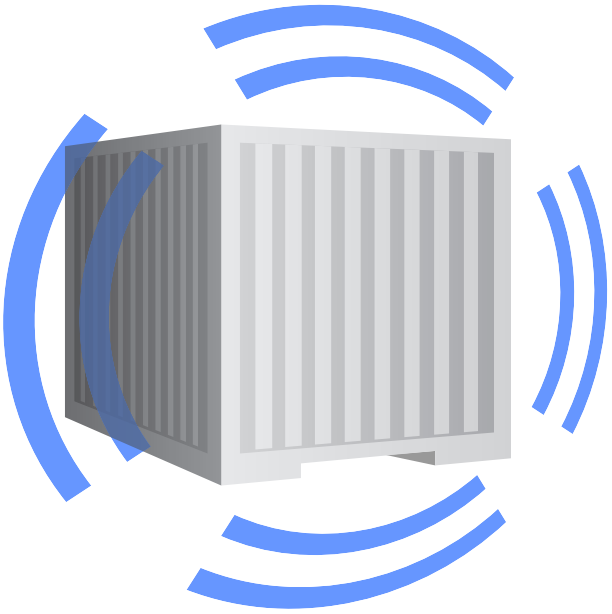
IT-Sicherheitsarchitektur zur Containerüberwachung



Dissertation

Jens Ove Lauf

9. Juni 2008

Gutachter:

Prof. Dr. Dieter Gollmann

Prof. Dr. Joachim Posegga

Technische Universität Hamburg-Harburg

Sicherheit in verteilten Anwendungen

`http://www.sva.tu-harburg.de/`

Harburger Schloßstraße 20

21079 Hamburg

Deutschland

Erklärung

Ich, Jens Ove Lauf, erkläre hiermit, dass ich die vorliegende Dissertation selbstständig sowie ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Emmelndorf, den 9. Juni 2008

Jens Ove Lauf

Inhaltsverzeichnis

Introduction

Abstract

This thesis proposes a security architecture for a container monitoring application which aims at increasing transport quality by damage detection. Furthermore, border control authorities get the possibility to better control the import of containers. A secure tunnel protocol between the sensor nodes in the container and an internet remote station is developed that protects confidentiality and integrity of sensor data in an energy efficient way. Another key building block of the architecture is a delegable access control mechanism, which ensures that only correct parties can access the database, which keeps records of the container transport.

Executive Summary

Since the Warsaw Pact fell in the early 1990s, there are only very few states, which do not take part in the free trade. The catchword globalisation stands for a new form of worldwide competition. Its fundament is container transport. Transporting a television set from Asia to Europe costs 10 $ – distance is no locational disadvantage anymore. The transport business has been booming for several years now. New technologies help minimise transportation costs. Unfortunately, these new trends also generate drawbacks. This thesis describes solutions for three problems the transportation sector has to face:

1. **Transport quality:** Cost minimisation leads to a growth of transportation damages as a side effect. These damages are seldom detected before delivery.
2. **Fear of terrorism:** Since 2001 national border controls try to improve entry control. Container import is a weak point of border protection.
3. **Heterogeneous contractual landscape:** Container transportation still is subject to individual contracts and various local legislations.

The basis for this thesis is the MASC system, where MASC stands for Monitoring and Security of Containers. Chapter 2 introduces this application and shows how the problems above are addressed by embedding sensor units in containers. New risks are raised by the automation of container documentation, monitoring and tracing. These risks are further analysed in chapter 3. As a result, the infrastructure of the container monitoring application is shown, which is designed to minimise the identified risks.

The core of this work is the Information Security part. It starts in chapter 4 that explains how a secure tunnel between container and database server is established. Generally, the security of a system is as

secure as its weakest element. That is why further protection means have to be implemented. These are the focus of the subsequent chapters. Limitations of the MASC system and differences to related developments are also highlighted.

Parts of the work was presented at conferences or published in journals [51, 52, 54, 53, 49, 50]. Two interviews lead to articles in newspapers or magazines [36, 72]. It is planned that a thesis summary will be published in the TUHH Spektrum magazine in October 2008 [50].

Acknowledgement

I would like to thank my supervisor Prof. Dieter Gollmann for his continual support and especially for many week-ends of helpful discussions and reviews of my work. The design of the MASC-ST protocol was constructively reviewed by Tuomas Aura and in earlier stages by Vebjoern Moen. I would like to extend my gratitude and appreciation to them and all other unknown reviewers who commented my publications. The basic idea of the container monitoring application to detect damages was raised by Christian Thorge Schmidt (Datenaura). In constructive meetings with Pierre C. Sames (Germanischer Lloyd), Lars Kjällström (British Maritime Transport) and others, the idea of the container monitoring application was refined.

Additionally, I want to thank all the students I supervised throughout the last 4 years. Their personal and technical support helped me to proceed with my dissertation. Thanks to Ana-Luisa Alam Zamora, Sevkan Taskan, Jalal Alousi, Rosa Isabel Palero Monllor, Tobias Tiedgen, Henrik Tolk, Karim Manuel Heredia Gonzáles, Konrad Windszus, Mike Gerdes, Vanessa Taylor Sánchez (today: Pietsch-Taylor), Hannah K. Lee and especially to Harald Sauff (Co-auther of [54]). Many student thesises were helpful input to this thesis [80, 106, 96, 2, 1, 102, 94, 66, 103].

Einleitung

Abstrakt

MASC beschreibt eine Containerüberwachungsarchitektur, die Vertraulichkeit und Integrität des Informationsflusses sicherstellt. Ein Baustein ist das entworfene MASC-ST, ein dynamisches und authentifiziertes Schlüsseletablierungsprotokoll. Die Zugriffskontrolle wird vom MASC-IS mit einem delegierungsfähigen Ansatz geregelt. Die ganzheitliche Sicherheitsarchitektur ist speziell den Anforderungen des Logistikwesens angepasst.

Zusammenfassung

Nicht zuletzt seit in den 90er Jahren der Warschauer Pakt zerfallen ist, gibt es nur noch wenige Staaten in der Welt, die sich nicht an der globalen freien Marktwirtschaft beteiligen. Unter dem Stichwort Globalisierung verändern sich nahezu alle Wirtschaftsstandorte weltweit. Das Fundament dieser neuen Epoche ist der Containertransport. Wegen der extrem geringen Transportkosten ist es überhaupt erst möglich, dass beispielsweise kleine europäische Fabriken in direkter Konkurrenz zu chinesischen Großkonzernen stehen. Selbst bei der großen Rezession zu Beginn des 21. Jahrhunderts in Europa war die Transportbranche ein durchgehender Wachstumssektor. Kapitel 1 beleuchtet diese Situation und schildert, dass der Kostendruck zwar den Boom erst ermöglicht, aber leider viele Kollateralprobleme erzeugt.

