Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung Anwendungen 1 - WiSe 2010/11

Jan Benjamin Jochheim

sichere Gruppenkommunikation

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Inhaltsverzeichnis

Kurzzusammenfassung

Unter *sicherer Gruppenkommunikation* versteht man die Gesamtheit der Maßnahmen um eine Kommunikation mit vielen Teilnehmern gegen Störer von innen und Außen zu sichern. Eine sichere Gruppenkommunikation stellt viele neue Herausforderungen an Kommunikationsprotokolle die in der herkömmlichen Kommunikation mit nur zwei Teilnehmern so nicht auftreten. Diese Arbeit liefert einen Grundlegenden Überblick über die auftretenden Probleme und vorhandenen Lösungsansätze.

1 Motivation und Einführung in sichere Gruppenkommunikation

Gruppenkommunikation findet heute Anwendung in verschiedensten Bereichen. Dabei ermöglicht die Gruppenkommunikation im Gegensatz zur Unicast-Kommunikation einen Versand eines Nachrichten-Paketes an mehrere Empfänger gleichzeitig.

Eine Gruppenkommunikation kann mit der Unterstützung des darunter liegenden Netzes (z.B. im IP-Multicast) sehr ökonomisch mit den Ressourcen des Netzes umgehen. Im Idealfall skaliert die Gruppenkommunikation optimal, so dass bei wachsender Größe des Empfänger-Kreises nur eine konstante Menge an Ressourcen beim Sender benötigt wird.

Einige Anwendungsbereiche (vgl. Perrig und Tygar (2002)) von Gruppenkommunikation sind:

- IP-TV, mit verlustbehafteten Echtzeit-Daten
- Multi-Player Spiele, mit Echtzeit-Daten ohne Verlust
- Kommunikation von Sensor-knoten, auf Rechnern mit geringer Rechenleistung und Speicherplatz
- Software-Verteilung, mit unverfälschten Daten

Dies sind nur einige Einsatzbereiche in denen sichere Gruppenkommunikation benötigt wird. Daran läßt sich jedoch gut erkennen, das die Anwendungsszenarien sowie die daran beteiligten Rechner sehr unterschiedlich sind.

Häufig soll der Empfang einer Gruppenkommunikation auf einige Teilnehmer beschränkt werden, aus diesem Grund ist eine Absicherung der Kommunikation bei vielen Anwendungsszenarien erforderlich.

Im Gegensatz zur herkömmlichen Unicast Sicherheits-Protokollen (bei denen es nur zwei Kommunikationspartner gibt), sind bei Gruppenkommunikations-Protokollen weitere Probleme zu berücksichtigen. Daher ist ein einfacher Transfer der Unicast-Protokolle auf den Multicast nicht möglich.

1.1 Problemstellungen der sicheren Gruppenkommunikation

Bei der Betrachtung der Sicherheit eines Systems gibt es stets verschiedene Aspekte die Beachtet werden müssen. Mit sicherer Gruppenkommunikation ist in diesem Fall gemeint dass der Empfänger sicher sein kann, dass eine empfangene Nachricht wirklich vom angegebenen Sender stammt (**Authentizität**) und der Sender eine Nachricht auf bestimmte **Empfänger eingrenzen** kann.

Unicast Protokolle

Für sichere Unicast-Verbindungen (bei der die sichere Kommunikation nur zwischen zwei Partnern verläuft) gibt es bereits eine große Vielfalt von Protokollen, wie SSL für das WWW, bei dem die Kommunikation zwischen Browser und Server abgesichert wird.

Leider versagen traditionelle Unicast Protokolle im Gruppenkommunikationsszenarien aus verschiedenen Gründen.

Schwierigkeiten der Gruppenkommunikation

Bei sicherer Gruppenkommunikation gibt es einige Randbedingungen die für die üblichen Unicast Sicherheits-Protokolle (die die Verbindung zwischen zwei Kommunikationspartnern absichern) häufig nicht von solch großer Bedeutung sind. Die Gruppenkommunikationsprotokolle müssen damit umgehen können dass Nachrichten verloren gehen können. Dies ist insbesondere im Hinblick auf Verschlüsselungs und Authentisierungsinformationen problematisch.

Viele Einsatzgebiete von Gruppenkommunikationsprotokollen nutzen leichtgewichtige Rechner wie z.B. Sensorknoten. Daher haben die Algorithmen häufig wenig Speicherplatz und Rechnenzeit zur Verfügung. Die Protokolle sollten daher mit wenig Resourcen nutzbar sein und zusätzlich möglichst wenig Kommunikationsbandbreite für ihre Aufgaben nutzen.

Vielfach ist es wünschenswert dass an jede einzelne Nachricht Informationen über ihre Echtheit angehangen werden. Daher müssen solche Sicherungsfunktionen in sehr kurzer Zeit (bei Audio und Videodatendiensten möglichst in Echtzeit) berechenbar sein.

Nicht in jedem Szenario sind alle Teilehmer einer Gruppenkommunikation gleich vertrauenswürdig. Daher dürfen es die eingesetzten Protokolle nicht ermöglichen das ein Teilnehmer Pakete eines anderen fälscht.

In vielen Szenarien soll der Empfängerkreis eingeschränkt werden. Damit die Vertraulichkeit gewahrt bleibt, nutzen die Protokolle Verschlüsselungsmethoden. Hier ist häufig das Problem des Schlüsselaustausches zwischen Sender und den Empfängern gegeben.

