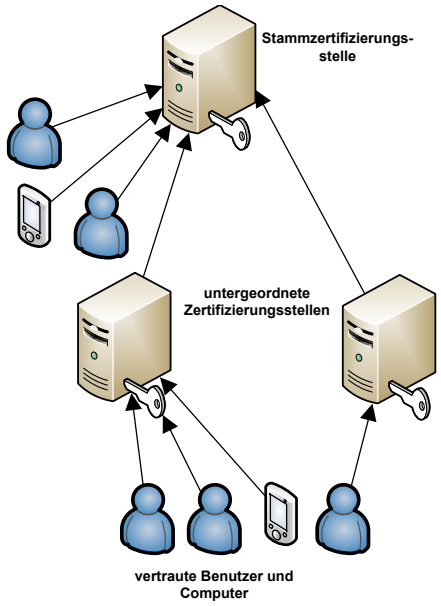

Abbildung 2. Public-Key-Infrastruktur

Das Wurzelzertifikat der Hierarchie ist mit dem eigenen privaten Schlüssel signiert, daher muss diesem explizit vertraut werden, da es keinen automatischen Algorithmus gibt, um dieses selbstsignierte Zertifikat auf Vertrauenswürdigkeit zu prüfen. Wohl kann durch die Signatur eine Manipulation durch Dritte ausgeschlossen werden. Neben dem Verfallsdatum des Zertifikats existiert ein weiterer Mechanismus, um Clientzertifikate bereits vorab für ungültig erklären zu können, etwa wenn sich der Zertifikatsinhaber nicht regelkonform verhält. Diese Liste mit gesperrten Zertifikaten, die von der ursprünglich ausstellenden Stelle bereitgehalten wird, heißt *Certficate Revocation List* (CRL) und kann beispielsweise über das Web von jedem abgefragt werden.

Da dieses Modell eine strenge Hierarchie (siehe Abb. 2 auf Seite 10) aufbaut, ist die Sicherheit der Stammzertifizierungsstelle maßgeblich für die Sicherheit der gesamten Zertifizierungshierarchie. Sollte es einem Angreifer gelingen, diese Stelle unter seine Kontrolle zu bringen, kann er alle bisher ausgestellten Zertifikate sperren lassen und außerdem neue ausstellen. Die auf dieser Hierarchie basierenden Vertrauensstellungen wären dann ungültig. Um dieses Risiko zu mindern existiert die Möglichkeit einer sogenannten *Cross-Zertifizierung*, bei der zwei Stammzertifizierungsstellen ihre Wurzelzertifikate gegenseitig signieren. Beide Hierarchien bekunden sich so gegenseitiges Vertrauen. Diese Cross-Zertifizierung verursacht jedoch mit der Anzahl der Stammzertifizierungsstellen einen quadratisch wachsenden Aufwand, da jede Stelle jedes Wurzelzertifikat der vertrauten Stellen signieren muss. Trotzdem wird das Risiko eines Single-Point-of-Failure abgeschwächt, da nun mehrere, gleichberechtigte Stammzertifizierungsstellen existieren. Kompromittiert ein Angreifer eine dieser Stellen, verfallen nichtsdestotrotz alle Client-Zertifikate, die von dort ausgestellt wurden, allerdings können alle anderen Stammzertifizierungsstellen nun ihre Kreuzzertifizierung widerrufen, indem sie die Sperrung in ihre CRLs eintragen, so dass sich der Schaden durch den Angriff auf eine einzige Stammzertifizierungsstelle beschränkt [6]. Hierarchische PKIs sind für drahtgebundene Kommunikation ein etabliertes Verfahren, um Vertrauen und Sicherheit zwischen unbekannten Benutzern herzustellen. Leider ist dieses Konzept für drahtlose, spontane Netzwerke nur schwer anwendbar, da die benötigte, zentrale Vertrauensinstanz oftmals nicht existiert, bzw. aufgrund von dynamischen Änderungen der Netzstruktur nicht immer von allen Netzteilnehmern erreichbar ist. Da eine zentrale Instanz außerdem immense Skalierungsprobleme mit sich bringt, müssen andere Lösungswege gefunden werden. Im folgenden wird ein Ansatz vorgestellt, der versucht, dieses Problem zu lösen.

2.3.5 PGP und das Web-of-Trust Das Web-of-Trust [7] verzichtet auf eine zentrale Instanz wie in Public-Key-Infrastrukturen, die Vertrauen zwischen den beteiligten Nutzern verwaltet und diese zertifiziert. An die Stelle der zentralen Lösung tritt ein soziales Netzwerk aus gegenseitiger Vertrauensbezeugung, in dem vertraute Parteien ihre Zertifikate gegenseitig signieren und so anderen ihre Vertrauensbeziehung bekunden. Zwei Nutzer (oder mobile Agenten), die vorher noch keinen direkten Kontakt miteinander hatten, werden bei der ersten Kontaktaufnahme versuchen, eine Kette von Vertrauensbeziehungen zur anderen Partei herzustellen. Dies lässt sich mit dem Satz „Vertraue jedem, dem jemand vertraut, dem du vertraust“ gut zusammenfassen. So entstehen im Laufe der Zeit große Netzwerke aus gegenseitigen Vertrauensbeziehungen (Chains of Trust) [6]. In Abb. 3 auf Seite 12 ist schematisch dargestellt, wie Vertrauensbeziehungen aufgebaut sind.

PGP (Pretty Good Privacy, entwickelt von Philip R. Zimmermann) implementiert die Idee des Web-Of-Trust. Jeder Teilnehmer erstellt sich ein Schlüsselpaar, bestehend aus privatem und korrespondierendem öffentlichem Schlüssel. Der öffentliche Schlüssel zusammen mit der E-Mailadresse der Person stellt

gleichzeitig das Zertifikat des Teilnehmers dar und wird von anderen Personen, die dieser vertrauen, mit ihrem privaten Schlüssel signiert. So entstehen nach und nach die erwähnten Vertrauensketten. PGP kann auch dahingehend angepasst werden, dass mehr als eine Person jemandem vertrauen muss, bevor dieser anderen Person ebenfalls vertraut wird.

Eine große Gefahr von PGP ergibt sich aus der Beschaffenheit der Zertifikate. Da dort die E-Mailadressen der Personen gespeichert und publiziert werden, ist Anonymität nicht zu realisieren. Durch die Bekanntmachung der Vertrauensstellungen im Netz ist eine Analyse sozialer Strukturen für Eindringlinge sehr einfach. Oftmals entstehen PGP-Communities, die alle untereinander sehr viel Kontakt pflegen und daher alle ihre Schlüssel gegenseitig signiert haben. Solche Communities sind sehr einfach ausfindig zu machen und beispielsweise zu Marketinganalysen zu missbrauchen. Die Probleme, die aus dieser ungeschützten Offenlegung von privaten Daten entstehen, werden in Kapitel 3 eingehend erläutert.

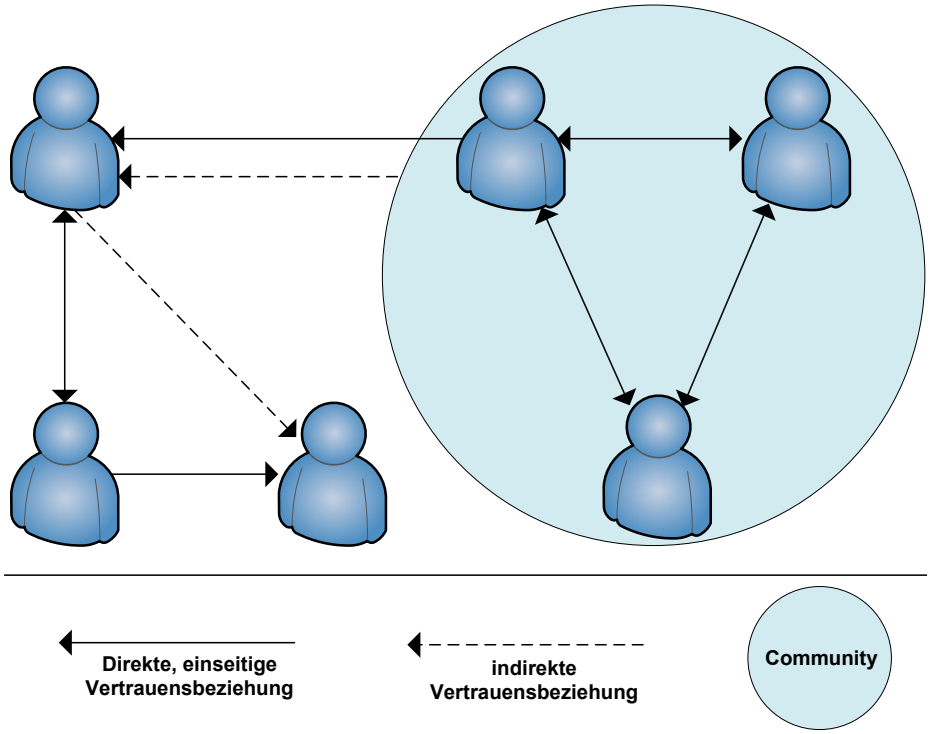

Abbildung 3. Web-of-Trust

Eine Analyse der Nachteile des von PGP aufgebauten Systems zeigt schnell, dass dieser Ansatz für große, dynamische Netzwerke nicht funktioniert, in der Knoten spontan ein- und wieder austreten. Zum einen erfordert der Aufbau von

Vertrauensketten einige Zeit und macht das Etablieren mehrerer vertrauter Knoten im Netz nötig. Da ein mobiles Netz jedoch schnellen Veränderungen unterliegt, ist es unmöglich, diese Vertrauensstellungen schnell genug herzustellen um damit effizient arbeiten zu können. Außerdem zeichnet sich ein Ad-Hoc-Netzwerk dadurch aus, dass zwei beliebige Knoten theoretisch in die Situation kommen können, miteinander kommunizieren zu müssen oder Pakete untereinander zu routen. Daher muss jeder Knoten die Vertrauensstellung aller anderen Knoten protokollieren und in einer Liste von der Größe der Knoten im Netz führen, welchen Knoten er vertraut. Diese Einschränkungen machen es unmöglich, PGP auf große Ad-Hoc-Netzwerke zu skalieren.

2.4 Erweiterte Public-Key-Infrastrukturen

Die Beschränkungen von zentralisierten PKIs und dem Web-Of-Trust zu umgehen, deren Vorteile aber gleichzeitig auf drahtlose Netzwerke anzuwenden, ist das Ziel der beiden Folgenden erweiterten Verfahren.

2.4.1 Polynomial Secret Sharing Bevor zwei Ansätze vorgestellt werden, die eine dezentrale Public-Key-Infrastruktur skizzieren, muss das Verfahren *Polynomial Secret Sharing* [8] erklärt werden, das beide Ansätze verwenden.

Polynomial Secret Sharing ermöglicht es, einen privaten Schlüssel auf mehrere Teile aufzuteilen, so dass der private Schlüssel nur bei Zusammenarbeit, unter Verwendung mehrerer Schlüsselteile errechnet werden kann. Diese Teile werden dann auf Knoten verteilt und kein Knoten kann den geheimen Teil eines anderen Knoten herausfinden, gemeinsam können die Knoten aber den privaten Schlüssel zur Signatur verwenden.