Drei Probleme werden genauer beleuchtet:

1. **Transportqualität:** Die niedrigen Kosten führen zu vielen Transportschäden, deren zeitnahe Entdeckung nicht möglich ist.
2. **Terrorismusgefahr:** Nicht nur die USA fürchten die wenig kontrollierte Containereinfuhr als mögliche Schwachstelle im Grenzkontrollsystem.
3. **Heterogene Vertragssituationen:** Jede Zollunion hat andere Formulararten. Jedem Transport liegen individuelle Verträge zu Grunde. Die Übergabe von Dokumenten erfolgt oft noch händisch; die Informationsverarbeitung hat die Transportdokumentierung noch nicht wirklich erreicht.

Kapitel 2 schildert, wie die MASC-Anwendung diese drei Probleme lösen kann. MASC steht für *Monitoring and Security of Containers* und stellt eine Kombination aus Internetanwendung und mit Sensoren ausgestatteten Containern dar. Diese sind vernetzt und tauschen Informationen aus. Alle am

Transport beteiligten Parteien können den Status des Containers einsehen, pflegen ihre Vertragsdaten ins System ein, und die Zollbehörden erhalten automatisierten Zugriff. Gleichfalls können die Antiterror-Maßnahmen durch die MASC-Anwendung wesentlich effizienter gestaltet werden. Allerdings birgt diese neue Technologie Gefahren: Die Prozesse würden sich nicht nur für legitime Benutzer vereinfachen, sondern auch für Angreifer. Dies können Konkurrenten, Piraten, Terroristen oder Computerhacker sein. Es ist daher erforderlich, die Architektur schon zur Designzeit schwachstellenfrei zu erstellen.

Kapitel 3 wägt das Für und Wider ab und beschreibt die MASC-Architektur im Detail. Besonders schützenswert ist die Verbindung zwischen Container und Internetserver. Hierfür wurde mit dem MASC-SecureTunnel ein eigenes Protokoll entworfen, welche sich an aktuelle Standards anlehnt, jedoch die Randbedingungen dieser Anwendungen im Besonderen berücksichtigt. Das Kapitel 4 beschäftigt sich mit dieser Schnittstelle.

Eine Analyse der Zugriffskontrolle folgt in Kapitel 5. Hier wird der MASC-Informationsdienst beleuchtet, der auf besonders schlanke Weise alle Parteien zusammenführt, ohne nennenswerten Extraaufwand zu erzeugen. Die spezielle Anforderung an diesen Dienst ist die delegierte Weitergabe von Zugriffsrechten, mit der sich die Vertragspartner und ihre Vertragsabschlüsse eins zu eins abbilden lassen. Die weiteren Kapitel dieser Arbeit zeigen auf, welcher Aufwand darüber hinaus betrieben werden muss, um die Überlegungen dieser Arbeit in die Realität zu überführen. Hierbei wurden auch Ergebnisse erster Feldversuche berücksichtigt.

Teile dieser Arbeit wurde auf Konferenzen und in Magazinen veröffentlicht [51, 52, 54, 53, 49, 50]. Des Weiteren erschienen nach Interviews mit dem Handelsblatt und der Computer Zeitung Artikel über Containerüberwachung [36, 72] und in der Spektrum-Zeitschrift der TU Harburg wird aller Vorraussicht nach eine Zusammenfassung über das MASC-Projekt in der Oktober Ausgabe 2008 erscheinen [50].

1 Einführung – Status Quo im Containertransportgeschäft

Der internationale Containertransport hat einen oft unterschätzten Einfluss auf den Wirtschaftswandel. Insbesondere die großen Containerfrachter sind verantwortlich für die extrem geringen Kosten im weltweiten Güterverkehr. Der große Sparzwang und die starke Konkurrenz zollen ihren Tribut – immer mehr Parteien rücken von der Minimierung der Transportkosten ab und wünschen sich einen qualitativ höherwertigen Transport, der durch Schadensvermeidung und -erkennung zu einer Gesamtkostenreduzierung führt. Diese Einführung erläutert die zurzeit vernachlässigten Qualitätsmerkmale und zeigt die wünschenswerten Verbesserungen auf.

1.1 Transportketten im Containergeschäft

Zum Verständnis der komplexen Geschäftsabläufe im Containertransportwesen ist es wichtig, alle Rollen und Verantwortungsbereiche zu erläutern. Es existieren immer weniger Güter, die für nur ein Transportmittel geladen werden. Kombinierter Verkehr mit Hilfe von Containerboxen dominiert den heutigen Warenverkehr. Die beteiligten Transporteure haben nur minimale Informationen über den Transport: Sie bekommen einen Container und transportieren ihn auftragsgemäß. Schiffe, Züge und LKWs können die gleichen Container laden und verlängern die Reichweite des Transportbereiches. Es ist nicht mehr notwendig, dass jede Firma einen eigenen Umschlagsplatz im Hafen betreibt. Die Container werden weltweit positionsunabhängig von einem Punkt *A* zu Punkt einem *B* transportiert.

1.1.1 Die logistische Transportkette

Der **Shipper** ist der Auftraggeber des Transportes. Ihm gehört im Regelfall die Ware, die von *A* nach *B* transportiert werden soll. Er betraut eine **Spedition** (engl. *Forwarding Agency*) mit der Transportorganisation. Streng genommen hat eine Spedition keine eigenen Transportmittel, sondern beauftragt lediglich Fremdfirmen mit Teiltransporten. Moderne Speditionen haben oftmals jedoch eigene Transportmittel und sind somit Spediteur und **Frachtführer** (*Carrier*) in einem. Unabhängig von der eigenen Rolle ist es die Aufgabe der Spediteure, eine Route zusammenzustellen und geeignete Frachtführer (bei Containerschiffen wird auch von **Reedereien** gesprochen) zu finden, die den Container auf Teiltransporten befördern. Viele Speditionen sind jedoch auf Regionen spezialisiert und beauftragen ihrerseits fachkundige Subspeditionen mit dem Transport durch bestimmte Staaten wie China. Neben den organisierenden Speditionen und den befördernden Frachtführern gibt es obendrein die Schnittstellen

dazwischen. **Containerdepots** beherbergen die abgestellten Leercontainer vor und nach ihren Transporten. Auf **Containerterminals** werden Container von Transportmitteln umgeschlagen. Sollten sie für längere Zeit verweilen und auf einen Weitertransport warten müssen, werden sie auf **Containeryards** geparkt.