1.2 Arten von Gruppenkommunikation

Unicast

Ein Unicast-Kommunikation findet immer zwischen zwei Gesprächspartnern statt, daher ist der Unicast ist im eigentlichen Sinne keine echte Gruppenkommunikation, er lässt sich aber dazu mißbrauchen. Wenn ein Sender N Teilnehmern eine Nachricht senden möchte, so muß er auch N mal das Nachrichtenpaket versenden.

In Peer-to-Peer Netzen (vgl. Shen u. a. (2009)) wird Unicast idr. eingesetzt, da es im Internet keine spezielle Infrastruktur benötigt.

Broadcast

Im Broadcast teilen sich die Teilnehmer idr. ein Übertragungsmedium. Das bedeutet z.B. im Fall von Ethernet oder auch 802.11 WLAN, dass beim Senden einer Broadcast-Nachricht sämtliche Teilnehmer nicht senden können, da das Medium belegt ist.

Multicast

Beim Multicast-Versand von Nachrichten wird der Sender von der Netzwerkinfrastruktur unterstützt, was bedeutet das es ausserhalb der Sender und Empfänger weitere Mechanismen gibt die das Eintreten neuer Teilnehmer in die Kommunikation sicherstellen und auch ihr Austreten ermöglichen.

Im Multicast werden Nachrichten durch die Netzwerkinfrastruktur repliziert, dies ist eine potentielle Gefahrenquelle die abgesichert werden muß. Wird der Eintritt in den Multicast nicht abgesichert, wäre es für einen Angreifer möglich, neue „Empfänger“ zu einer Multicast-Übertragung hinzu zu fügen und dadurch beim neuen Empfänger zusätzlichen Verkehr zu erzeugen. Da die Netzwerkinfrastruktur hier Pakete selbsttätig repliziert, wird diese Form des Angriffs **Traffic-Amplification** genannt.

2 Kryptografische Grundlagen

In diesem Kapitel werden einige kryptografische Verfahren dargestellt, die in vielen der heutigen, sicheren Gruppenkommunikations-Protokollen, in verschiedenen Kombinationen zum Einsatz kommen.

2.1 Symmetrische und Asymmetrische Kryptografie

Zur Verschlüsselung gibt es verschiedene Ansätze (vgl. Schneier (1995)). Die heutigen Kryptografieverfahren lassen sich in zwei Kategorien aufteilen.

- Symmetrische Kryptografie
- Asymmetrische Kryptografie

Die Verfahren der symmetrischen Kryptografie verwenden zur Ver- und Entschlüsselung denselben Schlüssel. Beim Versenden der Nachricht muss dieser Schlüssel sowohl dem Sender, zum Verschlüsseln der Nachricht bekannt sein, als auch dem Empfänger, damit dieser die Nachricht wieder entschlüsseln kann.

Bei der asymmetrischen Verschlüsselung gibt es zwei verschiedene Schlüssel, einen privaten und einen öffentlichen. Der öffentliche Schlüssel ist nicht geheim und kann veröffentlicht werden. Der private Schlüssel hingegen ist nur dem Empfänger bekannt. Der Empfänger kann mit dem privaten Schlüssel die mit dem öffentlichen Schlüssel verschlüsselten Nachrichten wieder entschlüsseln.

Digitale Signaturen lassen sich mittels asymmetrischer Verschlüsselung ebenfalls leicht realisieren. Dabei ist jedem Empfänger der öffentliche Schlüssel des Senders bekannt. Mit diesem öffentlichen Schlüssel kann er die Authentizität von Nachrichten überprüfen, aber selbst keine versenden.

Leider ist die asymmetrische Verschlüsselung für viele Anwendungsbereiche zu rechenaufwändig. Gerade bei der Übermittlung und Signierung jedes einzelnen Paketes eines Stroms von Echtzeit-Daten kann die asymmetrische Verschlüsselung zum Flaschenhals werden.

Bei symmetrischer Verschlüsselung ist der Schlüssel jedoch dem Sender und den Empfängern bekannt. Ein nicht vertrauenswürdiger Empfänger einer Gruppenkommunikation könnte somit Nachrichten fälschen, da ihm der Schlüssel des Senders bekannt ist. Eine symmetrische Verschlüsselung kann nicht direkt eingesetzt werden. Der Einsatz Asymetrischer Verschlüsselung ist für viele Einsatzzwecke ideal, jedoch für den Praxiseinsatz zu langsam. Daher gibt es verschiedene Ansätze, symmetrische Verschlüsselung einzusetzen oder geschickt mit Asymmetrischer zu kombinieren.

2.2 Kryptologisch sichere Hashfunktion

Hash-Funktionen sind in der Informatik weit verbreitet. Eine Hash-Funktion liefert eine Abbildung von Quelldaten auf ein Datum von vorab bekannter Länge. Idealerweise sollte die Abbildung unterschiedliche Quelldaten gleichmäßig innerhalb des Hash-Raumes verteilen.

Für den Einsatz innerhalb kryptografischer Vorgänge gibt es für Hash-Funktionen noch einige zusätzliche Anforderungen, die über die üblichen Eigenschaften hinaus gehen. Die wichtigsten Anforderungen sind die drei hier folgenden die anschließend erläutert werden.