Der Algorithmus basiert auf der Idee, dass es genau eine Lösungspolynom der Ordnung $k-1$ gibt, das k gegebene Punkte schneidet. Gibt man also k Punkte vor, gibt es genau eine Lösung des Polynoms

$$f(x) = f_0 + f_1 x + f_2 x^2 + .. + f_{k-1} x^{k-1}$$

Setzt man den konstanten Teil f_0 des Polynoms auf den Wert des zu schützenden Geheimnisses SK, erhält man:

$$f(x) = SK + f_1 x + f_2 x^2 + .. + f_{k-1} x^{k-1}$$

Analog kann das Geheimnis berechnet werden durch:

$$\begin{aligned} f(0) &= SK + f_1 x + f_2 x^2 + .. + f_{k-1} x^{k-1} \\ f(0) &= SK + 0.. + 0 \\ f(0) &= SK \end{aligned}$$

Man kann nun beliebige Punkte auf dem Graph $f(x)$ definieren und diese an die Teilnehmer an der verteilten Verschlüsselung weitergeben, wie in Abb. 4 auf Seite 14 skizziert. Das Geheimnis SK kann aus hinreichend vielen Punkten

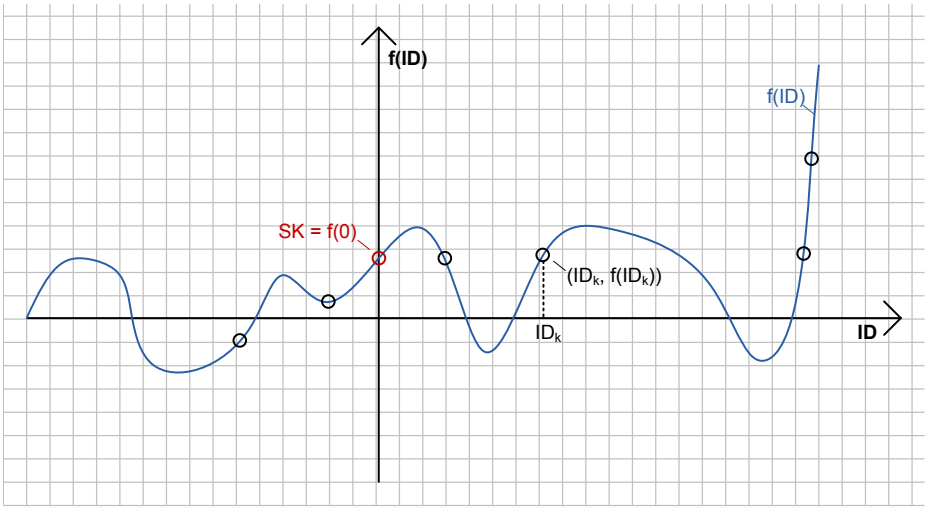

Abbildung 4. Lagrange-Interpolation

interpoliert werden. Der Einfachheit halber wird hier angenommen, dass jeder Teilnehmer i eine eindeutige Identifikationsnummer ID_i besitzt, und der Teilnehmer i den Punkt $(ID_i, f(ID_i))$ zugeteilt bekommt, wie in Abb. 4 ersichtlich.

Aus einer Menge M von Knoten, mit $|M| = k$, kann das Geheimnis SK dann errechnet werden, indem aus den Punkten $(M_i, f(M_i))_{i \in M}$ mittels Lagrange-Interpolation das Ursprungspolynom $f(x)$ errechnet wird. Zum detaillierten Vorgehen, wie mittels Lagrange-Interpolation aus k Punkte auf einem Graphen das zugehörige Polynom der Ordnung $k-1$ errechnet werden kann, siehe [1]. Der Wert des Geheimnisses ist dann $SK = f(0)$, wie oben gezeigt. Mit diesem Verfahren ist es also möglich, den privaten Signaturschlüssel SK einer verteilten Zertifizierungsstelle im Netz unter allen Knoten aufzuteilen, so dass beliebige k Knoten den Schlüssel gemeinsam wieder errechnen können.

2.4.2 Threshold Cryptography *Threshold Cryptography* [2] basiert auf der Idee, die zentrale Vertrauensinstanz der PKI mit Hilfe des Polynomial-Secret-Sharing-Algorithmus auf mehrere mobile Agenten zu verteilen, um die Abhängigkeit der mobilen Agenten von einer zentralen Stelle zu lösen. Sollte diese nämlich korrumpiert werden, sind alle Agenten angreifbar, da sämtliche Vertrauensstellungen, die auf der CA basieren, nun potentiell unsicher sind. Mittels Threshold Cryptography ist es möglich, eine verteilte Vertrauensinstanz zu erzeugen, indem mehrere Agenten gemeinsam über einen verteilten Verschlüsselungsalgorithmus eine Zertifizierungsstelle bilden und diese gemeinsa-

[1] Lagrange Interpolation:
`http://www.cut-the-knot.org/Curriculum/Calculus/LagrangeInterpolation.shtml`

me Vertrauensinstanz anschließend zur Authentisierung beliebiger teilnehmender Clients verwendet werden kann. Ein potentieller Angreifer müsste, um an den privaten Schlüssel zu gelangen, nun alle Teilnehmer der verteilten Verschlüsselung unter seine Kontrolle bringen, was wesentlich schwieriger sein sollte, als die Übernahme einer einzigen, zentralen Zertifizierungsstelle.

Die richtige Wahl der Anzahl der Agenten, die an der dezentralen Zertifizierungsstelle mitwirken, ist hier entscheidend, da der Ausfall eines einzigen Teilnehmers zum Zusammenbruch der Signaturfähigkeit der verteilten Verschlüsselung führt. Als Folge können keine weiteren Zertifikate mehr signiert werden. Diese Zahl der Teilnehmer sollte also weder zu klein gewählt werden, da es die Sicherheit bereits bestehender Vertrauensstellungen gefährden würde, wenn Angreifer durch das Kompromittieren weniger Knoten die komplette CA übernehmen könnten, andererseits darf die Anzahl der Teilnehmer auch nicht zu groß sein, da ansonsten der Ausfall eines Teilnehmers immer wahrscheinlicher wird.

Nachteilig bei dieser Methode ist der notwendige Verwaltungsaufwand bei Änderung der Verschlüsselungsparameter oder bei Wiederherstellung der geheimen Schlüssel unter den Knoten. Da keine zentralen Instanzen existieren ist das Kontaktieren der beteiligten Knoten nicht ohne weiteres durchführbar und bereitet oftmals immense Probleme. Aus diesem Grund wurde eine Erweiterung des Verfahrens entwickelt, die im Folgenden beschrieben wird.

2.4.3 Erweiterte Threshold Cryptography Wie im vorgenannten Kapitel knüpft die *Erweiterte Threshold Cryptography* von Haiyun Luo und Songwu Lu [6] an die Idee an, die Zertifizierungsstelle auf mehrere Knoten zu verteilen.

Der private Schlüssel der verteilen Zertifizierungsstelle wird auch hier in mehrere Teile aufgeteilt, doch im Gegensatz zur Threshold Cryptography, in der jeder Schlüsselteil dann auf genau einen Knoten verteilt wird, bekommt hier *jeder* Knoten im Netz einen Teil des Schlüssels zugeteilt. Auf diese Weise sind beliebige Mengen von k Knoten in der Lage, neue Zertifikate zu signieren. Damit entfällt die Gefahr der Threshold Cryptography, dass einer der Knoten ausfällt und der Schlüssel so verloren geht. Da potentiell jede Menge von k Knoten Zertifikate ausstellen kann, entsteht kein Flaschenhals durch die zertifizierenden Knoten wie in Threshold Cryptography. Der Mechanimus skaliert daher deutlich besser mit der Größe des Netzwerks.

Die Erstellung des privaten Schlüssels wird anfangs von einem sogenannten Dealer übernommen, der zuerst ein Schlüsselpaar generiert und anschließend das Secret-Sharing-Polynom erzeugt, mit dem der Schlüssel berechnet werden kann. Punkte auf diesem Polynom werden dann auf die initialen Knoten verteilt und der öffentliche Schlüssel wird im Netz publiziert. Anschließend muss der Dealer wieder aus dem Netz austreten, bzw. den generierten privaten Schlüssel sofort zerstören, da er sonst im alleinigen Besitz des vollständigen privaten Schlüssels wäre und daher ein lohnendes Ziel für Angreifer darstellen würde.

Nachdem der Schlüssel auf die initialen k Knoten verteilt worden ist, sind diese nun in der Lage, gemeinsam alle anderen Knoten zu signieren und die Schlüsselteile weiter zu verbreiten. Auf diese Weise kann der Schlüssel nach und

nach im gesamten Netzwerk verteilt werden, ohne dass die zentrale Instanz des Dealers weiterhin notwendig wäre. Damit skaliert dieser Ansatz deutlich besser als die erwähnten zentralisierten Verfahren. Geht man von der Annahme aus, dass jeder zu initialisierende Knoten in seiner unmittelbaren Nachbarschaft genügend Knoten mit allen Schlüsselteilen findet, so beschränkt sich der Aufwand zum Initialisieren des Knotens auf lokale Kommunikation und es ist kein aufwändiges Routing notwendig.

Ein weiteres Problem stellt die Berechnung des geheimen Schlüssels aus den Punkten durch Interpolation dar, denn diese darf auf keinem Teilnehmer allein durchgeführt werden, da dieser sonst Zugriff auf den gesamten privaten Schlüssel hätte. Soll ein neues Zertifikat $CERT$ für einen Knoten signiert werden, wird dieses von jedem Knoten K aus der teilnehmenden Menge M, mit $|M| = k$ getrennt signiert und man erhält k verschiedene, teilweise signierte Zertifikate $CERT^{SK_k}$, $\forall k \in M$. SK_k wird dabei vom Knoten K selbst berechnet, indem er eine Lagrange-Interpolation mit seinem Schlüsselteil (ID, f(ID)) und den IDs der anderen teilnehmenden Knoten durchführt:

$$SK_k = f(k) * \prod_{l \in M, l \neq k} (\frac{ID_l}{ID_l - ID_k})$$

Diese Teile können dann miteinander multipliziert werden und man erhält:

$$\begin{aligned} CERT^{SK_1} * CERT^{SK_2} * .. * CERT^{SK_k} = \\ = CERT^{SK_1 + SK_2 + .. + SK_k} \\ = CERT^{SK} \end{aligned}$$

Damit ist das Zertifikat $CERT$ mit dem privaten Schlüssel SK potenziert und somit nach dem RSA-Standard [3] signiert worden, ohne dass ein Knoten in den Besitz des vollständigen, privaten Schlüssels gekommen ist.

Die Initialisierung von unbekannten Knoten folgt einem Vertrauensmodell, ähnlich dem Web-of-Trust. Jeder Knoten hilft bei der Initialisierung eines neuen Knoten mit, falls ihm keine negativen Informationen über den zu initialisierenden Knoten vorliegen. Ein neu ankommender Knoten bekommt so auf jeden Fall ein Zertifikat ausgestellt.

