Identitäten in Mobilen Ad hoc Netzwerken

Frank Kargl, Stefan Schlott, Michael Weber
Abteilung Medieninformatik, Universität Ulm

Abstract: Dieser Beitrag beschäftigt sich mit der Frage, welche Eigenschaften Identitäten in Ad hoc Netzen aufweisen müssen und beschreibt das Identifizierungssystem MANET-IDs.

1 Einleitung

Zur Absicherung von Mobilen Ad hoc Netzwerken sind unter anderem die folgenden drei Fragestellungen zu lösen: Wie werden Knoten eindeutig identifiziert? Wie kann die Routingschicht vor Manipulationen geschützt werden? Wie wird egoistisches Verhalten vermieden und die Kooperation von Knoten gefördert?

Diese Fragen greift unsere *Sicherheitsarchitektur für Mobile Ad hoc Netzwerke* (kurz *SAM*) [KSW04] auf und stellt die Komponenten MANET-IDs (Identifikation von Knoten), SDSR („Secure Dynamic Source Routing“ [KGSW05]) und MobIDS („Mobile Intrusion Detection System “ [KKSW04]) zur Verfügung. In diesem Beitrag wollen wir uns auf die Identifikationskomponente konzentrieren und vor allem erläutern, welche Aspekte bei der Identifizierung von Knoten in Ad hoc Netzen besonders zu beachten sind.

2 Verwandte Arbeiten

Da bei Funkkommunikation immer die Möglichkeit besteht, dass ein omnipotenter Angreifer sich als Man-in-the-Middle (MitM) in die Kommunikation, eine „Trusted Third Party“ (TTP) in spontan gebildeten Ad hoc Netzen oft nicht zur Verfügung steht und auch Aufbau und Verwaltung einer „Public Key Infrastruktur“ (PKI) viel Arbeit bedeutet, versuchen eine Reihe von Arbeiten, das Konzept der „Certificate Authorities“ (CAs) auf Ad hoc Netze anpassen kann. Zentrale CAs bergen in MANETs eine Reihe von Nachteilen [YK02], deshalb wurden verschiedene Vorschläge gemacht, wie sich die Funktion einer CA durch „Threshold Cryptography“ (TC) im Netz verteilt realisieren lässt [ZH99, YK02, LZK+02].

Das Konzept der verteilten CA mit TC hat in der Praxis jedoch einige Nachteile, welche auch die genannten Arbeiten nicht ausräumen können, allen voran die Tatsache, dass TC sehr rechen- und ressourcenaufwändig und somit schlecht für ein Ad hoc Netz aus mobilen, leistungsschwachen Knoten geeignet ist. Auch die Fragen wie auf Netzpartitionierungen und Rejoins reagiert werden soll und wie die TC CA bei nur zwei Knoten initialisiert wird, werden nicht beantwortet.

Einen anderen Ansatz verfolgt [HBČ01]. Hier wird das von Pretty Good Privacy bekannte Prinzip des „Web of Trust“ auf Ad hoc Netze übertragen. Anstelle einer CA signieren sich Knoten ihre öffentlichen Schlüssel gegenseitig. Hubaux e.a. beschreiben, wie die Schlüssel und Zertifikate im Ad hoc Netz verteilt gespeichert und vor allem wiedergefunden werden können. Ein Nachteil dieser Lösung besteht darin, dass nicht garantiert wird, dass ein Zertifikat tatsächlich verifiziert werden kann. Hierzu muss nämlich im Vertrauensgraph eine

Kante vom Zertifikatinhaber zum Zertifikatprüfer existieren, was gerade im Anfangsstadium eines Ad hoc Netzes nicht gegeben sein dürfte.

Eines haben alle Arbeiten gemeinsam: es wird nicht wirklich darauf eingegangen, was eigentlich mit einer Signatur ausgedrückt wird und wie die Identität beschaffen sein muss, die an einen Schlüssel gebunden wird. So werden lediglich Lösungen entwickelt, welche die Zuordnung eines Schlüssels zu einer abstrakten und nicht näher spezifizierten ID prüfen können. Es bleibt unklar, ob sich ein Benutzer beispielsweise neue Identitäten generieren kann, indem er neue Schlüssel erzeugt oder Benutzer im „Web of Trust“ dazu bringt, eine neue Identität zu zertifizieren. Ob eine Identität dabei für ein Interface, pro Gerät oder Benutzer gilt, ist auch unklar. Diese Fragestellungen sollen im Folgenden diskutiert werden

3 Identitäten in Ad hoc Netzen

In einem Ad hoc Netz gibt es eine ganze Reihe von Gründen, weshalb eindeutige IDs gebraucht werden. Dies fängt bei den IDs an, welche das Routingsystem für die Zustellung von Datenpaketen benötigt. Auch das Sicherheitssystem muss für Verschlüsselung oder Signaturen die Identität des Empfängers feststellen können. Bei Systemen, welche die Zuverlässigkeit von Knoten bewerten und unzuverlässige Knoten aus dem Netz ausschließen[1], muss der entsprechende Knoten ebenfalls sicher identifizierbar sein.