- Einwegfunktion (preimage resistance)
- schwache Kollisionsresistenz (second preimage resistance)
- starke Kollisionsresistenz (collision resistance)

Das eine Hashfunktion eine **Einwegfunktion** sein soll, bedeutet das es bei einem gegebenen Hashwert nicht direkt auf den eigentlichen Inhalt geschlossen werden kann. Wenn a ein Hashwert ist der dem Angreifer bekannt ist, soll er von dem Hashwert nicht auf den Inhalt der Daten schliessen können. Das bedeutet wenn der Hashwert a bekannt ist und $a = h(x)$ gilt, soll es dem Angreifer nicht möglich sein auf x zu schliessen (ausser durch ausprobieren sämtlicher Kombinationen).

Die Anforderung der **schwachen Kollisionsresistenz** bedeutet, dass es nicht möglich sein soll, zwei verschiedene Nachrichten x_1 und x_2 zu erzeugen deren Hashwerte gleich sind. Wenn sich solche Nachrichten erzeugen lassen würden würde für sie $h(x_1) = h(x_2)$ gelten, wobei x_1 gegeben ist. Bei verschiedenen Nachrichten sollen die Hashes also unterschiedlich sein. Weil der Definitionsbereich immer größer als der Wertebereich einer Hashfunktion ist, werden sich Kollisionen nicht vermeiden lassen, allerdings wird hier nur gefordert dass diese nicht effizient berechenbar sind.

Eine **starke Kollisionsresistenz** hat dieselben Anforderungen wir die schwache Kollisionsresistenz, allerdings mit der Einschränkung dass die Nachrichten x_1 und x_2 vom Angreifer frei wählbar sind. Im Vergleich zu der schwachen Kollisionsresistenz hat Angreifer hier also mehr Freiheitsgrade um eine Nachricht zu fälschen. Es soll bei frei wählbaren x_1 und x_2 diese so setzen kann das $h(x_1) = h(x_2)$ gilt.

2.3 Message Authentication Codes

Eine Möglichkeit, die Echtheit von Daten, während einer Kommunikation zwischen zwei Teilnehmern festzustellen, sind Message Authentication Codes (MAC) (vgl. Rivest (1997)).

Das MAC-Verfahren setzt voraus, dass beide Parteien denselben geheimen Schlüssel kennen. Der Sender sendet seine Nachricht und hängt einen Message-Digest an die Nachricht an, die aus der Nachricht und dem geheimen Schlüssel gebildet wird. Der Message-Digest wird mit Hilfe kryptografisch sicherer Hash-Funktionen gebildet.

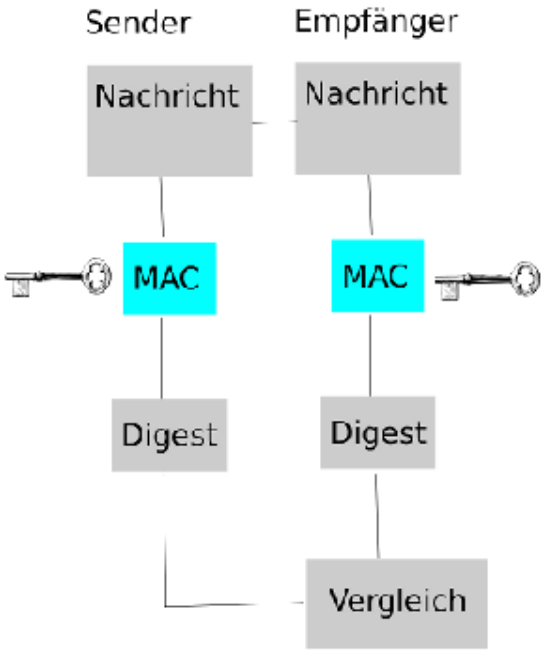

Abbildung 1: Message Authentication Codes

Der Empfänger kann die Echtheit des Paketes verifizieren, indem er die Hashfunktion ebenfalls auf das Daten-Paket und den Schlüssel anwendet. Das Vorgehen wird in Abbildung 1 gezeigt.

Das Konzept der MACs lässt sich im Umfeld sicherer Gruppenkommunikation mit nicht vertrauenswürdigen Kommunikationspartnern nicht direkt einsetzen, da allen Teilnehmern derselbe Schlüssel bekannt sein müsste. Allerdings lassen sich daran viele ähnliche Vorgehensweisen erläutern.

One-Way-Chain

Eine One-Way-Chain wird mittels wiederholter Anwendung einer kryptografischen Hash-Funktion $h()$ erzeugt. Dabei kann s ein zufälliger Startwert sein. Die Ergebnisse der jeweiligen Operation sind hier als H_n dargestellt.

- $H_1 = h(s)$
- $H_2 = h(h(s))$
- $H_3 = h(h(h(s)))$

- etc.

Die nach einem Ergebnis H_n folgenden Ergebnisse (H_{n+1}, etc.)können durch wiederholte Anwendung der öffentlich bekannten Hashfunktion berechnet werden. Der umgekehrte Weg, also die Ergebnisse H_{n-1} zu berechnen ist nicht effizient möglich, weil eine Kryptologisch sichere Hashfunktion eingesetzt wird.

3 Anforderungen an sichere Gruppenkommunikation

In diesem Kapitel werden Anforderungen an sichere Gruppenkommunikation vorgestellt und sicherheitsrelevante Anforderungen mit ihren Auswirkungen exmplarisch belegt.