Um sich später als kompromittiert erweisende Knoten bereits vor Ablauf der Zertifikate aus dem Netz auszuschließen, werden Sperrlisten netzwerkweit ausgetauscht. Damit wird sichergestellt, dass bekannte Angreifer trotz gültigem Zertifikat keinen Zugriff mehr auf das Netz bekommen, selbst wenn sie sich mit neuen Knoten verbinden, die nichts von der ursprünglichen Sperrung mitbekommen haben. Da Zertifikate ein festgelegtes Verfallsdatum besitzten, müssen die gesperrten Zertifikate in der Zertifikatssperrlisten nicht ewig vorgehalten werden, sondern können nach Ablauf automatisch gelöscht werden, da sich der Angreifer dann ohnehin nicht mehr mit diesem Zertifikat authentisieren kann.

Die Knoten sind dadurch gezwungen, ihre Zertifikate regelmäßig vor deren Ablauf zu erneuern und sich damit ihre Vertrauenswürdigkeit bestätigen zu lassen.

Als weiterer Sicherheitsmechanismus werden die Schlüsselpolynome der Knoten in regelmäßigen Abständen erneuert, um Kryptoanalysen erschweren. Der private Schlüssel wird dabei nicht verändert, da ansonsten alle signierten Zertifikate ihre Gültigkeit verlieren würden und die Initialisierung des Netzes von vorn beginnen müsste.

Stattdessen wird von einer Menge M, mit $|M| = k$ Knoten ein Polynom

$$g(x) = g_1 x + g_2 x^2 + ..g_{k-1} x^{k-1}$$
$$g(0) = 0$$

erzeugt. Wendet man beide Polynome $f(x)$ und $g(x)$ hintereinander an erhält man:

$$\begin{aligned} f'(x) &= f(x) + g(x) \\ &= f(0) + g(0) \\ &= SK + 0 \\ &= SK \end{aligned}$$

Der private Schlüssel wird also bei dieser Operation nicht verändert. Die am Update teilnehmenden Knoten $k_i \in M$ wenden das Polynom dann wie folgt an, um ihre Schlüsselteile anzupassen:

$$SK'_{k_i} = SK_{k_i} + g(ID_{k_i})$$

Damit haben die ersten k Knoten ein neues Polynom $f'(x)$ erzeugt und ihre Schlüsselanteile an dieses angepasst. Das Differenzpolynom $g(x)$ wird nun in das gesamte Netz propagiert, indem ein Mechanismus, ähnlich der Initialisierung eingesetzt wird, um Knoten nach und nach an das neue Polynom anzupassen. Um die Authentitizät des neuen Polynoms $g(x)$ sicherzustellen, wird dieses ebenfalls mit dem privaten Schlüssel SK signiert. Dies steigert die Sicherheit des privaten Schlüssels, da ein Angreifer nun nicht mehr in der Lage ist, durch lange andauerndes Abhören der Kommunikation genügend Informationen zu sammeln, um den privaten Schlüssel zu rekonstruieren.

Derzeit besteht in diesem Entwurf noch das Problem, dass das Netz in Unternetzwerke zerfallen kann, falls zwei oder mehr Stellen des Netzwerks gleichzeitig beginnen, das Polynom zu erneuern. Da der Schlüssel jedoch gleich bleibt, können sich diese Unternetze nach wie vor gegenseitig zertifizieren, allerdings können Knoten aus unterschiedlichen Unterteilen des Netzwerkes nicht mehr an einer gemeinsamen Zertifizierung neuer Knoten teilnehmen.

Weiterhin existiert in diesem Ansatz noch keine Technik zur Cross-Zertifizierung von Netzen mit unterschiedlichem privatem Schlüssel.

Dennoch ermöglicht es dieser Entwurf einem Ad-Hoc-Netzwerk, sich gegen eine geringe Anzahl von Angreifern zu verteidigen, sofern diese sich langsam genug bewegen. Schnell bewegende Angreifer sind eventuell schneller als die Verteilung der Zertifikatssperrliste und können so an mehreren Stellen angreifen,

bevor sie komplett gesperrt werden. Außerdem darf die Zahl der Angreifer nicht zu hoch sein, da diese sich sonst als legitime Knoten ausgeben könnten und so an genügend Schlüsselteile geraten, um sich anschließend zusammenzutun und gemeinsam aus ihren Schlüsselteilen den kompletten Schlüssel wiederherzustellen.

Das Verfahren funktioniert weiterhin nur, wenn ein Mechanismus existiert, der es Knoten ermöglicht, böswillige Knoten zu erkennen. Leider sieht der Entwurf bisher keine solchen Erkennungssysteme vor.

Die Autoren nehmen außerdem an, dass jeder Knoten eine feste Identifikationsmarke trägt, die nicht verändert werden kann. Darauf basiert die Identifikation von Knoten und Angreifern. Da in klassischen Ad-Hoc-Netzwerken eine solche (fälschungssichere!) ID nicht existiert – man denke nur an MAC-Adressen im WLAN – kann ein böswilliger Knoten, der dem Netzwerk bisher nicht negativ aufgefallen ist, diesen Umstand ausnutzen und sich mehrfach mit veränderter ID anmelden. Da jeder Knoten bei der Zertifizierung einen Teil des Schlüssels erhält, muss der Angreifer nur oft genug einen neuen Knoten simulieren, um an genügend Schlüsselteile zu gelangen. Damit kann dieser dann alleine den privaten Schlüssel des Netzwerks berechnen und somit die Sicherheitsvorkehrungen überwinden.

2.4.4 Fazit Nachdem einige interessante neue Ansätze vorgestellt wurden, zeigen die Schwächen dieser Verfahren, dass Sicherheit in Ad-Hoc-Netzwerken keine einfach zu garantierende Eigenschaft ist. Es existieren Ansätze, die versuchen, bekannter Konzepte auf den mobilen Bereich zu übertragen.

Allerdings setzen die vorgestellten Konzepte darauf, dass mobile Agenten einen eingebauten Bewertungsmechanimus besitzen, um das Verhalten anderer Knoten einzustufen, ohne diese Annahme weiter zu definieren. Um eine Zertifizierungshierarchie aufbauen zu können, müssen mobile Agenten beispielsweise entscheiden, ob sie der Stammzertifizierungsstelle vertrauen können. Ebenso müssen die Teilnehmer am erweiterten Threshold-Cryptography-Verfahren entscheiden, wie vertrauenswürdiges Verhalten definiert werden kann und wann ein Knoten als böswillig kategorisiert wird. Da die meisten Knoten in einem spontanen, drahtlosen Netzwerk noch nie zuvor Kontakt mit ihren Kommunikationspartnern hatten, gab es für diese bisher keine Gelegenheit zur Etablierung von Vertrauensstellungen durch länger beobachtetes, gebührliches Verhalten. Daher müssen Vertrauensstellungen häufig abgeleitet werden aus Bezeugungen Dritter. Doch selbst die Beobachtung, dass sich ein Knoten über einen längeren Zeitraum vertrauensvoll verhalten hat, kann nicht als Basis für künftiges Vertrauen dienen. Sobald über Vertrauen automatisiert entschieden wird, haben Angreifer die Möglichkeit, die Verhaltensweisen zu imitieren, die einen Knoten veranlassen, dem Angreifer mit der Zeit zu vertrauen und anschließend ihr schädliches Werk beginnen. Dies erschwert zwar spontane Angriffe auf Netzteilnehmer, böswillige Absichten können jedoch niemals ganz ausgeschlossen werden [9].

Ein verwandtes Problem sicherer Kommunikation ist die Identifikation von Knoten. Nur wenn Knoten zweifelsfrei identifiziert werden können, ist in den

vorgestellten Konzepten Sicherheit zu realisieren, denn diese Konzepte bauen darauf, die Identität des Kommunikationspartners aufgrund global eindeutiger Identifikationsmerkmale zu erkennen, um zu entscheiden, ob dem Agenten vertraut werden kann. Zum Einen fehlen diese Merkmale häufig in spontanen Zusammenschlüssen von Knoten oder sind nicht auf jedem Knoten ausgeprägt, zum anderen steht diese Voraussetzung im absoluten Widerspruch zum Wunsch, die eigene Privatshpäre zu bewahren und daher möglichst wenige Informationen über sich preiszugeben. Diesem Aspekt widmet sich das nachfolgende Kapitel.

3 Privatsphäre, Datenschutz und Privacy

Der Schutz von personenbezogenen Daten stellt den mobilen Nutzer vor eine Reihe von Problemen. Zum einen müssen, um mehr als nur grundlegende Dienste in Anspruch nehmen zu können, schon eigene Daten preisgegeben werden. Sei es beispielsweise auf unterer Protokollebene die Geräteadresse des eigenen Endgeräts oder Benutzerinformationen zur Authentisierung. Zum anderen ist nach der Herausgabe von persönlichen Daten jedoch kaum verfolgbar, an wen diese weitergegeben werden. Daher ist ein striktes Reglement bereits bei der Herausgabe der Daten notwendig, um der unbefugten Weitergabe und Verwendung zuvorzukommen. Zuerst werden nun einige Begriffe definiert, im Folgenden werden dann Verfahren vorgestellt, die die Herausgabe von Daten kontrollieren und gegebenenfalls unterbinden könnne.

3.1 Begriffsklärung

Die Begriffe *Privatsphäre*, *Datenschutz* und *Privacy* sind nicht klar gegeneinander abzugrenzen, es gibt viele Überschneidungen. Im Deutschen wird „Privacy“ meist mit „Privatheit“, „Privatsphäre“, „Datenschutz“ oder „informationelle Selbstbestimmung“ [10] übersetzt. Nach [11] bezeichnet Privatsphäre die Kontrolle über persönliche Daten. Diese Daten zeichnen sich dadurch aus, dass sie einer bestimmten Person zuzuordnen sind (Personenbezogenheit) und für diese Person von Bedeutung sind (Sensitivität). Der „Arbeitskreis Medien“ unterscheidet für den Telekommunikationsbereich drei Arten personenbezogener Daten [12], die sich auf den mobilen Kontext übertragen lassen. Zum einen nennt der Arbeitskreis sogenannte Bestandsdaten (Stammdaten), die relativ zeitstabile Eigenschaften oder Attribute einer Person bezeichnen. Hierzu zählen beispielsweise Kundennamen, Rufnummern oder Anschriften. In Ad-Hoc-Netzwerken sind diese Daten also die Knoten-Identifikationsnummer oder der zugehörige Benutzer. Zum anderen gibt es neben den Stammdaten sogenannte Verbindungsdaten (Im mobilen Netz als Bewegungsdaten bezeichnet), die Aktionen und Verhalten von Benutzern beschreiben, wie beispielsweise „Surfer #845 besucht Webseite #237“ oder „Knoten X stellt Verbindung mit Knoten Y über Knoten Z her“. Die dritte Art von anfallenden Daten sind die sogenannten Inhaltsdaten, die zwischen den Kommunikationspartnern ausgetauscht werden (Beispiel: E-Mails, Sprache). Diese Daten stehen unter besonderem gesetzlichen Schutz.

Im Folgenden werden die Begriffe Privatsphäre, Datenschutz und Privacy synonym verwendet werden, da alle die gleichen schützenswerten (persönlichen) Daten meinen und deren Schutz vor unbefugtem Zugriff.

3.2 Warum schützen?

Welche Gründe sprechen dafür, private Daten nicht willkürlich preiszugeben, sondern wohlüberlegt nur so viel von sich zu offenbaren, wie benötigt?