Etwas komplizierter wird die Situation dadurch, dass die Parteien oft nicht unterscheidbar sind. Speditionen sind häufig lokale Frachtführer, Containeryards meistens an Containerterminals angeschlossen. Reeder sind oftmals **Containerbesitzer**, von denen die Container für einen Transport geliehen werden. Genau genommen müsste darüber hinaus zwischen dem Schiffsbesitzer und dem Containerslotbesitzer auf Schiffen unterschieden werden. Zur Risikominimierung werden bei großen Containerfrachtern für die Laufzeit von mehreren Jahren Containerstellplätze (*Slots*) an andere Reedereien verkauft – diese können die Stellplätze dann komplett selber verwalten. Um eine Kostenreduzierung erreichen zu können, gibt es ferner die Möglichkeit, sich mit mehreren Shippern einen Container zu teilen. Im Gegensatz zu einem FCL-Transport („Full Container Load“), bei dem ein Shipper einen ganzen Container bucht, teilen sich mehrere Shipper beim LCL-Transport („Less than Container Load“) einen **Consolidation-Container**.

Diese Arbeit definiert zwei Zusammenfassungen von Parteien. Die **Transporteure** sind dabei alle Parteien, die physisch im Containertransport involviert sind. Es handelt sich dabei um die Transportführer, die Reedereien und die Containerstellplätze (Containerterminal, Containeryard und Containerdepot). Als **Logistiker** (*LSP – Logistic Service Provider*) werden hingegen alle Parteien außer Versicherer und Shipper definiert und somit die Menge der Transporteure und Spediteure.

Da Transportschäden häufig auftreten, sind die meisten Transporte versichert, wobei der Spediteur den Vertrag mit dem **Versicherer** abschließt (alternativ kann auch der Shipper den Vertragsabschluss übernehmen). Über die Transportversicherung hinaus sind alle Logistiker gegen eigenes Fehlverhalten versichert. Genaue Klauseln regeln, an welches Verhalten und welche Bedingungen ein Frachtführer gebunden ist. Allerdings ist der Shipper nur am Anfang und gegebenenfalls am Ende[1] des Transportes involviert. Dazwischen übergeben die verschiedenen Transporteure den Container untereinander. Dieses genau geregelte Übergabeprozedere wird *Interchange* genannt.

1.1.2 Der Interchange

Wenn ein Frachtführer einen Leercontainer bei einem Containerdepot abholt, überprüft er diesen von innen und außen. Zusätzlich muss dieser besenrein übergeben werden und die Identifikationsnummern müssen mit den Papieren übereinstimmen. Der Container darf keine groben Beschädigungen aufweisen. Wenn alle Bedingungen erfüllt sind und der Frachtführer den Container annimmt, bestätigt er seine Unversehrtheit. Der Container wird anschließend beim Shipper beladen, geschlossen und versiegelt. Bis zum Empfänger darf der Container nicht mehr geöffnet werden (Ausnahmen bilden lediglich Zoll-Untersuchungen). Daher können alle folgenden Interchanges auch nicht den Zustand der Ware

[1] Der Transportempfänger wird manchmal vom Shipper unterschieden und **Consignee** genannt.

überprüfen. Der Container wird lediglich einer äußeren Sichtkontrolle unterzogen und die Unversehrtheit des Siegels wird geprüft.

Bei einem modernen Containerterminal läuft das Entladen voll automatisiert ab. Kräne greifen den Container, um ihn vom Schiff abzuheben und einer zweiten Kraneinheit zu übergeben. Diese sorgt dafür, dass die richtigen Verteilerkräne den Container übernehmen können, während der Entladekran bereits den nächsten Container abhebt. Allerdings führen die Containerterminals sehr wohl einen Interchange durch. Dafür ist es eigens nötig, dass Personen auf den Kränen sitzen. Bevor der Container dem Verteilerkran übergeben wird, erhält der Kranführer über entsprechend angeordnete Kameras Sicht auf alle Seiten des Containers. Nach erfolgter Sichtprüfung muss er per Knopfdruck die Unversehrtheit bestätigen. Erst jetzt darf der Container dem Verteilersystem übergeben werden. Somit ist das Containerterminal haftbar für Schäden, die durch Fehlverhalten geschehen. Erst wenn der folgende Transporteur den Container (nach einem Interchange) annimmt, endet die Haftung des Zulieferers.

1.1.3 Die Vertragssituation

Da die Transporte über Ländergrenzen hinausgehen und auf hoher See kein hoheitliches Staatensystem für Strafverfolgung zuständig ist, sind die Verträge zwischen den Logistikern von besonderer Bedeutung. Diese müssen alle Details regeln und unter anderem definieren, an welches Länderrecht welche Parteien gebunden sind. Als interessantes Beispiel für derartige Verquickungen sei folgender Fall zusammengefasst, von dem die englische Juristen-Seite Online-DMC berichtet [26].

Die „Hyundai-Fortune“

Die Firma Uni-Fruitveg importiert Hami-Melonen von China nach Singapur. Bei Hami-Melonen handelt es sich um eine Melonen-Art, die nur im Kreis Shanshan in der Region von Hami (China) angebaut wird. Uni-Fruitveg transportiert seine Hami-Melonen in einem 40-Fuß-Kühlcontainer (*reefer container*). Für die Strecke von Hong-Kong nach Singapur wurde die Firma „EMF International“ mit dem Frachttransport beauftragt. Bei einem Transport von 1 473 Kartons kamen 1 232 stark beschädigt an. Fruitveg konnte nachweisen, dass die geforderten 3°C nicht eingehalten wurden und die Temperatur bis zu 25°C gestiegen war. Der entstandene Schaden wurde auf knapp $ 8 400 beziffert.

Die Ladungspapiere (**Bill-of-Lading**) regelten, dass alle rechtlichen Schritte in Korea durchzuführen waren. Die Firma Fruitveg ging jedoch vor ein Gericht in Singapur, um eine Verhandlung in Singapur zu erwirken. Weil die Schiffsbesitzer Singapur nicht als Verhandlungsort akzeptierten, ging es bei diesem Fall lediglich um die zuständige Gerichtsbarkeit.