3.1 Feedback-Implosionen vermeiden

Bei einer Gruppenkommunikation schickt ein Sender eine Nachricht an eine Vielzahl von Empfängern. Bei einem Nachrichten-Versand ist es wichtig, dass die Anzahl der Empfänger möglichst keinen direkten Einfluss auf den einzelnen Empfänger hat. Bei traditioneller Kommunikation zwischen zwei Kommunikationspartnern wird der Empfang einer Nachricht, vom Empfänger an den Sender bestätigt. Es findet also eine Kommunikation in rückwärtiger Richtung statt. Da solch eine Kommunikation bei einer wachsenden Anzahl von Empfängern einen Flaschenhals darstellt und **Feedback-Implosionen** hervorrufen würde, ist effiziente Gruppenkommunikation meist ohne eine Empfangsbestätigung realisiert. Dies birgt natürlich die Gefahr eines Daten-Verlustes, welches das entsprechende Kommunikations-Sicherungsprotokoll handhaben muss.

3.2 Kurzes Authentisierungs-Delay

Die Zeitspanne vom Empfang einer Nachricht bis zu der feststellung das diese Nachricht authentisch ist, wird Authentisierungs-Delay genannt. Bei vielen, der in Challal Y. (2004) vorgeschlagenen Verfahren, ist das Zwischenpuffern der Daten bis zur Feststellung der Authentizität einer Nachricht notwendig. Ziel der meisten Protokolle ist es, diese Zeit möglichst gering zu halten. Je größer der Zeitrahmen bis eine Authentisierung von Daten-Paketen stattfindet, desto größer ist auch der benötigte Datenpuffer auf der Empfänger-Seite.

3.3 Vertraulichkeit der Nachrichten

Gruppenkommunikation bedeutet nicht immer eine Kommunikation mit sämtlichen Teilnehmern einer Gruppe. Häufig möchte man den Empfang einer Nachricht auf einen kleinen Empfänger-Kreis beschränken. Die Vertraulichkeit des Versands einer Nachricht wird hier durch die Verschlüsselung sichergestellt.

Eines der auftretenden Probleme ist der Schlüssel-Austausch mit den berechtigten Empfängern einer Nachricht.

Schlüsselaustauschproblem

Bei sicherer Gruppenkommunikation stellt sich bei vielen Empfängern immer die Frage, ob diese auch vertrauenswürdig sind. Bei abgesicherter Unicast-Kommunikation werden häufig Sitzungsschlüssel ausgetauscht die beiden Parteien bekannt sind (z.B. bei den Message Authentication Codes). Bei solch einer Kommunikation ist es kein Problem, wenn der Empfänger Nachrichten mit dem Sitzungsschlüssel des Senders fälschen würde, denn er könnte höchstens die eigene Sitzung stören.

Bei sicherer Gruppenkommunikation muss jedoch davon ausgegangen werden, das einige Empfänger der Nachrichten nicht vertrauenswürdig sind. Bei einem globalen Sitzungsschlüssel könnte ein Teilehmer den Schlüssel weitergeben und somit die gesamte folgende Kommunikation für dritte offenlegen.

Verschlüsselungsverfahren

Mit dem Einsatz einer symmetrischen Verschlüsselung müsste jedem Empfänger, der zur Verschlüsselung verwendete Schlüssel sein.

Hingegen beim naiven Einsatz von langsamer Public-Key Verschlüsselung müsste jedes Daten-Paket mehrfach vom Absender für jeden Empfänger versendet werden, jeweils mit dem öffentlichen Schlüssel des Empfängers verschlüsselt.

Da Public-Key-Kryptografie im Vergleich zur symmetrischen Kryptografie sehr langsam ist, sind viele Verfahren dazu übergegangen, einen asymmetrischen Schlüssel zu generieren und mit diesem die Nachricht zu verschlüsseln. Dieser Schlüssel wird wiederum mit dem öffentlichen Schlüssel des Empfängers verschlüsselt und an das Daten-Paket an gehangen. Der Empfänger muss dann nur noch die Schlüssel „auspacken“ und damit die Nachricht entschlüsseln.

Dieses Vorgehen ist effizient, jedoch skaliert es bei vielen Empfängern nicht gut da der asymmetrische Schlüssel für jeden Empfänger an das Datenpaket angehangen werden müsste.

3.4 Absender-Authentisierung

In einer Gruppenkommunikation werden Nachrichten idr. von einem Sender an viele Empfänger gesandt. Dabei möchten die Empfänger sicherstellen, dass die Nachricht wirklich vom angegebenen Absender stammt und unverfälscht ist (vgl. Paar und Pelzl (2009)).

Die Daten dürfen nicht verändert worden sein, bzw. wenn es eine Veränderung gab, muss der Empfänger diese zuverlässig erkennen können. Ohne eine Authentisierung der Nachrichten des Senders, könnte jeder Teilnehmer einer Gruppenkommunikation die Rolle des Senders „kapern“ und Nachrichten in dessen Namen versenden.

Ein optimales Verfahren sind digitale Signaturen. Den Empfängern müsste dazu nur einmalig der öffentliche Schlüssel des Senders bekannt gegeben werden. Anschließend könnte der Sender jedes einzelne Daten-Paket signieren. Leider ist dieses Verfahren wie in Kapitel 2 dargestellt, nicht effizient für viele Daten zu bewerkstelligen, da die benötigten asymmetrischen Verschlüsselungsverfahren hierfür zu langsam sind.