Offensichtlich kann ein Onlineshop nur durch persönliche Angaben ein personalisiertes Angebot erstellen, vielen Geschäftspartnern wäre es unangenehm, nur mit Pseudonymen zu kommunizieren ohne das Gegenüber wenigstens namentlich zu kennen und Dienstanbieter in mobilen Umgebungen – beispielsweise ein öffentlicher Druckerdienst in der Stadtbibliothek – benötigen klar verifizierbare Angaben über den Benutzer, um die angebotenen Leistungen abrechnen zu können.

Andererseits sollte nicht alles über seine Person preisgegeben werden, da manche privaten Details möglicherweise missbraucht werden können, beispielsweise weil sie rufschädigend wirken könnten. Außerdem macht die Offenlegung aller privaten Details berechenbar und man wird ein leichtes Ziel für Identitätsdiebstahl und *Social Engineering*, also die Erschleichung von Zugangsdaten und geheimen Informationen durch das Ausnutzen sozialer Kontakte und der Kenntniss privater Informationen. Der Schutz der eigenen Privatsphäre sollte daher ein wichtiges Anliegen sein, gerade weil vielen gar nicht bewusst ist, welche Daten wann preisgegeben wurden und wo diese bereits überallhin weitergegeben wurden. Man muss sich darüber im Klaren sein, dass einmal herausgegebene Informationen in der heutigen Welt kaum noch zurückgenommen werden können, denn „das Internet vergisst nichts“ [11].

Nach [13] gibt es vornehmlich vier Argumente (*Aspects of Privacy*) für das Schützen personenbezogener Informationen: Jeder Mensch sollte die *Kontrolle über die Preisgabe seiner eigenen personenbezogenen Daten* besitzen. Der Verlust dieser Kontrolle der Privatsphäre resultiert in einem Verlust der informationellen Selbstbestimmung und führt den Menschen in die Abhängigkeit von seinen eigenen Daten.

Privatsphäre ist *Mittel zum Zweck*, um sich vor unerwünschter Kontaktaufnahme (beispielsweise Werbung) schützen zu können.

Jeder Mensch sollte über seine personenbezogenen Daten verfügen können um sein *Recht alleine gelassen zu werden* in Anspruch nehmen zu können.

Privatsphäre ist *Teil der Würde des Menschen*, denn sie dient dazu, den Menschen vor unbegründeten Anschuldigungen oder Abhörungen zu schützen. Dadurch hält sie das Informationsgleichgewicht zwischen Personen aufrecht, da niemand das Recht hat, über einen anderen unverhältnismäßig viele Informationen auszukundschaften. Dieses Argument dient nicht nur dazu, den Informationsfluss zwischen zwei Personen in der Waage zu halten, sondern Privatsphäre als regulierendes Element soll mittels Gesetzen und Normen zum Schutz der Privatsphäre verhindern, dass eine Informations-Elite entsteht, die Unterprivilegierte willkürlich ausspäht.

3.3 Gesetzliche Regelung

In Deutschland wird der Umgang mit personenbezogenen Daten durch das *Bundesdatenschutzgesetz* (BDSG) geregelt. Die rechtliche Grundlage für den Schutz von personenbezogenen Daten ergibt sich bereits aus dem Recht auf informationelle Selbstbestimmung nach dem Grundgesetz [14]. Dies wurde durch das sogenannte „Volkszählungsurteil“ des Bundesverfassungsgerichts 1983 bestätigt [15].

Nach dem BDSG ist das Aufzeichnen von Verbindungsdaten nur unter für Berücksichtigung der Notwendigkeit einer Speicherung dieser Daten erlaubt, wie beispielsweise für Abrechnungszwecke oder Daten, die zum Erstellen einer Verbindung benötigt werden. Darüber hinaus unterliegen Inhaltsdaten der Verbindung dem Fernmeldegeheimnis und dürfen daher nicht aufgezeichnet werden. Ausnahmen bilden Aufzeichnungen um die missbräuchliche Nutzung aufzudecken und zu verhindern.

Auf internationaler Ebene bilden die Richtlinien der *Fair Information Principles* (FIP), aufgestellt von der OECD 1980 [16], die Grundlage vieler Datenschutzgesetzte weltweit. Sie postuliert fünf Prinzipien, die die Basis aller Datenschutzerklärungen sein sollten und im folgenden erläutert werden:

Bevor Datenschutz und Privatsphäre von Belang werden, muss der Nutzer sich überhaupt im Klaren sein, dass persönliche Daten automatisiert abgefragt und gespeichert werden. Daher stellt das Prinzip *Notice/Awareness* das grundlegendste Prinzip dar. Alle nachfolgenden Prinzipien wären wirkungslos, wenn der Benutzer nicht über die Geschehnisse und Datenschutzrichtlinien informiert würde.

Es sollte dem Nutzer jederzeit ermöglicht werden, die Verwendung seiner Daten zu bestimmen. Dabei ist es nicht nur von Bedeutung, ob der Nutzer der Verarbeitung der Daten zur Bearbeitung einer konkreten Anfrage zustimmt, sondern der Benutzer sollte auch bestimmen können, welche weiteren Verwendungen seine personenbezogenen Daten finden dürfen. Diese Entscheidungsfreiheit geht als das Prinzip *Choice/Consent* in die FIP ein.

Um die Richtigkeit der Daten und deren regelgerechte Verwendung festzustellen sollte dem Benutzer jederzeit Zugriff auf die über ihn gespeicherten Daten gewährt werden (*Access/Participation*).

Das Prinzip *Integrity/Security* verlangt, dass personenbezogene Daten beim Dienstanbieter sicher gespeichert werden und verpflichtet den Anbieter zum sofortigen Löschen veralteter Daten und zur Sicherung der Daten gegen unerlaubte Zugriffe durch technisch und organisatorisch zumutbare Lösungen.

Da alle Bemühungen zum Datenschutz ohne eine Stelle, die die Einhaltung der gesetzten Richtlinien durchsetzt, wirkungslos sind, verlangt das letzte der fünf Prinzipien mit der Bezeichnung *Enforcement/Redress* das Vorhandensein einer solchen Instanz.

Es ist ersichtlich, dass es durchaus nationale und internationale Bemühungen gibt, Datenschutz durch gesetzliche Regelungen durchzusetzen. Allerdings stoßen diese Bemühungen nicht erst durch die Allgegenwärtigkeit von Computern auf Hindernisse.

3.4 Probleme

Eines der derzeit größten Probleme, die den Datenschutz betreffen, ist sicherlich das fehlende Bewusstsein, dass es im eigenen Interesse notwendig ist, persönliche Daten zu schützen. Obwohl von der zunehmenden „Datenschieberei“ (dem Verkauf und dem Handel mit Adressdaten) letztlich jeder persönlich betroffen ist, hinterfragen die wenigsten Menschen diese Datensammelwut. Nach Meinung von Gerhard Kongehl ist das erst 2001 novellierte BDGS bereits heute veraltet und entspricht nicht mehr den heutigen technischen Gegebenheiten. Da es zu einer Zeit verabschiedet wurde, in der das Internet kaum verbreitet war, nimmt es zu wenig Bezug auf die dadurch aufkommenden Bedrohungen [17].

Einerseits geht das Bundesdatenschutzgesetz nach Expertenmeinung also nicht weit genug, andererseits widerspricht das bisher gültige Verbot der Speicherung von Daten auf Vorrat der Idee von *intelligenten Agenten*, die Daten für bisher unbekannte Zwecke speichern, um später bei Bedarf darauf zugreifen zu können. Ein weiterer, großer Kritikpunkt am Bundesdatenschutzgesetz ist die fehlende Überwachung der Umsetzung durch (staatliche oder delegierte) Stellen. Zwar ist die Verwendung von persönlichen Daten gesetzlich geregelt, ob sich ein Unternehmen an diese Gesetze hält oder die Daten heimlich doch weiterverkauft und für andere als die vorgesehenen Zwecke verwendet, wird momentan nicht überprüft.

Eine Gefahr, die vom leichtfertigen Umgang mit persönlichen Daten ausgeht, ist der *Identitätsdiebstahl*, also das unberechtigte Aneignen einer fremden Identität und deren Missbrauch. So wurden 1999 bereits 39.000 Fälle gemeldet, in denen US-Sozialversicherungsnummer (Social Security Number, SSN) missbraucht wurden [11]. Es muss nicht erst zum Identitätsdiebstahl kommen, um das Missbrauchspotential von persönlichen Daten zu verdeutlichen. Heute dienen weiterverkaufte Adressdaten oft für Werbung zum Beispiel mittels sogenannter *Cold Calls*. Damit werden Anrufe bezeichnet, die ohne Einwilligung oder den Wunsch des Betroffenen geschehen und meist Werbezwecken und den Verkauf von Waren oder Dienstleistungen als Ziel haben. Mittlerweile floriert ein reger Handel mit Adressdaten und es haben sich Firmen etabliert, die als einzigen Geschäftszweck den Handel mit Adressen haben. Doch längst werden nicht mehr nur Kontaktadressen vermarktet, sondern zusätzlich vielerlei Informationen, wie etwa das Einkommen, Wohngegend oder soziographische Informationen der Adressinhaber. Der gläserner Bürger ist bereits Wirklichkeit geworden. Bemerkenswert ist hierbei, dass der Staat bereits durch strikte Datenschutzgesetze eingeschränkt wird und vielerlei Versuche unternimmt, diese Restriktionen, die ursprünglich nur für das Verfolgen von Schwerverbrechen und zum Zwecke der Terrorismusbekämpfung gelockert wurden, aufzuheben und auf Bagatellvergehen wie Urheberrechtsverletzungen auszudehnen. Damit werden ganze Bevölkerungsteile kriminalisiert. Private Unternehmen hingegen horten ungebremst immer mehr Daten ihrer (potentiellen) Kunden und können ihre Sammelleidenschaft ohne staatliche Regulierung fortsetzen.

3.5 Erweiterte Problematik in mobilen Szenarien

Neben den bereits beschriebenen Problemen beim Datenschutz komm in mobilen Szenarien zusätzliche Fragen auf, die den Schutz persönlicher Daten betreffen. Einige dieser Probleme werden nun vorgestellt um im Anschluss Lösungen für diese vorzustellen.

3.5.1 Sicheres Auffinden von Diensten Schon das Auffinden von Dienstanbietern kann sich für den Dienstnutzer äußerst komplex gestalten, da verschiedene Interessen einem einfachen, verzeichnisbasierten Auffinden im Wege stehen. Bevor die verschiedenen Argumente gegen ein direktes Auffinden erläutert werden, erfolgt ein Blick auf die klassische serviceorientierte Welt, dargestellt in Abb. 5 auf Seite 24. Hier publizieren Dienstanbieter ihre Dienste in einem sogenannten Verzeichnis (*register*), beispielsweise in einer UDDI-Registry [18]. Dienstnutzer fragen dann dieses zentrale Verzeichnis ab (*find*) und stellen eine Verbindung zum Dienstanbieter her (*bind*).