Das Gericht nannte zehn Gründe dafür, dass eine Gerichtsbarkeit in Singapur zulässig sei. Unter anderem sah der Richter nicht ein, warum sich ein koreanisches Gericht besser eigne, da das Schiff in Panama geflaggt sei, der Shipper aus Hongkong komme, der Kühlcontainer aus Shenzen stamme und die Schiffbesitzer ein Büro in Singapur unterhalten. Angesichts der geringen Schadensgröße wäre der Aufwand, alles vor ein koreanisches Gericht zu tragen, nebst der Übersetzung der Unterlagen bei dieser

Sachlage nicht gerechtfertigt. Dem Reeder wurde zudem angelastet, seine vermeintlichen Beweise dem Gericht vorenthalten zu wollen, weil er mutmaßlich auf das Greifen der koreanischen Verjährungsfrist spekuliere.

Dieser Fall zeigt exemplarisch, wie unsauber und schlecht differenziert die Rechtslage ist. Daher spielen die Versicherungen eine gewichtige Rolle. Wäre der Shipper adäquat versichert gewesen, hätte sich die Versicherung um Schadensregulierung bemüht und selber entscheiden können, ob sie für $ 8 400 plus Verwaltungskosten vor ein Gericht zieht.

Der Vertragsbaum

Durch das gewachsene Logistikgeschäft, dessen weltweiter Vernetzung und den Containern als interoperable Transportbehältnisse sind bei heutigen Containertransporten oftmals sehr viele Parteien involviert. Dem Shipper ist neben Versicherer und Spediteur höchstens der die Lieferung in Empfang nehmende Frachtführer bekannt. Die restlichen Parteien der Transportkette bleiben ihm meist verborgen. Aber selbst der Spediteur kennt oftmals nur die direkt Beauftragten. Transportführer geben mitunter Transporte bei veränderter Auftragslage an andere Firmen weiter oder teilen ihre Teilstrecke in Unterteilstrecken auf und vergeben diese weiter an Untertransportführer. Subspeditionen organisieren ihren Transport im Regelfall, ohne ihre Auftraggeber von Vertragsabschlüssen in Kenntnis zu setzen. Auf diese Weise entsteht eine Baumstruktur, bei der nur benachbarte Knoten voneinander wissen.

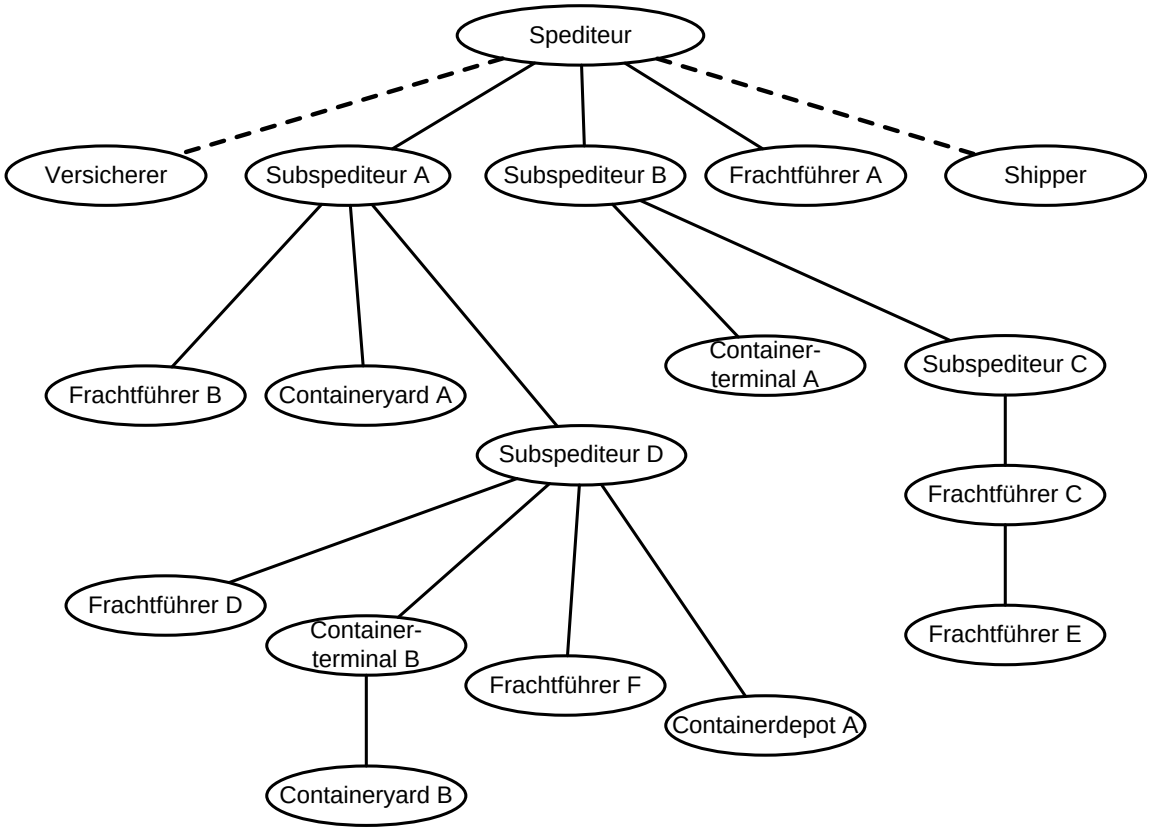


Abbildung 1.1: Beispiel Vertragsbaum

Es gibt wenige Geschäftsfelder, die eine vergleichbare Heterogenität und Vielzahl transitiver Vertragsweiterführungen aufweisen wie das Containertransportwesen. Abbildung 1.1 zeigt anhand eines Beispiels, wie dies in der Praxis aussieht. Jede Kante bezeichnet eine Vertragsrelation, während die jeweiligen Knoten am Ende einer Kante die Vertragspartner symbolisieren. Jede durchgezogene Linie bezieht sich auf einen Vertrag zwischen LSPs – diese Verträge haben also direkten Einfluss auf die Transportkette. Die gestrichelten Kanten sind die Verträge zwischen Spediteur und Shipper einerseits und Versicherung andererseits.

Exemplarisch wird in Abbildung 1.1 ein Vertragsbaum mit einem Spediteur, vier Subspediteuren, sechs Frachtführern, einem Containerdepot, zwei Containerterminals und zwei Containeryards beschrieben. Es sind 16 Logistiker involviert, zuzüglich Shipper und Versicherer. Diese Zahlen sind bei einem Transport von Ost-Asien nach Europa durchaus realistisch.

Neben den längerfristig geplanten Parteien in der Transportkette können auch kurzfristig Änderungen an dem Vertragsbaum auftreten und somit neue Parteien zeitnah hinzukommen. Sind auf einem Containerschiff noch Containerstellplätze frei oder wird ein Schiff durch ein anderes mit einer abweichenden Kapazität ersetzt, kann es sein, dass Ladung auf Schiffe möglicherweise anderer Reedereien umgelagert wird. In solchen Fällen erfolgt umgehend eine Baumerweiterung. Dies ist eine weitere Verkomplizierung dieser Vertragssituation. Wegen dieser hohen Interoperabilität ist der Wettbewerb entsprechend groß, mit Auswirkungen auf die Transportkosten.