3.5 Admission Control

Eine Gruppenkommunikation umfasst in vielen Szenarien eine Reihe von Empfängern, die häufig wechseln. Es können also im Laufe einer Kommunikation neue Empfänger hinzu kommen und andere die Gruppe wieder verlassen.

Unter Admission Control werden Maßnahmen verstanden, die ein sicheres Eintreten in eine Gruppenkommunikation ermöglichen.

Ein Angriffspunkt einer Gruppenkommunikation in Peer-to-Peer Netzen ist der Sybil Angriff. Dabei werden von einem Angreifer verschiedene Identitäten angenommen. Diese könnten z.B. verwendet werden, um neue Empfänger zu einer bestehenden Gruppensitzung hinzuzufügen, die jedoch entweder nicht existieren oder nicht beitreten wollten.Dadurch ist es möglich, den Datenverkehr an diese neuen „Teilnehmer“ umzuleiten, und deren Netze stark zu belasten. Ohne eine zentrale Autorität, der alle Kommunikationspartner vertrauen ist solch eine Sybil Attacke nicht immer zu unterbinden (vgl. Douceur (2002)), allerdings lassen sich ihre Auswirkungen begrenzen.

Eine Lösung wird, für ein ähnlich gelagertes Problem im IPv6 Umfeld vorgeschlagen, bei dem ein Host beweisen möchte, dass die IP-Adresse von welcher er sendet, auch wirklich seine eigene ist. Dabei wird das Prinzip kryptografisch generierte Identifier (*Cryptographically generated Identifiers*, CGIs) verwendet, um kryptografisch generierte IPv6 Adressen zu bilden, sogenannte *Cryptographically Generated Adresses*.

Es gibt verschiedene Ansätze Ryu und Mun (2007),Baumgart (2008) CGAs auch zur Overlay-Adressierung einzusetzen. Dabei werden CGAs auf die Gruppenkommunikation durch eine Kombination aus der IP-Adresse und dem öffentlichen Schlüssel eines Knotens angewandt.

4 Vorstellung ausgesuchter sicherer Gruppenkommunikations-Protokolle

In (Challal Y. (2004)) geben die Autoren eine Übersicht und Einordnung über den Forschungsstand von Gruppenkommunikations-Protokollen, die der Quell-Authentisierung dienen. Sie unterteilen die vorgestellten Protokolle in:

- Protokolle mit nicht-Abstreitbarkeit
- Protokolle ohne Abstreitbarkeit

Zu den Protokollen ohne nicht-Abstreitbarkeit zählen Sie 3 Typen von Protokollen auf:

- geheime Informations-Asymmetrie
- Zeit-Asymmetrie
- Hybride Asymmetrie

Die Namen deuten bereits darauf hin. dass zwischen Sender und Empfänger eine Asymmetrie aufgebaut wird, welche es dem Sender ermöglicht, Pakete mit Authentisierungs-Informationen zu füllen, die jedoch, durch kryptografische Hürden geschützt, von einem Empfänger nicht gefälscht werden können.

Bei der **geheimen Informations-Asymmetrie** nutzt der Sender eine Menge von Schlüsseln, mit denen er seine Pakete signiert. Den Empfängern bietet er jedoch nur eine Teilsicht auf diese Schlüssel, so dass diese ein Paket nur als vom Absender stammend identifizieren können, jedoch selbst keine Pakete fälschen können. Der Sender muss seine Menge von Schlüsseln regelmäßig austauschen, da beim Signieren einer Nachricht jeweils ein Teil seiner Schlüssel veröffentlicht wird. Wird mit diesem Verfahren an mehrere getrennte Empfängergruppen gesandt, wird dieses Problem noch größer da dann mehrere Empfänger zusammenarbeiten könnten, um an sämtliche geheime Schlüssel zu gelangen und damit Pakete zu fälschen. Nachrichten, die mit diesen Verfahren versendet wurden, können sofort nach Erhalt verifiziert werden. Zu diesem Verfahren zählt das von Desmedt vorgeschlagene (Desmedt u. a. (1992))

Protokolle der **Zeit-Asymmetrie** versenden zunächst die Nachricht mit einem Signatur-Datensatz. Nach einer bestimmten Zeit, bei der davon ausgegangen wird, dass jeder Empfänger das Paket erhalten hat, wird der Schlüssel der Signatur veröffentlicht, so daß der zuvor gesendete Signatur-Datensatz geprüft werden kann. Diese Verfahren sind immer auf eine

Zeit-Synchronisation mit dem Absender angewiesen. Nachrichten, die mit diesem Verfahren versandt wurden, können erst nach dem Senden des Schlüssels verifiziert werden, daher ist eine Zwischenpufferung notwendig. Ein Verfahren aus diesem Bereich ist TESLA (Perrig u. a. (2000)).

In der **Hybriden Asymmetrie** werden die Vorteile der beiden vorangegangenen Verfahren genutzt, um den jeweiligen Nachteilen (das Zwischenpuffern und die Zusammenarbeit mehrerer böswilliger Empfänger) zu entgehen. Eines der Protokolle aus diesem Bereich, das u.a. vom Entwickler von *TESLA* stammt, ist BiBa (Perrig (2001)).