Bei der Suche nach Dienstleistungen gibt es mehrere Möglichkeiten für einen Nutzer, einen passenden Dienst im Verzeichnis ausfindig zu machen. Ist der genaue Name des Dienstes bekannt, kann in den sogenannten *White Pages* in der UDDI-Registry nach der Adresse des Anbieters gesucht werden, ähnlich einem Telefonbucheintrag. Sollte der Name des gesuchten Dienstes unbekannt sein, können die *Gelben Seiten* (*Yellow Pages*) der Registry abgefragt werden. Dort sind Dienste nach Kategorie, ähnlich einem Branchenbuch, gelistet.

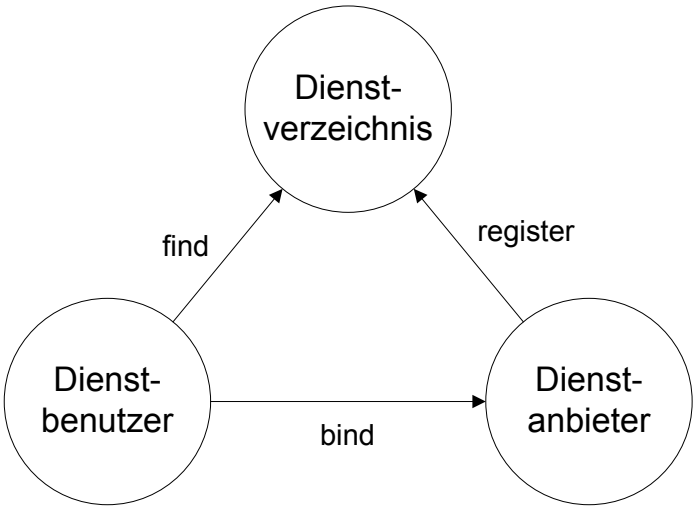

Abbildung 5. serviceorientierte Architektur

Da jeder in einem klassischen Verzeichnis gelistete Dienst von potentiell jedem Rechner aus gefunden werden kann, ist dieser verstärkt von Denial-of-Service-Angriffen gefährdet. Es liegt daher im Interesse der Dienstanbieter, möglichst nur einem eingeschränkten, vorher festgelegten Nutzerkreis, Zugang zum Dienst zu gewähren. Um Angriffe im Vorfeld minimieren zu können bietet es sich an, selbst das Auffinden des Dienstes nur diesen berechtigten Benutzern zu gewähren. Damit der Verzeichnisdienst entscheiden kann, wer auf welche Dienste zugreifen darf, ist es notwendig, dass der Diensterbringer genau spezifizie-

ren kann, unter welchen Bedingungen das Auffinden ermöglicht werden soll und welche Anforderungen der Dienst an Sicherheitsmerkmalen bietet und erfordert. Ebenso muss der Dienstnutzer in der Lage sein, seine Datenschutzanforderungen an den gesuchten Dienst formal ausdrücken zu können. Hierzu ist eine formale Beschreibungssprache wie etwa P3P (siehe unten) notwendig. [19]

3.5.2 Drahtlose, spontane Kommunikation Im Vergleich zur drahtgebundenen Kommunikation ist drahtlose Kommunikation wie sie in dieser Arbeit betrachtet wird, spontan. Das bedeutet, dass ein Netzwerk dynamisch aus zur Verfügung stehenden Knoten entsteht, ohne einer detaillierten Vorkonfiguration zu unterliegen. Dadurch unterliegen diese Netze einem steten Wandel und bestehen aus vorher unbekannten Teilnehmern.

Mit Änderungen an der Netztopologie ändern sich oftmals auch die Routingpfade durch das Netzwerk, daher müssen diese ebenso spontan auf- und wieder abgebaut werden können, wie die Knoten in das Netz ein- und austreten. Statisches Routing über vertrauenswürdige Knoten ist nicht möglich.

Es ergeben sich zahlreiche Probleme aus diesen Eigenschaften, unter anderem ist die Sicherheit des Netzwerkes gefährdet, da Angriffe von beliebigen Punkten innerhalb des Netzwerkes aus erfolgen können und *Edge-Firewalls* (*Edge*, deutsch: *Kante*, beschreibt eine Grenze zwischen Netzwerken) wie in drahtgebundenen Netzen hier nicht einsetzbar sind. Zum einen erfordern diese einen zu hohen Verwaltungsaufwand, um die Verbindungsregeln in Echtzeit an die Netztopologie anzupassen, andererseits gibt es im Ad-Hoc-Netzwerk kein „Edge“, auf der diese Firewall sinnvoll einsetzbar ist.

Da Kommunikation in diesen Netzen also erheblich leichter abgehört werden kann, ist eine Sicherung der Daten gegen unbefugten Zugriff äußerst wichtig.

3.6 Lösungsideen und -wege

Da das Interagieren mit Diensten und anderen mobilen Agenten stets die Preisgabe einiger Informationen erfordert, wäre es schlicht utopisch anzunehmen, dass persönlichen Daten komplett geheimgehalten werden können. Daher wird im folgenden der Begriff der k-Anonymität erläutert, um anschließend auf Basis dieses Begriffs einige Ansätze vorstellen zu können, die k-Anonymität garantieren können. Im Anschluss daran werden noch einige interessante Neuerungen vorgestellt, mit deren Hilfe die Herausgabe persönlicher Daten und deren Verwendung zumindest besser kontrolliert werden kann.

3.6.1 k-Anonymität Der Begriff der *k-Anonymität* wurde 1998 von Pierangela Samarti und Latanya Sweeney eingeführt [20]. Abstrakt beschreibt er die Nichtunterscheidbarkeit von k Tupeln anhand ihrer festgestellten Attribute. Für Ad-Hoc-Netzwerke bedeutet dies, dass Knoten in einer Menge von k mobilen Agenten von einem Dienst nicht genauer identifiziert werden können, weil ihre

Daten (Adresse, Identifizierungsinformationen usw.) für den Dienst absolut identisch erscheinen. Eine Möglichkeit, dies zu erreichen, ist beispielsweise das Herstellen der Verbindung über einen Proxy-Server. So kann der Dienst anhand der Verbindungsdaten nicht erkennen, mit welchem Client er kommuniziert, da alle von der Adresse des Proxys aus kommunizieren. Der Knoten verschwindet also in einer Menge. Wenn für ein Netzwerk gilt, dass alle Knoten mindestens k-anonym sind, so ist das Netzwerk k-anonym. Mit Hilfe des nun eingeführten Begriffes können nun Entwürfe vorgestellt werden, die es ermöglichen, k-Anonymität zu verwirklichen.

3.6.2 Das MIX-Modell von Chaum Für die nachfolgenden Ansätze grundlegend ist das MIX-Modell von Chaum [21]. Dieses zeigt eine Möglichkeit zur k-anonymen Nutzung von Netzwerken über sogenannte Vermittlungsrechner (*Mixe*). Um das Abhören der Kommunikationswege zu verhindern, schlägt Chaum asymmetrische Verschlüsselung zwischen allen beteiligten Knoten vor. Die Nachricht wird also auf jeder Kommunikationsstrecke neu verschlüsselt (*Punkt-zu-Punkt-Verschlüsselung*). Außerdem sollte die Nachricht zusätzlich vor dem Versenden durch den ursprünglichen Absender mit dem öffentlichen Schlüssel des Empfängers verschlüsselt werden, um böswillige Mixe am Zugriff auf die Daten zu hindern (*Ende-zu-Ende-Verschlüsselung*).

Damit der Empfänger der Nachricht dem Sender antworten kann, führt Chaum sogenannte anonyme Rückadressen ein. Diese ermöglichen dem Empfänger der Nachricht, dem ursprünglichen Sender zu antworten, ohne dessen Adresse zu kennen. Bei synchroner Kommunikation wie über das HTTP-Protokoll erfolgt die Antwort des Empfängers über den gleichen zugrundeliegenden Kanal, daher ist eine Rückadresse nicht erforderlich. Aber bei asynchronen Nachrichten wie beispielsweise SMTP ist das Antworten an einen anonymen Absender aus offensichtlichen Gründen ein Problem. Chaum löst dieses Problem, indem die Nachricht vor der Zustellung an den Empfänger durch den Mix um ein Pseudonym für den Absender ergänzt. Geht eine Rückantwort an dieses Pseudonym beim Mix ein, kann dieser das Pseudonym anhand einer Berechnungsfunktion auf den eigentlichen Empfängernamen zurückrechnen und zustellen.

3.6.3 Proxies (JAP, Rewebber) Die beiden Projekte JAP (*Java Anon Proxy*, entwickelt von der Technischen Universität Dresden [22]) und *Rewebber.com* (Projekt der FernUni Hagen [23]) implementieren das MIX-Konzept und stellen HTTP-Proxies zur freien Verfügung. Während JAP eine Kaskade von mindestens drei hintereinandergeschalteten Mixen bietet, ähnlich der Abb. 6 auf Seite 27, werden Daten bei Rewebber.com über einen einzigen Mix geleitet. Allerdings wird bei letzterem neben der Senderadresse auch die Empfängeradresse verschleiert und Benutzer können verschlüsselte URLs vom Proxy abfragen, ohne zu wissen, welche tatsächliche Webserveradresse dahintersteckt. Die URL wird zu diesem Zweck codiert und muss über eine Zahlenkombination vom Rewebber-Proxy abgerufen werden (z.B. http://../?ID=A12FF730D19FF). Der Proxy wertet die Zahlenkombination aus und erhält über eine interne Tabelle

die zugehörige URL. Diese wird abgerufen, alle Links werden durch Zahlenkombinationen ersetzt und der Original-Link wird in der Tabelle gespeichert. Anschließend wird das HTML-Dokument an den Client ausgeliefert.

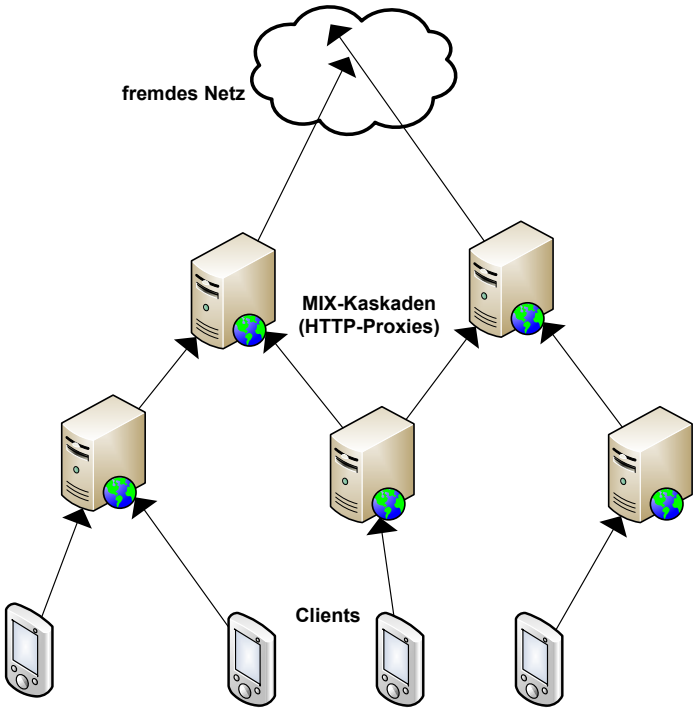

Abbildung 6. Kaskadierte Mixe

3.6.4 Crowds und The Onion Router *Crowds* wurde von Reiter und Rubin entwickelt [11], das bekannte TOR (*The Onion-Router*)-Netzwerk ist im Rahmen des Free Haven Project [24] entstanden. Im Gegensatz zu kaskadierten MIXen im vorigen Kapitel implementieren diese beiden Verfahren ein P2P-Netzwerk zwischen allen teilnehmenden Knoten. Dieses P2P-Netzwerk ist in Abb. 7 auf Seite 28 schematisch dargestellt. Um die Anonymität zu sichern, werden alle Anfragen auf zufälligen Wegen durch das P2P-Netz geroutet, anstatt vordefinierten Kaskaden zu folgen. Zusätzlich zur Anonymisierung von HTTP-Verbindungen können über ein TOR-Netzwerk beliebige TCP-Verbindungen weitergeleitet werden. Da bei jeder Anfragen zufällig entschieden wird, ob sie direkt an den Empfänger geschickt wird, oder an einen weiteren Nachbarknoten, kann jeder Knoten die Urheberschaft einer Anfrage abstreiten und behaupten, er sei nur ein Knoten auf dem Routingpfad.