1.1.4 Transportkosten

Gemäß einer Studie der österreichischen Arbeiterkammer zeigt Tabelle 1.1, wie teuer der reine Transport einer Tonne beliebiger Fracht in einem 40-Fuß-Container pro Kilometer ist [39]. Die Zahlen basieren auf Informationen des Österreichischen Instituts für Wirtschaftsforschung.

Verkehrsmittel	€/Tonnen-Kilometer
LKW	€ 0,036
Bahn	€ 0,025
Binnenschiff	€ 0,015
Hochseeschiff (Linie)	€ 0,001
Luftfahrt	€ 0,102

Tabelle 1.1: Frachtsätze in Österreich (Quelle: [39], Tabelle 41)

Die Frachtsätze Österreichs sind trotz der Alpen eine gute Annäherung für alle Staaten in Zentraleuropa. Zweifelsohne ist der Übersee-Containertransport die günstigste Transportart. Daher verwundert es nicht, dass über 90% des Welthandels von der internationalen Schiffsindustrie befördert werden [8]. Gleichzeitig werden ungefähr 95% aller Waren in Containern transportiert [23].

Die „Sunday Times“ berichtet in ihrer Ausgabe vom 30. April 2006 (Abschnitt 3 „Economy“, Seite 12), dass die typischen Kosten für einen Transport eines 20-Fuß-Containers von Asien nach Europa genauso hoch sind wie für einen einzelnen Passagier auf der gleichen Strecke per Flugzeug.

Transportgut	Einheit	typischer Ladenpreis	Transportkosten
Fernsehgerät	1 Stück	$ 700	$ 10,00
DVD-Rekorder	1 Stück	$ 200	$ 1,50
Staubsauger	1 Stück	$ 150	$ 1,00
Whisky	1 Flasche	$ 50	$ 0,50
Kaffee	1 kg	$ 15	$ 0,15
Kekse	1 Dose	$ 3	$ 0,05
Bier	1 Dose	$ 1	$ 0,01

Tabelle 1.2: Typische Übersee-Transportkosten [8]

Tabelle 1.2 zeigt die güterabhängigen typischen Transportkosten, die für bestimmte Waren bei einem Transport von Ost-Asien nach Europa beziehungsweise in die USA anfallen. Es werden immer größere Containerschiffe gebaut, um die Kosten pro Container weiter senken zu können. Maßnahmen zur Verbesserung der Transportbedingungen bleiben dabei jedoch auf der Strecke.

1.1.5 Der Schadensfall

Es werden zwei Arten von Transportschäden unterschieden. Sichtbare Schäden am Container sind Schadensfälle, die bei Interchanges entdeckt werden können. Hierzu zählen gewaltsam aufgebrochene, ausgebrannte und beschädigte Container. Diese Schäden bleiben längstens bis zum nächsten Interchange unbemerkt. Wenn der Schadensfall beim Beladen eines Containerschiffs eintritt, dauert es schlimmstenfalls drei bis vier Wochen, bis die Ladung gelöscht wird – dies sollte aber schon als die längste anzunehmende Wartezeit angesehen werden.

Zu diesen erkennbaren Schäden liefert die USA-Today interessante Zahlen. In einem Artikel [18] werden Experten zitiert, die konstatieren, dass in den Jahren 2004 und 2005 mehr als 600 Piratenangriffe mit dem Ziel des Warendiebstahls bekannt wurden. Der gleiche Artikel berichtet, dass jährlich 2 000 bis 10 000 Container von Schiffen herunterfallen und verloren gehen.

Die meisten Schäden zählen allerdings zur zweiten Kategorie und gehen nicht mit äußerer Beschädigung einher. Sie sind vielfach auf schlechte Transportbedingungen zurückzuführen. Gestürzte Container oder falsche Umgebungsbedingungen im Container sind hier zu nennen. Außerdem versuchen Kriminelle beim Containerdiebstahl zunehmend ihre Spuren zu verwischen: Neuverplombte Container werden mitunter nicht als aufgebrochen entdeckt. Das größte Problem dabei ist, dass diese Schäden oftmals erst am Ende einer Transportkette festgestellt werden [73] und jeder der beteiligten Frachtführer seine Schuld ausschließt. Die Aufgabe der Versicherungen, den Haftenden auszumachen, ist meistens schwierig. Oftmals sind die Recherchearbeiten so aufwändig, dass es sich nicht lohnt, nach einem Schuldigen zu suchen, sodass die Versicherung für den Schaden aufkommt.

Da zunehmend Schäden auftreten, verzeichnen die Transportversicherungen im Containerverkehr steigende Beiträge. Im Besonderen ist Kriminalität im Containertransport eine hohe Belastung für die Versicherer. Bei einer Schadenverhütungstagung des Fachausschusses Transport beim Gesamtverband der Deutschen Versicherungswirtschaft (GDV) hat Kapitän AG Thomas Ziehn von der Gerling Versicherung auf verschiedene Fälle des Warendiebstahls aufmerksam gemacht [108]. Seine angeführten Beispiele beziehen sich meist auf Diebstahl der Ware in Bereichen, wo der Container unbeaufsichtigt gelassen wird. In sämtlichen dokumentierten Fällen wurde der Diebstahl mit einfachen Mitteln begangen: Die Diebe hatten genügend Zeit und waren ungestört. Die Fälle zeigten aber auch, dass teilweise viel teure Recherchearbeit notwendig ist, die ebenfalls die Versicherungssummen erhöht. Darüber hinaus bleibt zu vermuten, dass etliche Fälle unaufgeklärt bleiben, da die von Ziehn beschriebenen Recherchemethoden unzureichend für aufwändigere Betrügereien scheinen.