Um ein wirklich aussagestarkes Identifizierungssystem zu entwickeln, sind einige Fragen zu beantworten:

Was ist eine Identität? Wir bieten hierzu folgende Definition an:

Definition Identität: Die Identität eines Objekts ist eine eindeutige und unabänderlich mit diesem Objekt verknüpfte Eigenschaft, welche bei einem Objekt während seiner gesamten Existenz gleich bleibt und auch nicht auf andere Objekte übertragen werden kann.

In dieser Definition stecken eine Reihe von Anforderungen oder Kriterien, denen eine Identität genügen muss: Eindeutigkeit[2], unveränderliche Verknüpfung mit dem Objekt, lebenslange Gültigkeit und keine Übertragbarkeit auf andere Objekte. Während herkömmliche Routing-Adressen zwar meist eindeutig sind, sind alle anderen Kriterien nicht gegeben. Diese sind aber notwendig, um im Rahmen einer Sicherheitsarchitektur Angriffe zu verhindern, bei denen Knoten Identitäten austauschen bzw. neue generieren. Verfügt ein Knoten über beliebig viele Identitäten, ist eine Absicherung der Routing-Funktion kaum mehr machbar, da kooperative Sicherheitsmechanismen dann an der schieren Zahl der virtuell kooperierenden, böswilligen Knoten scheitern.

Definition Identifikator: Ein *Identifikator* ist ein Merkmal (oder eine Gruppe von Merkmalen), welches geeignet ist, ein Objekt zu identifizieren, das heißt, seine Identität zweifelsfrei festzustellen, und welches den Kriterien der Identität genügt.

Im Falle eines Ad hoc Netzes wären die Objekte bspw. Knoten. Hier sind verschiedene Identifikatoren denkbar, die Kriterien müssen dann aber durch geeignete Maßnahmen si-

[1] wie dies bspw. MobIDS macht

[2] auch in dem Sinne, dass ein Objekt nicht mehrere Identitäten bzgl. des gleichen Identifikators (s.u.) haben kann

chergestellt werden. Kann sich ein Knoten selbständig Identifikatoren für verschiedene Identitäten generieren, dann ist die Eindeutigkeit einer ID nicht mehr gegeben.

Welches Merkmal dient der Identifizierung? Da Knoten aus Sicht eines Ad hoc Netzes nicht über ein „natürliches“ Identifizierungsmerkmal verfügen, muss ein geeigneter Identifikator gefunden werden. Als Identifikatoren kommen dabei *geräteabhängige Identifikatoren*, *generische Identifikatoren* oder *Public-Key (PK) Identifikatoren* in Frage. Geräteabhängige Identifikatoren, wären beispielsweise MAC Adressen eines Interfaces. Generische Identifikatoren sind Zahlenfolgen, die oft auch als *universally unique identifiers* (*UUIDs*) bezeichnet werden. Schließlich kann man auch festlegen, dass der öffentliche Teil eines Schlüsselpaares als Identifikator dienen kann.

Geräteabhängige Identifikatoren müssen vor allem fälschungssicher und nicht übertragbar realisiert werden. Dies kann z.B. durch Trusted Hardware geschehen. Bei einer kryptographischen Absicherung dieser Merkmale stellt sich wieder die Frage nach der CA. Bei generischen und PK Identifikatoren stellt sich das Problem, dass die Eindeutigkeit sichergestellt und eine Weitergabe an andere Knoten ausgeschlossen sein muss.

Wie werden Identitätsänderungen verhindert?

Behauptung: *Ist die Erzeugung eines Identifikators alleine Sache des mobilen Knotens, so kann eine Identitätsänderung nicht verhindert werden.*
Argumentation: Sei $\mathcal{A}(X)$ das Verfahren, welches der Knoten mit Namen X zur Erzeugung seines Identifikators ID_X anwendet. Neben X gehen in $\mathcal{A}$ keine weiteren Informationen von außen ein. Der Algorithmus von $\mathcal{A}$ muss X bekannt sein, um ID_X erzeugen zu können. Folglich kann X den Algorithmus auch mit einem anderen Namen Y ausführen und erhält somit $\mathcal{A}(Y) = \mathrm{ID}_Y$. Wenn ID_X gültig ist, dann ist auch ID_Y gültig.