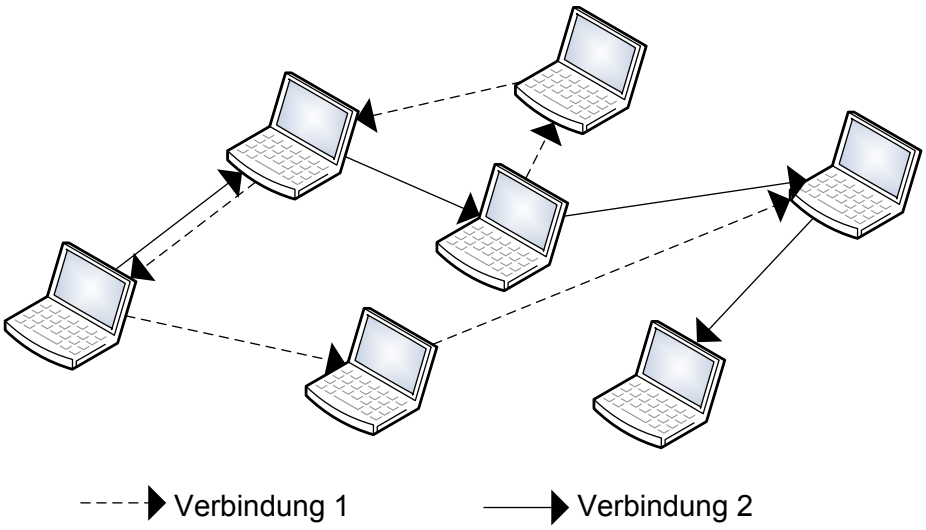

Abbildung 7. Crowds

Die vorgestellten Architekturen erlauben die Anonymisierung von Kommunikationspfaden und leisten so einen wichtigen zum Schutz der Privatsphäre. Das gesamte System kann jedoch durch kompromittierte Knoten ausgehebelt werden, indem ein Angreifer deren ein- und ausgehende Nachrichten protokolliert und so den gesamten Routingpfad wiederherstellen kann, falls er genügend kompromittierte Knoten im Netz verteilt. Es mag sein, dass es in verteilten Netzen schwer ist, viele Knoten gleichzeitig abzuhören, doch sollte man die Möglichkeiten großer Serviceprovider oder der Geheimdienste nicht unterschätzen. Diese sind sicherlich in der Lage, Nachrichten von einer breiten Anzahl Knoten gleichzeitig abzufangen. Um diesem Angriff vorzubeugen, können Nachrichten in einem Knoten gesammelt und dann gemeinsam versendet werden, so dass die Kopplung zwischen Empfang und Senden der Nachricht verloren geht. Außerdem können gelegentlicher Dummy-Nachrichten zwischen Knoten ausgetauscht werden, um so statistisches Rauschen zu erzeugen. Beide Techniken verschlechtern jedoch die Netzperformance, da erstere die Latenzzeiten der Nachrichten erhöht und letztere für gesteigerten Nachrichtenverkehr sorgt. Zusammen mit dem Problem der bisherigen Architekturen – der Notwendigkeit komplexe Vorarbeit in Form von Schlüsselverteilung durchführen zu werden müssen, bevor alle Knoten im Netzwerk miteinander verschlüsselt kommunizieren können – bedeutet das, dass Mixe zu statisch sind und zu viel Konfigurationsaufwand benötigen, um in mobilen Umgebungen erfolgreich einsetzbar zu sein. Da in diesen Umgebungen die Kommunikation dynamisch und zwischen spontan ein- und austretenden Knoten erfolgt, ist das Konzept von Chaum nicht geeignet. Ein weiterer Kritikpunkt an der Idee ist, dass sie ausschließlich dazu dient, Kommunikationsverhalten zu verschleiern. Daher sind auf jeden Fall zusätzliche Datenschutzmaßnahmen not-

wendig, um die Vertraulichkeit der Daten sowohl an den Endpunkten, als auch am Ziel zu gewährleisten [25].

Folgende Konzepte stellen Ideen vor, wie Datenschutzrichtlinien spezifiziert werden können und wie Anonymität auch in mobilen Szenarien gewährleistet werden kann.

3.6.5 Spezifikation von Datenschutzanforderungen und -zusicherungen mittels P3P Das Projekt *Platform for Privacy Preference* des W3C-Konsortiums [11] stellt eine Spezifikation für den Austausch von Datenschutzerklärungen dar. Der Austausch der standardisierten, in XML verfassten Erklärungen kann mittels HTTP oder eines beliebigen anderen Protokolls erfolgen. Dieses Dokument wird vom Dienstanbieter an den Benutzer bzw. dessen Agenten übermittelt. Es enthält die maschinenlesbare Datenschutzerklärung des kontaktierten Dienstes, bestehend aus einer Liste der erhobenen Daten, den Gründen für die Erhebung und deren weiteren Verwendung/Weitergabe. Sie soll dem Dienstnutzer ermöglichen, selbst zu entscheiden, ob die Verwendung der Daten den eigenen Datenschutzwünschen entspricht. Da diese Erklärung gleichzeitig automatisiert verarbeitbar ist, kann ein Endgerät auch ohne Zutun des Benutzers darüber entscheiden, ob die P3P-Erklärung des Dienstanbieters mit den gespeicherten Datenschutzanforderungen des Gerätebesitzers konform ist. Das eigentliche Ziel von P3P, die beteiligten Akteure (Dienstnutzer und -erbringer) über die Datenschutzpraktiken verhandeln zu lassen, wurde in der ersten Version bisher nicht implementiert [10]. Damit wäre es beispielsweise möglich, der Verwendung der Daten zu Marketingzwecken automatisiert zu widersprechen, oder genaue Einschränkungen auszuhandeln, an wen die Daten weitergegeben werden.

So ist P3P derzeit ein reines Austauschformat für die Datenschutzerklärung des Dienstanbieters. Daher sind weitere Maßnahmen erforderlich, um den abgegebenen Erklärungen glaubhaftigkeit zu verleihen. P3P könnte etwa durch Zertifizierungsmechanismen ergänzt werden, mit deren Hilfe ein unabhängige und vertrauenswürdige Stelle bestätigen kann, dass sich der Dienstanbieter an die abgegebene Datenschutzerklärung hält. So könnte das Vertrauen in diese Technik gesteigert und die Autonomie von mobilen Agenten erhöht werden, da diese mit Hilfe von P3P automatisiert entscheiden können, welche Dienste den Datenschutzanforderungen ihres Besitzers entsprechen.

3.6.6 Conditionally Anonymous Digital Signatures Die Idee von Conditionally Anonymous Digital Signatures [26] ist, dass mobile Agenten dieselbe Fähigkeit besitzen sollten wie ihre menschlichen Besitzer, nämlich entscheiden zu können, ob sie sich gegenüber Dritten ausweisen oder lieber anonym bleiben wollen.

Um dies zu erreichen, wird eine Public-Key-Infrastruktur aufgebaut, die, anders als „normale“ PKI, mit Zertifikaten arbeitet, die auf ein Pseudonym des Antragstellers lauten, anstelle seines richtigen (identifizierbaren) Namens. Ein Zertifikat enthält also den öffentlichen Schlüssel und ein beliebiges Pseudonym

des Antragstellers, sowie die digitale Signatur dieser Daten durch die Zertifizierungsstelle. So ist die Identität des Antragstellers zwar der Zertifizierungsstelle bekannt, nicht jedoch einem Dritten, der das Zertifikat lesen kann.

Vorteilhaft bei dieser Lösung ist die Möglichkeit zweier Agenten, trotz gegenseitiger Anonymität, eine digital signierte und damit verschlüsselte Verbindung aufzubauen. Die Agenten können, solange sie keine Zweifel am Goodwill des gegenüber haben, völlig frei und anonym kommunizieren. Sollte es notwendig sein, die Identität eines Agenten aufzudecken, beispielsweise um unberechtigte Handlungen zu verfolgen, kann die Zertifizierungsstelle die Anonymität des Zertifikats auflösen und die Identität des Antragstellers preisgeben. Nachteilig bei diesem Verfahren ist die Notwendigkeit, dass alle Kommunikationspartner ein Zertifikat besitzen müssen und es treten dadurch die bereits genannten Nachteile einer öffentlichen PKI auf. Zusätzlich kommt in diesem Ansatz hinzu, dass neue Pseudonyme immer bei der Zertifizierungsstelle (oder einer untergeordneten Stelle) angemeldet und bestätigt werden müssen, bevor sie benutzt werden können.

3.6.7 Privacy Awareness System (pawS) Das System „pawS“, wurde von Marc Langheinrich, Institute of Information Systems, ETH Zürich, entworfen [27] [10]. Er geht von der Annahme aus, dass digitale Spuren zwar weitgehend unkenntlich gemacht werden können, solange man auf soziale Interaktionen verzichtet, die Möglichkeiten zur Anonymisierung jedoch stark beschränkt werden, sobald reale Bezüge erforderlich werden (wie beispielsweise Lieferadressen, echte Namen, Bankverbindungen). Da diese Daten aus der realen Welt keiner Möglichkeit der Weitergabekontrolle in Form von Wasserzeichen unterliegen, ist es unmöglich, die weitere Verarbeitung der einmal herausgegebenen Daten im Nachhinein zu bestimmen und Aussagen zu machen, welcher Dienstanbieter die Prinzipien des Datenschutzes verletzt und personenbezogene Daten weitergegeben hat.