Ebenfalls von Interesse für die Versicherer ist das Temperatur-Luftfeuchte-Verhältnis im Container. So entstehen viele Schäden durch schlechte Transportbedingungen oder unsachgemäße Verpackung. Frau Dr. Renate Scharnow schreibt in einem Fachvortrag: „Der Transporteur muss sich also völlig darauf verlassen, dass der Ablader eine für den Transport geeignete Qualität liefert und kein verdeckter Mangel, zum Beispiel in Form eines zu hohen Wassergehalts, vorliegt“ [81]). Diese Bedingungen sind im Besonderen für hygroskopische Waren wie „Kakao, Kaffee, Hirse, Backobst, Sago, Pfeffer, Milchpulver, Felle, Textilien, Rattanmöbel, Jute, indische Kunstgewerbeartikel“ [81] von Bedeutung. Im Besonderen ist Schweißwasserbildung am Container oder der Ladung direkt für Ladungsschäden verantwortlich. Ihre Untersuchungen zeigen, wann Schweißwasser auftritt, welche genauen Temperaturübergänge notwendig sind und welche Randbedingungen gelten müssen. In ihrem Vortrag beschreibt Frau Dr. Scharnow weiter, wie Feuchtigkeit hygroskopische Waren verdorben hat und wie durch Untersuchungen nachgewiesen werden konnte, dass die Ware schon zu feucht eingeliefert wurde (weitere Schadenbeispiele siehe [82]).

Jede Ladung hat natürlich andere Anforderungen. So müsste ein Interchange genau genommen ladungsspezifisch ablaufen, um eine höhere Aussagekraft zu erhalten. Bananen müssen zum Beispiel mit einer konstanten Temperatur zwischen 12,5 und 13,5°C transportiert werden. Bei höheren Temperaturen reifen die Bananen beim Transport und kommen braun beim Kunden an. Bei niedrigeren Temperaturen bleiben die Bananen grün und reifen nicht mehr. Als weiteres Beispiel seien Blumenzwiebeln genannt. Im Besonderen sind Tulpenzwiebeln sehr anfällig gegen Schimmelpilzbefall (Fusarium und Penicillium). Die US-amerikanische Agrarbehörde beschreibt, welche Transportbedingungen gelten müssen und wie Schäden entdeckt werden können [24]. Wenn die Ethylen-Konzentration größer als 0,1 ppm sei, seien die Zwiebeln mit ziemlicher Sicherheit von Fusarien befallen. Um die Penicillium-Ausbreitung zu bekämpfen, seien eine hohe Ventilation und eine Luftfeuchte von 90% nötig. Tulpenzwiebeln seien bei einer konstanten Temperatur von 17°C zu verschiffen und unter einer Ventilationsrate von 150 m^3h^{-1}, wenn das Ausbrechen dieser Pilze verhindert werden solle.

De facto ist es nicht möglich, mit den aktuellen Abläufen die Unversehrtheit der Ware beim Interchange zu überprüfen. Das Öffnen jedes Containers wiederum ist aus rechtlichen und zeitlichen Grün-

den ausgeschlossen. Neue Ideen und Technologien sind nötig, um die Transportqualität nachhaltig zu erhöhen.

1.1.6 Internationale Überprüfbarkeit

Ein schwieriges Kapitel im internationalen Transportwesen ist die Rechtslage auf hoher See. Das Hoheitsgebiet von Staaten reicht bis zu zwölf Seemeilen ins Meer hinein; alles, was weiter entfernt ist, wird als internationales Gewässer bezeichnet. Für diesen Bereich gibt es keine Justizgewalt und auch keine Exekutive. Einzig die *International Maritime Organization* (IMO) als Quasi-Legislative definiert internationale Regeln und Gesetze. Die IMO-Mitgliedstaaten übernehmen deren Vorschläge in nationales Recht und zwingen einheimische Schiffe, sich an diese Regeln zu halten. Durch diese Bindung von Schiffen an Länder wird die problematische Rechtslage in die Hoheitsgebiete von Staaten verlagert. So gilt für alle Schiffe aus IMO-Mitgliedstaaten ein weltweites Gesetz, an das sie im Heimatland rechtlich gebunden sind. Allerdings gibt es immer wieder Schiffe, die in den wenigen Ländern geflaggt sind, die nicht die IMO-Regeln adoptiert haben. Dies führt zu Problemen, wenn diese Schiffe in Vorfälle verwickelt sind. Allerdings geht die IMO in sicherheitsrelevanten Punkten etwas weiter, da die Zahl der Schiffe zunimmt, die sich der IMO zu entziehen versuchen. Schiffe, die Häfen von IMO-Mitgliedsstaaten anlaufen, müssen sich verpflichten, IMO-Sicherheitsstandards zu genügen. Auf diese Weise können sich international agierende Schiffe nicht allen IMO-Regeln entziehen.

Jedes Schiff muss gemäß der IMO-Richtlinien zu einer regelmäßigen Überprüfung. Vergleichbar mit dem TÜV bei PKWs, existieren globale **Verifizierungsunternehmen**, die die Konformität von Schiffen, Hafenanlagen, Containern und Bohrinseln überprüfen. Diese Schiffsverifizierer, wie der „Germanische Lloyd“ oder „Bureau Veritas“, sind Vermittler zwischen dem Gesetzgeber, der als verlängerter Arm der IMO deren Regeln vorschreibt, und den Reedern, die ihre Schiffe überprüft haben müssen. Die Verifizierer kontrollieren überdies die Konformität von Containern und Hafenanlagen.

Da sich die Verifizierer im offenen Wettbewerb befinden, ist ihnen auf der einen Seite das Kundenverhältnis sehr wichtig. Auf der anderen Seite sind sie verpflichtet, die IMO-Regularien korrekt zu überprüfen. Kommt nach einem Schiffsunglück heraus, dass die Verifizierer ihre Arbeit verfehlt haben, können diese zur Rechenschaft gezogen werden. In letzter Instanz führt dies zum Entzug des Verifiziererstatus, was den Verlust der Existenzgrundlage bedeutet. Bedingt durch diese Situation, dass der Verifizierer berechtigtes Interesse daran hat, zu beiden Seiten ein gutes Verhältnis zu pflegen, sind die Verifizierer vertrauenswürdige Dritte, die als Vermittler zwischen staatlicher Hoheitsgewalt und der Industrie positioniert sind.

1.1.7 Sarbanes-Oxley

Der Sarbanes-Oxley-Act ist ein US-amerikanisches Gesetz, welches von dem Demokraten Paul S. Sarbanes und dem Republikaner Michael Oxley entworfen wurde. Es dient der verbindlichen Regelung der Unternehmensberichterstattung von börsennotierten Firmen. Diese Maßnahmen zum Anlegerschutz

waren notwendig geworden, nachdem die Firmen Enron und Worldcom in Bilanzskandale verwickelt waren, die durch verschleierte Quartalsberichte hervorgerufen wurden.

Dieses Gesetz soll die Richtigkeit und Verlässlichkeit der offengelegten Finanzdaten nachhaltig sicherstellen. Alle an US-Börsen notierten Firmen müssen sich Sarbanes-Oxley-konform verhalten. Das Gesetz liegt einem aufsichtsrechtlichen System für Wirtschaftsprüfgesellschaften zu Grunde, die Berichte dieser Unternehmen überprüfen. Dieses basiert auf Richtlinien, die Risiken und Sicherheiten der vorgelegten Zahlen einschätzen sollen.