Somit wird im Umkehrschluss deutlich, dass ein Knoten seine Identität nicht komplett selbst generieren darf. Vielmehr muss mindestens eine weitere Partei an diesem Prozess beteiligt werden, die pro Knoten nur genau eine Identität vergibt. Die anderen Netzwerkknoten müssen sich dabei darauf verlassen können, dass diese dritte Partei ihre Aufgabe zuverlässig und gewissenhaft durchführt und nicht doch einem Knoten mehrere Identitäten zukommen lässt bzw. einzelne Knoten des MANETs bevorzugt. Aus diesem Grund bezeichnen wir diese Instanz als *Trusted Third Party (TTP)*. Es lässt sich also feststellen: *Ohne Beteiligung einer Trusted Third Party können keine verlässlichen Identifikatoren generiert werden.*

Viele der bisherigen Systeme berücksichtigen dies nicht und lassen daher beliebig viele Identitäten pro Knoten zu. Die meisten der bisher vorgeschlagenen Sicherheitsmechanismen[3] reagieren dabei empfindlich auf viele IDs pro Knoten. Die Forderung nach einer TTP steht im Gegensatz zu der vorherigen Argumentation, dass diese in Ad hoc Netzen schädlich wäre. Aufgabe einer Identifikationslösung für MANETs muss es daher sein, diesen Widerspruch möglichst aufzulösen. Dies versuchen wir mit der Konzeption des MANET-ID Systems.

[3] z.B. sichere Routingprotokolle, IDS usw.

4 MANET-IDs

Eine MANET-ID soll eine starke Identität im Sinne der aufgestellten Kriterien darstellen. Jedes Gerät wird hierzu bei seiner Fertigung mit einem Public-Key Schlüsselpaar (PK_X, SK_X) ausgestattet. Der öffentliche Schlüssel dient dabei als Identifikator des Knotens und wird MANET-ID genannt: $ID_X = PK_X$. Als Identifikator für das Routing wird eine crypto-based address (CBA) unter Verwendung einer Hashfunktion (h) generiert: $CBA_X = h(ID_X$ [Aur05]. Um die Erstellung neuer Identitäten zu verhindern, führen wir eine TTP ein. Diese signiert die MANET-IDs: $cert = E_{SK_{CA}}(PK_X)$. Wichtig ist, dass die Signatur keine weiteren Informationen an den Schlüssel bindet sondern lediglich dessen Gültigkeit dokumentiert. Somit entfallen aufwändige Identitätsprüfungen o.Ä.

Ein besonderes Problem ist der Rückruf von erteilten Zertifikaten, z.B. von Knoten, die als böswillig erkannt wurden. Dies kann durch Zertifikatsrückruflisten erfolgen, was aber hohen Kommunikations- und Speicheraufwand bei den Knoten erfordert. Alternativ kann man mit Zertifikaten sehr kurzer Gültigkeit arbeiten, was aber einen hohen Aufwand wegen der ständigen Verlängerung von Zertifikaten bedeutet. Basierend auf einer Idee von Micali [Mic96] haben wir ein Zertifikatsrückrufsystem für Ad hoc Netze entwickelt. Hierzu wird das Zertifikat um eine Seriennummer, einen Verifikator Y und ein Verfallsdatum erweitert: $cert = E_{SK_{CA}}(PK_K,\ serial,\ Y,\ valid\_until)$

Der Gültigkeitszeitraum T des Zertifikats wird in n Abschnitte der Dauer t eingeteilt, so dass $t = T/n$. Bei Ausstellung des Zertifikats generiert die CA eine Hashchain. $Y = Y^n$ ist deren Ende, wobei $Y^{i+1} = h(Y^i)$ und h eine kryptographische Hashfunktion (z.B. SHA-1). Den Startwert Y_0 hält die CA geheim. Wann immer ein Knoten mit anderen Knoten kommuniziert und diese ein Zertifikat prüfen müssen, schickt er den Verifikator $V_i = H^{n-i}(Y_0)$ für den momentanen Abschnitt i mit. Die anderen Knoten prüfen, ob $h^i(V_i) = h^i(h^{n-i}(Y_0)) = H^n(Y_0) = Y$.

Falls ja, ist der Verifikator gültig, sonst ungültig. Zu Beginn eines neuen Abschnitts $i + 1$ muss jeder Knoten von der CA einen neuen Verifikator V_{i+1} abrufen. Ist die Identität bei der CA gesperrt worden, gibt diese keine neuen Verifikatoren aus, die Identität verfällt. Typischerweise wählt man für T ein Jahr und $n = 365$. Somit muss ein Verifikator pro Tag und Knoten übertragen werden. Dies kann bspw. über MANETs mit Internetzugang oder GPRS erfolgen. Da wir nur eine lose Zeitsynchronisation der Knoten erwarten und Knoten vielleicht gelegentlich die CA nicht kontaktieren können, akzeptiert ein Knoten auch den vorherigen und nächsten Verifikator V_{i-1} und V_{i+1}. Natürlich kann auch der Empfänger die CA kontaktieren und selbst den aktuellen Verifikator erfragen. Ein Nachteil ist, dass ein Knoten selbst bei invalidiertem Zertifikat dieses noch ein oder zwei Abschnitte lang nutzen kann. Dies ist in der Praxis meist akzeptabel. Ist ein Zertifikat abgelaufen, muss der Knoten N die CA kontaktieren und eine Verlängerung seines Zertifikats beantragen. Hierbei wird auch eine neue Hashchain generiert.