Daher entwickelte Langheinrich ein Modell zum Erkennen von Diensten, die persönliche Daten verarbeiten, und zum automatisierten Aushandeln von Datenschutzrichtlinien zwischen diesen Diensten und dem mobilen Agenten. Das Modell ist in Abb. 8 auf Seite 31 dargestellt, die Zahlen im Text in Klammern beziehen sich auf die Zahlen in der Abbildung. In diesem Modell publizieren Dienste über Nahfunkmedien wie Bluetooth oder WLAN permanent ein sogenanntes Privacy Beacon, das beispielsweise mittels P3P spezifizierte Auskunft über gesammelte Daten und deren Verwendungsrichtlinien des jeweiligen Dienstes enthält (1). Kommt ein mobiler Agent in Reichweite und empfängt ein Privacy Beacon, delegiert der Agent die Auswertung des Beacon an einen Privacy Proxy, der an einer beliebigen Stelle der Infrastruktur mit Zugriff auf den Dienstanbieter sitzt (2). Der Proxy hält die personenbezogenen Daten des Benutzers in einer Datenbank vorrätig und hilft dem Agenten einerseits Energie zu sparen, indem er die Verarbeitung der Richtlinien übernimmt, andererseits verschleiert er die Identität des Agenten gegenüber dem Dienstanbieter (3). Kommt der Proxy zu Schluss, dass die Datenschutzrichtlinien des Anbieters nicht mit

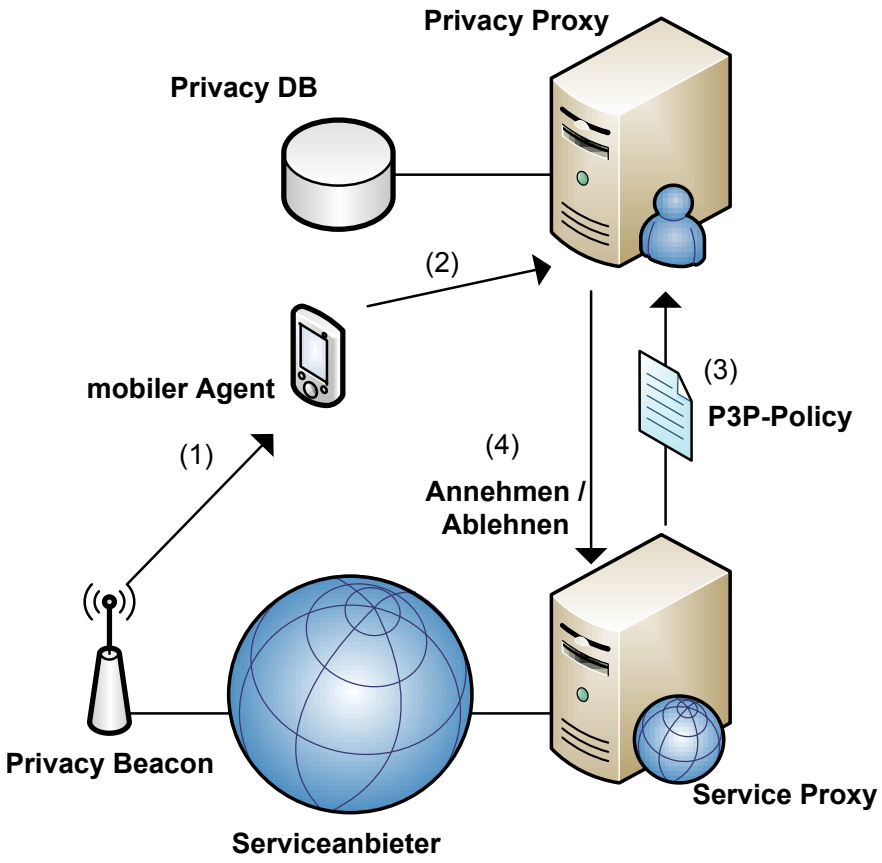

Abbildung 8. Funktionsweise von pawS

denen des Benutzers übereinstimmen, kontaktiert dieser bei proaktiven Diensten wie Videoüberwachungssystemen direkt den Service Proxy des angebotenen Dienstes und fordert die Deaktivierung des Dienstes (4). Selbstverständlich werden Dienstanbieter diesem Wunsch nicht immer nachkommen können, in diesem Fall ist eine Interaktion mit dem mobilen Agenten nicht möglich, da sie sich auf keine gemeinsam akzeptierte Datenschutzrichtlinie einigen konnten. Bei reaktiven Diensten wie beispielsweise Druckdiensten verhindert der Proxy den Abruf personenbezogener Daten aus seiner Datenbank durch den Dienst. Der Proxy speichert außerdem alle Richtlinien, unter denen der Zugriff auf die persönlichen Daten des Benutzers erlaubt wurde und stellt so eine Möglichkeit des Auditings bereit. Die grundlegende Problematik, dass nicht festzustellen ist, ob ein Dienst die Daten weitergibt, sobald er sie einmal erhalten hat, lässt sich damit jedoch nicht lösen.

3.6.8 Data Confidentiality and Secure Computation Die bisher vorgestellten Lösungsansätze betrachten lediglich die Anonymisierung von Kommunikationsverhalten oder Teilnehmern mittels Pseudonymen und verschlüsseln Daten bestenfalls auf ihrem Weg durch das Netz. Der Dienstanbieter verpflichtet sich jedoch nicht bindend zur Einhaltung der abgegebenen Richtlinien. Die Frage, wie der eigentliche Empfänger die übermittelten, persönlichen Daten weiterverarbeitet und ob dieser sich auch den Policies entsprechend verhält, bleibt daher unbeantwortet. Genau dies ist jedoch ein großes Problem, da selbst die beste Kommunikationsanonymität nicht davor schützen kann, dass der Dienstanbieter die persönlichen Daten der Nutzer missbraucht. An dieser Stelle versucht das Prinzip *Data Confidentiality and Secure Computation* (DCSC) [28] anzuknüpfen, indem es dem Dienstanbieter alle Klartextdaten vorenthalten und diesem trotzdem Berechnungen auf den Daten ermöglichen will. Um den Schutz der Inhaltsdaten und so Privatsphäre der mobilen Agenten zu erreichen, hebt dieses Verfahren die Verbindung zwischen dem Inhalt der Daten und dem Absender auf und versucht auf diese Weise Anonymität zu erzeugen.

Hierzu bedient sich das DCSC-Verfahren zweier kryptographischer Techniken: Die erste Technik, *Data Confidentiality* (DC), dient dazu, privaten Daten vor dem Versand mittels einer Einwegfunktion zu verschlüsseln. Es handelt sich hierbei um eine permanente Verschlüsselung und die versandten Daten können nicht wieder in Klartext lesbar gemacht werden.

Um trotzdem die benötigten Informationen aus dem chiffrierten Text ableiten zu können, muss der Empfänger die zweite Technik von DCSC anwenden. Mittels *Secure Computation* (SC) können Funktionen über verschlüsselten Daten berechnet werden, ohne die Daten im Klartext lesen zu müssen. Dadurch ist es möglich, den genauen Inhalt der Nachricht geheim zu halten und dem Empfänger trotzdem zu ermöglichen, Berechnungen auf den Daten durchführen zu können. Ein beliebtes Beispiel ist das *Millionärsproblem* [25] nach [29]:

Zwei Millionäre möchten herausfinden, welcher von ihnen der reichere ist, allerdings möchte keiner von beiden seine Vermögensverhältnisse offenlegen. Durch geschicktes Anwenden von RSA-Funktionen können beide Millionäre diese Frage tatsächlich lösen. Angenommen, Alice und Bob möchten ihr Vermögen vergleichen. Beide wissen, dass beide Vermögen im Bereich von 1 bis 10 Millionen US$ liegen.

Bob denkt sich zuerst eine Zufallszahl x aus und verschlüsselt diese mit dem öffentlichen Schlüssel von Alice. Das Ergebnis sei C. Von C zieht Bob dann sein Vermögen (in Millionen) ab, und sendet das Ergebnis an Alice. Zum besseren Verständniss ist das Verfahren in Abb. 9 auf Seite 33 dargestellt.

Alice entschlüsselt nun angefangen bei Bobs Wert $C - V$ zehn Werte

$$Y_i = (C - V + i)^{SK_{Alice}}, i = 1..10$$

mit ihrem privaten Schlüssel. Man beachte, dass der einzig richtig entschlüsselte Wert, der bei dieser RSA-Entschlüsselung auftritt, der V -te Wert der Reihe ist, da

$$Y_V = C - V + V$$

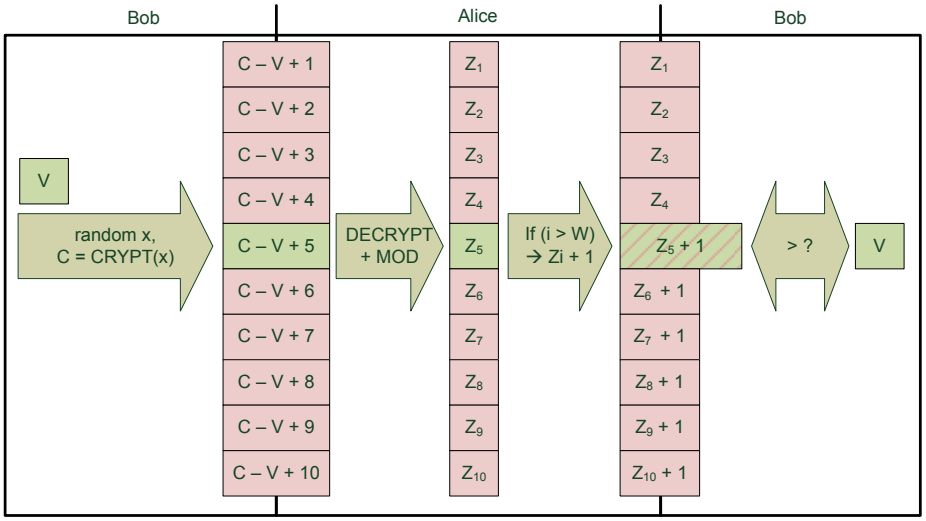

Abbildung 9. Eine Lösung des Millionärsproblem

gilt. Alle anderen Werte wurden auch RSA-entschlüsselt, sind streng genommen aber fehlerhaft, was von Alice nicht erkannt werden kann. Dies ist in der Grafik durch rote Unterlegung angedeutet worden.

Für jeden dieser Werte berechnet Alice anschließend den Modulus von einer beliebigen Primzahl p, deren Bit-Anzahl gleich der Hälfte der Bit-Anzahl von Bobs Zufallszahl ist.

Auf diese Wertereihe addiert Alice nun ab der Position, ab der ihr Vermögen stehen würde, eins auf und sendet die Reihe dann mit der Primzahl zurück an Bob. Falls Alice also W Millionen hätte, hätte sie ab der $W+1$-ten Position auf die Werte $C_i \bmod p$ eins aufaddiert ($C_i \bmod p + 1$.

Bob wiederum berechnet nun ausgehend von seiner ursprünglichen Zahl x den Modulus $x \bmod p$ und vergleicht die Zahl an der Stelle V in der Liste. Falls gilt $x \bmod p = C_V$, dann ist Alice reicher oder genau so reich wie Bob ($V \leq W$), falls gilt $x \bmod p \neq C_V$, dann ist Bob der Reichere von Beiden ($V > W$).

Verallgemeinert können mit dieser Lösung also zwei Zahlen darauf getestet werden, welche von beiden größer ist. Durch mehrmaliges Vergleichen können so auch mehrere Zahlen in eine Rangfolge gebracht werden. Ein Nachteil dieser Methode besteht allerdings darin, dass der Wertebereich (im Beispiel: 1 .. 10), komplett verschlüsselt und zwischen den Teilnehmern übertragen werden muss. Entweder muss der Bereich also hinreichend klein gewählt werden, oder Werte müssen gruppiert werden, wodurch diese Methode an Genauigkeit einbüßt.