Wie das nachfolgende Beispiel zeigt, ist die Transportsicherheit nicht nur für Frachtführer und Frachtversicherer ein wichtiges (und damit überprüfungswürdiges) Unternehmensfeld, sondern auch für im- oder exportlastige Firmen.

4700 Mazdas fast gesunken Der Automobilhersteller Mazda hat im Jahr 2006 erfahren, welches Risiko ein Schiffstransport für ihn bedeuten kann. Wie unter anderem der „Motor Vehicile Lemon Law"-spezialisierte Weblog berichtete, ging ein mit 4 700 Mazda-Automobilen beladenes Schiff in 60%ige Schräglage, nachdem der wasserbasierte Ballastausgleich versagte [17]. In Folge dessen konnte Mazda keinen der Neuwagen mehr wie geplant verkaufen. Mazda hatte allerdings Glück im Unglück, weil die Autos geborgen, Teile wiederverwertet und einige Autos auch repariert werden konnten. Die genaue Schadensgröße wurde von Mazda nicht beziffert.

Mazda verkaufte laut der Frankfurter Allgemeinen Zeitung (unter Berufung auf *Thomson Financial Datastream*) im Jahr 2005 1,14 Millionen PKW. Der Verlust von 4 700 Autos ist bei Kenntnis dieser Zahlen vermutlich kein großes Betriebsrisiko. Anders kann dies bei kleineren oder finanziell angeschlagenen Autoherstellern sein. Zurzeit werden Transportrisiken nicht adäquat von den Sarbanes-Oxley-zertifizierten Wirtschaftsprüfern berücksichtigt. Bei einer höheren Sensibilisierung für das Thema kann dies in der Zukunft aber durchaus passieren. Unabhängig von der gängigen Praxis wirken sich qualitätssteigernde Maßnahmen in jedem Fall positiv auf die Risikobetrachtungen der Firmen aus und senken somit die Sekundärkosten[2].

1.1.8 Bewertung der aktuellen Situation im Containertransportwesen

Der gemischte Containertransport ist in erster Linie kostengünstig. Es gibt zwar Ansätze zur Einführung qualitätssteigernder Maßnahmen in einigen Teilbereichen, jedoch umfassen diese nicht die gesamte Transportkette. Besonders der Schiffscontainertransport ist so auf Kostenminimierung ausgelegt, dass ein einzelner Shipper nicht in der Lage ist, Maßnahmen wie eine Transportüberwachung zu ordern – es wird einfach nicht angeboten. Besonders bei wertvoller Ware wären Mehrkosten pro Containertransport leicht zu verschmerzen, wenn ein effektiver Diebstahlschutz, eine Temperatur- oder Feuchtigkeitsüberwachung oder ein Sofortwarnsystem für den Transport bestellt werden könnte.

[2] Die Sekundärkosten sind im Gegensatz zu den reinen Transportkosten (Primärkosten) die Kosten, die von Versicherungssummen, Verwaltungsaufwand und Risikoabschätzungen herrühren.

Ein weiteres Problem stellt die komplexe Situation der Vertragspartner dar. Kleine, günstige Unterfrachtführer schließen ihre Verträge heute noch oft mit handschriftlich ergänzten Fax-Kopien ab. Wenn aber Verbesserungsmaßnahmen eingeführt werden sollen, stellt sich das Problem, dass alle Logistiker in einer Transportkette dabei involviert sein müssten. Das ist wohl auch einer der Hauptgründe, dass die Situation seit Jahren stagniert.

1.2 Auswirkungen der Terroranschläge des 11. September 2001

Die Terroranschläge des 11. September 2001 haben nicht nur weltweites Bestürzen ausgelöst. Neben umstrittenen Militäraktionen wurden eine Menge neuer Bestimmungen und Gesetze eingebracht, die unter anderem massive Auswirkungen auf den internationalen Containerverkehr haben. Besonders die USA sahen sich in einer Vorreiterrolle und gründeten die neue Behörde „Homeland Security“. Dieser Behörde ist die „U.S. Border and Customs“ unterstellt, die für den Großteil der Änderungen verantwortlich zeichnet. Neben den amerikanischen Behörden hat ebenfalls die IMO reagiert und für ihren Einflussbereich ein großes Änderungspaket mit auf den Weg gegeben.

Eine interessante Berechnung ist in dem Papier „Container Security“ [23] zu finden; hier werden verschiedene offizielle Quellen herangezogen und die Kosten für das Schließen eines Hafens auf Grund eines Terroranschlags abgeschätzt. EINE BILLION DOLLAR solle dieses Unterfangen gesamtwirtschaftlich kosten. Das zwölftägige Schließen eines Hafens mit gleichzeitiger Suche nach einer nicht detonierten Bombe verschlinge davon alleine 58 Milliarden Dollar. Dieses macht deutlich, weshalb diese neuen Maßnahmen so schnell und geballt verabschiedet wurden. Im Folgenden werden die neuen Regularien erläutert und eine Kostenabschätzung des vermuteten Mehraufwandes durchgeführt.

1.2.1 ISPS

Der ISPS-Code steht für „International Ship and Port Facility Security Code“ und wurde im Jahr 2002 von der IMO publiziert. Sie gilt als Erweiterung zu dem SOLAS-Vertrag („Safety Of Life At Sea Convention“), der die Menschen an Bord schützen soll. Seit Juli 2004 gilt das zweiteilige legislative Dokument für internationale Schiffe genauso wie für die Häfen, in denen diese Schiffe anlegen. Der erste Teil dieses Dokumentes definiert die Sicherheitsanforderungen an Schiffe und Häfen, wobei sich der zweite Teil darauf beschränkt vorzuschreiben, wie diese Anforderungen in der Praxis anzuwenden sind. Die umgesetzten Grundprinzipien des ISPS-Codes sind die rechtzeitige Erkennung der Sicherheitsrisiken, eine Einführung von deren Messmethoden und die Realisierung von Rollen mit entsprechenden Verantwortungsbereichen bezüglich Schiffen, Häfen und deren Einbettung in Regierungen und Behörden. [34, 43, 13]

Der ISPS-Code wurde nach den Terroranschlägen des 11. September 2001 initiiert, nachdem mit Flugzeugen Verkehrsmittel für Terroranschläge verwendet wurden. Die unmittelbare Nähe von Pier 13 zur Wallstreet in Manhattan machte vielen Menschen deutlich, dass von Schiffen durchaus eine Gefahr

ausgehen kann. Es genügen Bomben mit einer Reichweite von 100 Metern, um in vielen bedeutenden Hafenstädten einen immensen Schaden anzurichten. ABC-Waffen können die Wirkung dramatisch verstärken. Darüber hinaus haben Schiffe selber eine nicht zu vernachlässigende Kraft und können als Waffe gegen hafennahe Ziele eingesetzt werden, weswegen der ISPS-Code nicht nur für die Schiffe gilt, sondern ebenso für deren Schnittstelle zum Festland.