Wie deutlich wird, benötigen MANET-IDs eine Reihe von Voraussetzungen: Die Knoten haben sporadisch[4] eine *Verbindung zur CA*, um einen neuen Verifikator abzurufen. Um die Gültigkeit von Zertifikaten zu prüfen, müssen die Geräte über *lose synchronisierte Uh-*

[4] in der Größenordnung von einigen Tagen

ren mit einer Genauigkeit im Stundenbereich verfügen. Bei Kontakt mit der CA lässt sich diese Synchronisation realisieren. Ins MANET integrierten und verteilten CA-Lösungen haben wie geschildert eine Reihe von Nachteilen. Die *CA* bei MANET-IDs *ist nicht ins MANET integriert* und wird bereits bei Geräteherstellung tätig. Um Missbrauch zu verhindern, sind Zertifikate nur eingeschränkt gültig und müssen dann verlängert werden. Die CA ist für die Zertifikatsverlängerung verantwortlich. Hierzu muss die CA den Knoten zumindest sporadisch zur Verfügung stehen. Der öffentliche Schlüssel der CA ist allen MANET Knoten bekannt. Die CA ist hinreichend gegen Angriffe geschützt. Die Knoten müssen in der Lage sein, zumindest *vereinzelte Public-Key Operationen* effizient durchzuführen. Nichtsdestotrotz werden diese weitestgehend vermieden, um die Effizienz des Systems zu gewährleisten.

5 Zusammenfassung

In diesem Beitrag haben wir die Frage von Identitäten in Ad hoc Netzen diskutiert. Im Gegensatz zu anderen Arbeiten wurde hierbei auch betrachtet, welche Anforderungen an diese Identitäten zu stellen sind, bevor diese in ein System eingebettet werden. Ausgehend von diesen Überlegungen wurde dargelegt, wie mit den MANET-IDs ein System realisiert werden kann, welches die geforderten Kriterien berücksichtigt. Die notwendigen Annahmen sind für viele Systeme in der Praxis gegeben.

MANET-IDs werden in der SAM Sicherheitsarchitektur eingesetzt, um im Rahmen des Secure Dynamic Source Routing Protokolls Knoten zu authentisieren und Veränderungen von Routen zu verhindern [KGSW05]. Im Mobile Intrusion Detection System (MobIDS) erlauben Sie die Identifizierung und langfristige Sperrung von Knoten [KKSW04].

Literatur

[Aur05] T. Aura. RFC3972: Cryptographically Generated Addresses (CGA). `http://www.ietf.org/rfc/rfc3972.txt`, 2005.

[HBČ01] Jean-Pierre Hubaux, Levente Buttyán und Srđan Čapkun. The Quest for Security in Mobile Ad Hoc Networks. In *Proceeding of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, 2001.

[KGSW05] Frank Kargl, Alfred Geiß, Stefan Schlott und Michael Weber. Secure Dynamic Source Routing. In *Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS-38)*, Hilton Waikoloa Village, HA, Januar 2005.

[KKSW04] Frank Kargl, Andreas Klenk, Stefan Schlott und Michael Weber. Advanced Detection of Selfish or Malicious Nodes in Ad hoc Networks. In *Proceedings of 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004)*, Heidelberg, Germany, August 2004.

[KSW04] Frank Kargl, Stefan Schlott und Michael Weber. Integrierte Sicherheit für Mobile Ad-hoc Netzwerke. In *WMAN 2004, part of the Informatik 2004*, Lecture Notes in Informatics (LNCS), Ulm, Germany, September 2004.

[LZK+02] Haiyun Luo, Petros Zefros, Jiejun Kong, Songwu Lu und Lixia Zhang. Self-securing Ad Hoc Wireless Networks. In *Seventh IEEE Symposium on Computers and Communications (ISCC)*, 2002.

[Mic96] Silvio Micali. Efficient Certificate Revocation. Bericht TM-542b, MIT Laboratory for Computer Science, Marz 1996.

[YK02] Seung Yi und Robin Kravets. Key Management for Heterogeneous Ad Hoc Wireless Networks. Bericht UIUCDCS-R-2002-2290, UILU-ENG-2002-1734, University of Illinois, 2002.

[ZH99] Lidong Zhou und Zygmunt J. Haas. Securing Ad Hoc Networks. *IEEE Network*, 13(6):24–30, 1999.