Das Finden der richtigen Algorithmen zum Verschlüsseln und zum Berechnen der SC-Funktion ist nicht trivial, doch für viele Probleme, insbesondere Boolsche-Funktionen mit n-Eingabewerten bereits erfolgreich gelöst worden. Da Funktionen mit Ergebnissen von mehr als einem Bit, angenommen n-Bit, auf n

boolsche Funktionen mit je einem Bit Ausgabe umgeformt werden können, kann diese Lösung auf ein breites Spektrum von Funktionen ausgedehnt werden [25].

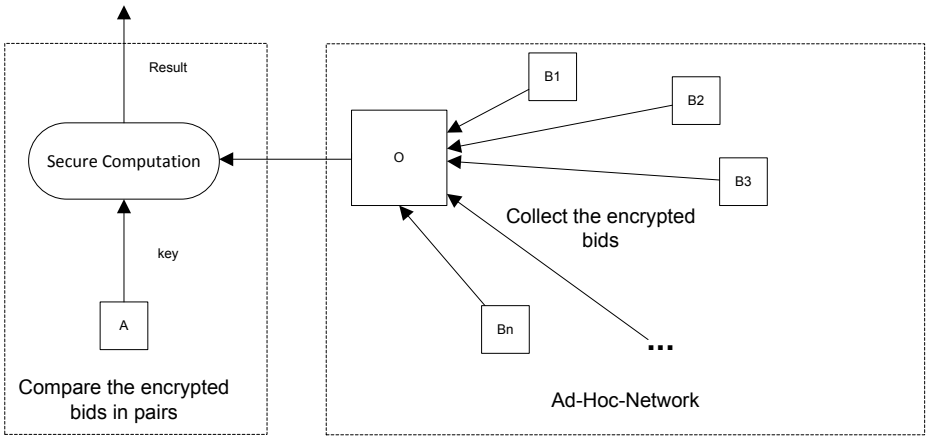

Abbildung 10. Auktion mit Hilfe von DCSC

Die Arbeitsweise dieses Entwurfs soll anhand eines Beispiels verdeutlicht werden, siehe dazu auch Abb. 10 auf Seite 34. Angenommen, in einem Ad-Hoc-Netzwerk findet eine Auktion mit mehreren Teilnehmern $B_1..B_N$ statt. Keiner der Bieter möchte sein Gebot für andere – nicht einmal dem Auktionator – lesbar preisgeben, stattdessen sendet er das Gebot DC-verschlüsselt an den Auktionator O. Dieser muss nun feststellen, wer Höchstbietender ist, auch wenn er die genauen Beträge nicht kennt. Hierzu wendet er Methoden der Secure Computation an, um mit deren Hilfe das Ergebnis zu ermitteln. Eine Möglichkeit, die Berechnung durchzuführen, ist es, die Lösung des Millionärsproblems paarweise auf die Gebote anzuwenden, um so das höchste zu finden.

Wie zu sehen war, ist es mittels DSCS möglich, bestimmte Berechnungen auf Daten durchzuführen, ohne den Klartext wiederherzustellen. Leider funktioniert das vorgestellte Verfahren nicht mit beliebigen Daten, denn beispielsweise müssen Adressen zwangsweise zurück in den Klartext dechiffriert werden, bevor eine Bestellung versendet werden kann.

4 Fazit

Es wurden nun verschiedene Ansätze vorgestellt, die Sicherheit und Datenschutz in mobilen Netzwerken unterstützen.

Leider wurde festgestellt, dass Sicherheit, welche auf Vertrauen basiert, nicht automatisch hergestellt werden kann, sondern dass der Mensch als Entscheidungsinstanz eingebunden werden muss. Dies führt dazu, dass sich mobile Agenten noch nicht völlig autonom bewegen können, da sie nicht wissen, wem sie in einem fremden Netz vertrauen können.

Gleichzeitig wurde gezeigt, dass ein hohes Maß an Sicherheit nur durch eindeutige Identifikation bestehen kann. Im zweiten Teil der Arbeit stellte sich allerdings heraus, dass Datenschutz eine gewisse Anonymität erfordert und der Tradeoff zwischen beiden Bedürfnissen von keinem der vorgestellten Verfahren zufriedenstellend überbrückt wird.

Eine denkbare Möglichkeit, Sicherheit und Datenschutz in mobilen Netzwerken noch weiter zu unterstützten, bestünde beispielsweise darin, das vorgestellte Verfahren der erweiterten Threshold Cryptography mit der Idee der Conditionally Anonymous Digital Signatures zu kombinieren und so die Vorteile beider Ansätze zu vereinen.

Nichtsdestotrotz versprechen die vorgestellten Konzepte eine sinnvolle Unterstützung von mobilen Knoten in Ad-Hoc-Netzwerken und zeigen die Notwendigkeit, weshalb Daten und Benutzer geschütz werden müssen.

Die beschrieben Angriffsszenarien lassen erahnen, wie vielfältig die möglichen Bedrohungen in mobilen Umgebungen sind, und sollen mögliche weitere Arbeitsfelder aufzeigen.

Ein großer Ansporn für weitere Arbeiten ist auch, das Bewusstsein der Benutzer zum gewissenhaften Umgang mit ihren persönlichen Daten anzuhalten, da derzeit ein sehr unvorsichtiger Umgang mit diesen Daten zu beobachen ist. Man muss sich nur einige Profilseiten der Nutzer von Social-Networks wie dem StudiVZ[2] oder englischen Pendants wie MySpace[3] ansehen, um einen erschreckend leichtsinnigen Umgang mit persönlichsten Daten festzustellen. So lange der Nutzer von sich aus so bereitwillig Daten herausgibt, sind hochtechnisierte Lösungen zu deren Schutz leider nutzlos.

Für die Zukunft bleibt die Hoffnung, dass das Datenschutzbewusstsein der Bevölkerung steigt, nicht zuletzt, da selbst der Staat derzeit als Datenkrake in Form von Gesundheitskarte, Biometrieausweis und lebenslangen Identifikationsnummern in Erscheinung tritt und viele Debatten über das Recht auf informationelle Selbstbestimmung angestoßen hat.

[2] Studiverzeichnis `http://studivz.net`
[3] MySpace `http://myspace.com`

Literatur

1. BSI: Mobile Endgeräte und mobile Applikationen, Postfach 20 03 63
   53133 Bonn. (2006) BSI-Broschüre.
2. Yan, Z.: Security in Ad Hoc Networks. `http://citeseer.ist.psu.edu/536945.html`
3. Rivest, R.L., Shamir, A., Adelman, L.M.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Technical Report MIT/LCS/TM-82 (1977)
4. Hauer, P.: Asymmetrische Verschlüsselung. Das Verfahren sowie die Vor- und Nachteile. Vortrag, Schulstraße 15
   18311 Ribnitz-Damgarten (December 2006)
5. International Telecommunications Union: ITU-T RECOMMENDATION X.509. `http://www.itu.int/ITU-T/asn1/database/itu-t/x/x509/1997/` (1997) [Online; accessed 15-June-2007].
6. Luo, H., Lu, S.: Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks. Technical report, Computer Science Department
   University of California, Los Angeles, Los Angeles, CA 90095-1596 (2000)
7. Abdul-Rahman, A.: The PGP Trust Model. Technical report, Department of Computer Science
   University College Londen, Gower Street, London WCIE 6BT, United Kingdom (1997)
8. Shamir, A.: How to Share a Secret. Commun. ACM **22**(11) (1979) 612–613
9. Langheinrich, M.: When Trust Does Not Compute – The Role of Trust in Ubiquitous Computing. Workshop on Privacy at Ubicomp 2003 (October 2003)
10. Langheinrich, M.: Personal Privacy in Ubiquitous Computing – Tools and System Support. Doktorarbeit (May 2005)
11. Langheinrich, Moschgath, Vogt: Privacy im Zeitalter von Ubiquitous Computing. Doktorandenseminar (WS 2000/01)
12. Arbeitskreis Medien: Mobilfunk und Datenschutz. Konferenz der Datenschutzbeauftragten des Bundes und der Länder **45. Sitzung** (February 1994)
13. Friedewald, M., Vildjiounaite, E., Punie, Y., Wright, D.: The Brave New World of Ambient Intelligence: An Analysis of Scenarios Regarding Privacy, Identity and Security Issues. Lecture Note in Computer Science **3934** (March 2006) 119–133
14. Deutscher Bundestag: Bundesdatenschutzgesetz. `https://www.datenschutzzentrum.de/material/recht/bdsg.htm` (1990) [Online; accessed 10-May-2007].
15. Dr. Benda, Dr. Simon, Dr.Hesse, Dr. Katzenstein, Dr. Niemeyer, Dr. Heußner, Niedermaier, Dr. Henschel: BVerfGE 65, 1 - Volkszählung. Urteil des Bundesverfassungsgerichts (October 1983)
16. Federal Trade Commission: Fair Information Principles. Privacy Online: A Report to Congress., 600 Pennsylvania Avenue, N.W.,
   Washington, D.C. 20580, Federal Trade Commission (1998)
17. Kongehl, G.: Experte: Neues Datenschutzgesetz schon veraltet. `http://www.heise.de/newsticker/meldung/18395` (June 2001)
18. OASIS: UDDI Executive Overview: Enabling Service Oriented Architecture. Technical report, Organization for the Advancement of Structured Information Standards (2004)
19. Trabelsi, S., Pazzaglia, J.C., Roudier, Y.: Enabling Secure Discovery in a Pervasive Environment. Lecture Note in Computer Science **3934** (2006) 18–31

20. Samarati, P., Sweeney, L.: Protecting Privacy When Disclosing Information: k-Anonymity and its Enforcement Through Generalization and Suppression. http://citeseer.ist.psu.edu/samarati98protecting.html (1998)
21. BSI: Anonymisierungsverfahren im Internet: Das MIX-Modell. Technical report, Bundesamt für Sicherheit in der Informationstechnik, Postfach 20 03 63 53133 Bonn (2001)
22. Köpsell, S.: AnonDienst - Design und Implementierung. http://anon.inf.tu-dresden.de/develop/Dokument.pdf (January 2004)
23. Rieke, A., Demuth, T.: JANUS: Server Anonymity in the World Wide Web. 10th Annual EICAR Conference (March 2001)
24. Dingledine, R., Mathewson, N., Syverson, P.: TOR: The Second-Generation Onion Router. Technical report, The Free Haven Project (2004)
25. Yao, A.C.: Protocols for Secure Computations. Technical report, University of California Berkeley, California 94720 (1982)
26. Yao, M., Henricksen, M., Foo, E., Dawson, E.: Offer Privacy in Mobile Agents Using Conditionally Anonymous Digital Signatures. Lecture Notes in Computer Science **3184** (2004) 132–141
27. Langheinrich, M.: A Privacy Awareness System for Ubiquitous Computing Environments. Lecture Note in Computer Science **2498/2002** (2002) 237–245
28. Peng, K., Dawson, E., Gonzalez, J., Nieto, Okamoto, E., López, J.: A Novel Method to Maintain Privacy in Mobile Agent Applications. Lecture Note in Computer Science **3810/2005** (2005) 247–260
29. PPCL: Solution to the Millionaire's Problem. http://www.proproco.co.uk/million.html (1998) [Online; accessed 15-June-2007].