Zu den Protokollen mit nicht-Abstreitbarkeit zählen die Autoren aus (Challal Y. (2004)) folgenden Typen auf:

- Signatur-Propagierung
- Signatur-Verbreitung (Signature dispersal)
- Differed signing

Bei der **Signaturpropagierung** enthält jede Nachricht signierte Authentisierungs-Informationen über andere Nachrichten (vgl. EMSS-Protokoll in Perrig u. a. (2000)). Damit propagiert sich die Authentisierungs-Information durch die Kette von Nachrichten. Bei den Protokollen aus dem Bereich **Signatur-Verbreitung** sammelt der Sender zuerst eine Reihe von Paketen und berechnet für jedes Paket einen Hash-Wert. Über die Aneinanderfügung dieser Hash-werte wird erneut ein Block-Hash gebildet der signiert und mit jedem Paket versandt wird. Der Empfänger kann nach dem Empfang aller Pakete eines Blocks den Hash berechnen und somit die Pakete authentisieren. Werden die Authentisierungs-Informationen geschickt in einem Baum verteilt, wie im SAIDA-Protokoll (vgl. Park u. a. (2002)), dann sind die Authentisierungs-Informationen auch gegen den Verlust einiger Nachrichten-Pakete gesichert. Die Signaturen werden geschickt über mehrere Pakete verteilt (Signatur Amortisation).

Beim **Differed signing** werden Einwegschlüsselpaare verwendet. Diese Schlüssel-paare werden anschließend dazu verwendet, um eine (mit wenig Rechenaufwand herzustellende) einmal Signatur herzustellen. Der Empfänger kann die Unterschrift des Einmal-Schlüssels verifizieren, weil der Sender beweist, dass der öffentliche Einmal-Schlüssel von ihm stammt. Den Beweis erbringt der Sender, indem er den öffentlichen Schlüssel mit einem weiteren, dem Empfänger bekannten „Langzeit-Schlüssel“ signiert. Da das zeitaufwendige berechnen mit asymmetrischen Verfahren (das Signieren mit dem „Langzeit-Schlüssel“) auch vor dem Vorhandensein der eigentlichen Daten erfolgen kann, eignen sich diese Verfahren gut für Echtzeitdaten. Eines der Verfahren aus diesem Bereich ist On-line/Off-line Signing (vgl. Even u. a. (1996)).

Im folgenden wird exemplarisch das TESLA-Protokoll, ein Vertreter von Gruppenkommunikations-Protokollen „ohne Abstreitbarkeit“ mit Zeit-Asymmetrie vorgestellt.

Das TESLA-Protokoll adressiert einen Teil der oben angesprochenen Probleme. Seine wichtigste Funktion ist die Absender-Authentisierung.

Die beiden vorgestellten Protokolle kümmern sich speziell um die

- Authentizität der Daten, dabei wird nur dem Sender vertraut.
- Skalierbarkeit bei vielen Empfängern.
- Tolerierung von Paket-Verlust
- Effizienz bei hohen Paketraten.

4.1 Das TESLA-Protokoll

Die Abkürzung TESLA steht für *Timed Efficient Stream Loss-Tolerant Authentication* und ist ein Protokoll für die Absender-Authentifizierung von Gruppenkommunikation.

Absender-Authentifizierung bedeutet, dass TESLA den Empfängern eines Datenpakets die Möglichkeit bietet, eindeutig festzustellen, ob es auch vom angegebenen Sender stammt.

Setup-Phase

In der Anfangsphase muss der Empfänger einmalig mit dem Sender einen Schlüssel austauschen und sich mit dem Sender lose zeit-synchronisieren.

Die Schlüssel-Übertragung muss über einen sicheren Weg geschehen, wie z.B. über eines der bekannten asymmetrischen Verschlüsselungsverfahren. Wie genau der Schlüssel-Austausch funktioniert ist Sache der Implementation und bei TESLA nicht genau spezifiziert.

Schlüssel-Generierung und Veröffentlichung

Der Sender muss vorab eine One-Way-Chain generieren , die er in (aus Sicht der Generierung) rückwärtiger Weise als Authentifizierungsschlüssel nutzt. Wird ein Daten-Paket gesendet, so enthält es zum einen die Nutzdaten, sowie einen Authentifizierungsschlüssel, der mittels einer Hash-Funktion aus den Nutzdaten N konkateniert mit dem n-ten Schlüssel der One-Way-Chain bildet.

Mit jedem Paket wird auch ein älterer Teil der One-Way Chain veröffentlicht. So ist es möglich die Authentifizierungsschlüssel älterer Pakete nachzuvollziehen.

Ein jedes Datenpaket hat daher den folgenden Inhalt:

Abbildung 2: One-Way Chain der TESLA Pakete

- Nutzdaten N
- Authentifizierungsschlüssel aus $h(H_n||N)$
- älterer Eintrag der One-Way-Chain H_{n+c}

In 2 ist eine Kette von Datenpaketen mit der One-Way-Chain und der Veröffentlichungsrichtung der Schlüssel, sowie der Generierungsreihenfolge der One-Way-Chain zu sehen.

Der Empfänger muss bis zur Veröffentlichung des Schlüssels die Pakete zwischenpuffern und darf die Pakete erst nach dem Empfang des Schlüssel-Paketes als gültig betrachten. Um das, durch das Zwischenspeichern auftretende Delay zu verkürzen, gibt es Ansätze wie TIK (*TIK: TESLA with instant Key disclosure* (vgl. chun Hu u. a. (2001)) die einen Schlüssel mit dem direkt folgenden Paket veröffentlichen. Dies ist natürlich nur möglich, wenn die Ausbreitungsgeschwindigkeit der Pakete für alle Netzteilnehmer gleich ist (z.B. im IEEE 802.11 W-LAN bei direkten Nachbarn).

4.2 TESLA und andere Multicast-Protokolle

Unter der Vielzahl der existierenden Gruppenkommunikations-Protokollen (siehe Challal Y. (2004)) ist TESLA einer der Protokoll-Vorschläge, die im Einsatz sind und mit dem unter dem RFC4082 (Perrig u. a. (2005))auch bereits eine Standardisierung vorliegt.

Dabei bietet TESLA eine große Robustheit gegen Paketverlust, eine gute Skalierbarkeit und kommt zudem mit einem kleinen Overhead aus. Ein Nachteil ist allerdings die benötigte lose Zeitsynchronisierung mit dem Sender. Zum einen kann während der Synchronisation eine direkte Kommunikation mit dem Sender erforderlich sein, was die Skalierbarkeit beeinträchtigt, zum anderen sind Empfänger mit unterschiedlichen Netzwerk-Delays problematisch. Eine Lösung sind hier verschiedene TESLA-Instanzen mit jeweils eigener Zeitspanne bis zur Veröffentlichung des Schlüssels.

Leider bietet TESLA auch keine nicht-Abstreitbarkeit des Sendens einer Nachricht.

Ein weiteres interesanntes Protokoll ist EMSS. EMSS steht je nach Quelle für *Efficient Multicast Stream Signature* (in Perrig und Tygar (2002)) oder *Efficient Multi-chained Stream Signature* (in Challal Y. (2004)) und gehört zu den Protokollen die Signaturpropagierung nutzen.

Im Gegensatz zu TESLA bietet EMSS eine nicht-Abstreitbarkeit des Absenders. Es hat eine große Robustheit gegen Paketverlust, einen niedrigen Overhead, mit einer geringfügig höheren Verzögerung zur Feststellung der Authentizität der Pakete.

Das EMSS Protokoll benötigt im Gegensatz zu TESLA keine Zeitsynchronisation mit dem Empfänger und liefert zusätzlich die Funktion der nicht-Abstreitbarkeit. Das bedeutet, dass der Empfänger eines Nachrichtenpakets einer dritten Partei zweifelsfrei beweisen kann, das dieses Nachrichtenpaket vom Sender stammt. Bei TESLA ist dies nicht möglich, da nach der Veröffentlichung eines Paket-Schlüssels ein fälschen der Pakete möglich ist.

5 Zusammenfassung und Ausblick

Es gibt eine Reihe von viel versprechenden Ansätzen von denen einige bereits in der Praxis eingesetzt werden. Unter den Protokollen In (Challal Y. (2004)) geben die Autoren einen Überblick über viele Gruppenkommunikations-Protokolle von denen in dieser Arbeit nur ein Teil dargestellt wurde.

In Wählisch u. a. (2009) stellen die Autoren einen gemeinschaftlichen Ansatz zur Authentisierung von Nachrichten in Peer to Peer Netzen dar. Dabei unterstützen sich die Hosts gegenseitig, so daß das Filtern einer gefälschten Nachricht auch den anderen Teilnehmern zugute kommt. Dieser Ansatz gibt, je nach Ausprägung keine vollständige Garantie das gefälschte Pakete erkannt werden, aber er ist ein interesannter kooperativer Ansatz.

Im weiteren Vorgehen sollten mehrere erfolgversprechende Protokolle auf ihre Performanz und die Einsatztauglichkeit für das o.g. Szenario in der Simulationsumgebung OMNet++/OverSim (vgl. Baumgart u. a. (2007)) getestet werden.

Da keines der vorgestellten Protokolle alle der Anforderungen an sichere Gruppenkommunikation abdeckt, wird es im weiteren Vorgehen sinnvoll sein, aus dem „Baukasten“ vorhandener Methoden ein neues Protokoll zu erstellen oder aus den bekannten Protokollen zum kombinieren.

5.1 Risikoabschätzung des weiteren Vorgehens

Das Risiko der Simulation mit vorhandenen Verfahren kann als relativ **niedrig** eingestuft werden, da es sich vielfach um Protokolle wie TESLA handelt, die bereits im Praxis-Einsatz genutzt werden. Die Spezifikationen der Protokolle liegen offen, allerdings gibt es kaum nutzbare implementierungen, daher wird solch ein Protokoll selbst (anhand der Spezifikation) implementiert werden müssen.

Die Implementierung eines *gänzlich eigenen Protokolls* ist mit einem wesentlich größeren Risiko und Aufwand verbunden. Da es sich um Sicherheits-Protokolle handelt, sollten hierbei nicht nur ausreichende Tests durchgeführt werden, sondern zusätzlich unbedingt veröffentlicht und begutachtet werden, da das Risiko besteht das

Ein eigenes Protokoll wird jedoch sicher immer auf vorhandene Mechanismen wie Kryptografie-Verfahren und Hashes zurückgreifen, daher stufe ich das Risiko hier auf **mittel** ein.

Literatur

[Baumgart 2008] BAUMGART, Ingmar: P2PNS: A Secure Distributed Name Service for P2PSIP. In: *Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*. Washington, DC, USA : IEEE Computer Society, 2008, S. 480–485. – URL http://portal.acm.org/citation.cfm?id=1371610.1372869. – ISBN 978-0-7695-3113-7

[Baumgart u. a. 2007] BAUMGART, Ingmar ; HEEP, Bernhard ; KRAUSE, Stephan: OverSim: A Flexible Overlay Network Simulation Framework. In: *Proceedings of 10th IEEE Global Internet Symposium (GI '07) in conjunction with IEEE INFOCOM 2007, Anchorage, AK, USA*, Mai 2007, S. 79–84

[Challal Y. 2004] CHALLAL Y., Bouabdallah A.: *A taxonomy of multicast data origin authentication: Issues and solutions*. 2004

[Desmedt u. a. 1992] DESMEDT, Yvo ; FRANKEL, Yair ; YUNG, Moti: Multi-receiver/multi-sender network security: efficient authenticated multicast/feedback. In: *Proceedings of the eleventh annual joint conference of the IEEE computer and communications societies on One world through communications (Vol. 3)*. Los Alamitos, CA, USA : IEEE Computer Society Press, 1992, S. 2045–2054. – URL http://portal.acm.org/citation.cfm?id=131607.131630. – ISBN 0-7803-0602-3

[Douceur 2002] DOUCEUR, John R.: The Sybil Attack. In: *Revised Papers from the First International Workshop on Peer-to-Peer Systems*. London, UK : Springer-Verlag, 2002 (IPTPS '01), S. 251–260. – URL http://portal.acm.org/citation.cfm?id=646334.687813. – ISBN 3-540-44179-4

[Even u. a. 1996] EVEN, Shimon ; GOLDREICH, Oded ; MICALI, Silvio: On-line/off-line digital signatures. In: *Journal of Cryptology* 9 (1996), S. 35–67. – URL http://dx.doi.org/10.1007/BF02254791. – 10.1007/BF02254791. – ISSN 0933-2790

[chun Hu u. a. 2001] HU, Yih chun ; PERRIG, Adrian ; JOHNSON, David B.: Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks, 2001

[Paar und Pelzl 2009] PAAR, Christof ; PELZL, Jan: *Understanding Cryptography: A Textbook for Students and Practitioners*. 1st. Springer Publishing Company, Incorporated, 2009. – ISBN 3642041000, 9783642041006

[Park u. a. 2002] PARK, Jung M. ; CHONG, Edwin K. P. ; SIEGEL, Howard J.: Efficient Multicast Packet Authentication Using Signature Amortization. In: *Security and Privacy, IEEE Symposium on* 0 (2002), S. 227. – ISSN 1540-7993

[Perrig u. a. 2005] PERRIG, A. ; SONG, D. ; CANETTI, R. ; TYGAR, J. D. ; BRISCOE, B.: *Timed Efficient Stream Loss-Tolerant Authentication (TESLA)*. Webseite. 2005. – URL http://www.ietf.org/rfc/rfc4082.txt

[Perrig 2001] PERRIG, Adrian: The BiBa one-time signature and broadcast authentication protocol. In: *Proceedings of the 8th ACM conference on Computer and Communications Security*. New York, NY, USA : ACM, 2001 (CCS '01), S. 28–37. – URL http://doi.acm.org/10.1145/501983.501988. – ISBN 1-58113-385-5

[Perrig und Tygar 2002] PERRIG, Adrian ; TYGAR, J. D.: *Secure Broadcast Communication in Wired and Wireless Networks*. Norwell, MA, USA : Kluwer Academic Publishers, 2002. – ISBN 0792376501

[Perrig u. a. 2000] PERRIG, Adrian ; TYGAR, J. D. ; SONG, Dawn ; CANETTI, Ran: Efficient Authentication and Signing of Multicast Streams over Lossy Channels. In: *Proceedings of the 2000 IEEE Symposium on Security and Privacy*. Washington, DC, USA : IEEE Computer Society, 2000, S. 56–. – URL http://portal.acm.org/citation.cfm?id=882494.884425. – ISBN 0-7695-0665-8

[Rivest 1997] RIVEST, Ron: *Lecture on Computer and Network Security: MACs*. 1997. – URL http://web.mit.edu/6.857/OldStuff/Fall97/lectures/lecture3.pdf

[Ryu und Mun 2007] RYU, Seonggeun ; MUN, Youngsong: Enhancement for security of peer-to-peer by expanding CGA mechanism. In: *Proceedings of the 2007 international conference on Computational science and Its applications - Volume Part II*. Berlin, Heidelberg : Springer-Verlag, 2007 (ICCSA'07), S. 1062–1071. – URL http://portal.acm.org/citation.cfm?id=1802954.1803059. – ISBN 3-540-74475-4, 978-3-540-74475-7

[Schneier 1995] SCHNEIER, Bruce: *Applied cryptography (2nd ed.): protocols, algorithms, and source code in C*. New York, NY, USA : John Wiley & Sons, Inc., 1995. – ISBN 0-471-11709-9

[Shen u. a. 2009] SHEN, Xuemin ; YU, Heather ; BUFORD, John ; AKON, Mursalin: *Handbook of Peer-to-Peer Networking*. 1st. Springer Publishing Company, Incorporated, 2009. – ISBN 0387097503, 9780387097503

[Wählisch u. a. 2009] WÄHLISCH, Matthias ; SCHMIDT, Thomas C. ; HEGE, Gabriel: Overlay authocast: distributed sender authentication in overlay multicast. In: *Proceedings of the 28th IEEE international conference on Computer Communications Workshops*. Piscataway, NJ, USA : IEEE Press, 2009 (INFOCOM'09), S. 387–388. – URL http://portal.acm.org/citation.cfm?id=1719850.1719945. – ISBN 978-1-4244-3968-3