Ebenfalls können Schiffe Ziele eines Angriffs sein oder von bewaffneten Piraten überfallen und unter deren Kontrolle gebracht werden. Daher gilt der ISPS-Code für alle internationalen Frachtschiffe mit einer Größe über 500 BRZ[3], alle internationalen Fahrgastschiffe und mobile Offshore-Bohrinseln.

Für jedes dieser Schiffe muss die verantwortliche Reederei eine Risikobewertung durchführen (Ship Security Assessments, SSA) [14], einen Plan zur Gefahrenabwehr anfertigen (Ship Security Plan, SSP) und die verantwortlichen Personen bestimmen. So gibt es für jedes Schiff einen SSO (Ship Security Officer) und für die Reederei zusätzlich einen CSO (Company Security Officer). Der Reeder muss zudem für die Mitarbeiter Schulungen, Ausbildungen und Übungen anbieten. Darüber hinaus gelten für alle Aktivitäten Aufzeichnungspflichten. Sind alle Anforderungen erfüllt und überprüft worden, bekommt das Schiff ein ISSC (International Ship Security Certificate).

In jedem Hafen existiert seit Einführung des ISPS-Codes eine Port-Authority. Dabei handelt es sich um eine Behörde, die die Richtlinien des ISPS-Code überwacht und durchsetzt. Als eine Maßnahme führt diese Behörde eine Sicherheitsanalyse aller Hafenanlagen mit internationalem Verkehr durch (Port Facility Security Assessments, PFSA). Auf der Grundlage einer Risikoanalyse erstellt sie einen Gefahrenabwehrplan (Port Facility Security Plan, PFSP). Die Verantwortlichen einzelner Risikobereiche müssen Maßnahmen durchführen, die zu einer Minimierung des Risikos führen. Als Verantwortlicher für die Erstellung der Gefahrenabwehrpläne wurde der PFSO (Port Facility Security Officer) eingeführt. Die Port-Authority muss ebenfalls Schulungen und Übungen für ihre Mitarbeiter anbieten.

Seit dem 1. Juli 2002 ist es Personen ohne Erlaubnis der Port-Authority nicht mehr gestattet, internationale Schiffe zu betreten. Es muss vorher immer ein Antrag gestellt werden und anhand einer Besucherliste darf der Eintritt nur angemeldeten Personen erlaubt werden. Lediglich bestimmten Behörden, Lotsen und Notdiensten wird unter bestimmten Voraussetzungen Zugang zum Schiff gewährt. [42]

Die Konformität zum ISPS-Code bescheinigenden Zertifikate können nur von einer RSO (*Recognized Security Organisation*) ausgestellt werden. Die RSOs müssen speziell geschulte Betriebe sein, denen vom Gesetzgeber und von der Industrie gleichermaßen vertraut wird. Da die Verifizierer bereits diese vertrauenswürdige Rolle einnehmen, ist es nur konsequent, dass diese Institutionen auch die ISPS-Sicherheitsrichtlinien überprüfen. Eine Verifizierungsgesellschaft kann zur RSO werden, wenn sich ihre Mitarbeiter speziellen IMO-Schulungen unterziehen.

[3] Die Bruttoraumzahl ist Volumenmaß – ehemals Bruttoregistertonne

1.2.2 US-amerikanische Initiativen der Containersicherheit

C-TPAT

Parallel zu den Maßnahmen der IMO gibt es verschiedene weitere Bemühungen, den internationalen Frachtverkehr abzusichern. Die US-amerikanische Zollbehörde (US-CBP) hat ein Programm aufgelegt, das die Transportkette gegenüber Terrorismus absichern soll. Die C-TPAT (Customs Trade Partnership against Terrorism) wurde im November 2001 begründet und regelt die US-amerikanischen Einfuhrbestimmungen für Waren. Die Teilnahme an dem Programm ist freiwillig. Als Gegenleistung werden die Einfuhrbestimmungen gelockert.

Jeder Partner hat der US-CBP dabei mehrere Zugeständnisse zu machen. Je nach Rolle des Partnerbetriebs in der internationalen Transportkette müssen die Arbeitsprozesse um feste Sicherheitsprogramme erweitert werden. Dabei ist dies nicht auf die Bereiche beschränkt, die mit dem Beladen und dem Verpacken der Ware in Zusammenhang stehen. Zusätzlich müssen die Angestellten in definierten Schlüsselpositionen vorgeschriebene C-TPAT-Trainingsprogramme besuchen und in speziellen Bereichen des Betriebs eine physische Zugangskontrolle installiert werden. Ferner muss jeder Partner der US-CBP erlauben, stichprobenhafte Tests bei den Betrieben vor Ort durchzuführen.

Als Gegenleistung erhält jeder Partner Vorteile bei der Warenabfertigung auf dem amerikanischen Kontinent. So wurde eine „GreenLane“ definiert, bei der die Ware nur stichprobenartig überprüft wird. Container von C-TPAT-Partnern werden über diese „GreenLane“ eingeführt, alle anderen Container in eine Warteschlange eingereiht und einzeln überprüft. Die Zeitersparnis bei C-TPAT-Nutzern kann auf diese Weise bei bis zu mehreren Tagen liegen. [99]

Zurzeit können nur amerikanische Unternehmen der C-TPAT beitreten. Zulieferbetriebe sind aber auch zunehmend in Europa betroffen.

24-Hour-Advance-Vessel-Manifest-Rule

Die Zollbehörde der USA hat im Jahr 2002 ein Einfuhrgesetz erlassen, dass Container über den Seeweg nur noch eingeführt werden dürfen, wenn deren Frachtpapiere vor der BELADUNG des Schiffes außerhalb der USA der amerikanischen Zollbehörde zugänglich gemacht werden. Das AMS („Automated Manifest System“) definiert dabei die Vorgehensweise und gilt für sämtliche Transportmittel, die für den Transport der Container in die USA verwendet werden. Für den Seeverkehr gilt, dass ein Schiff nur einlaufen darf, wenn sämtliche Frachtpapiere in elektronischer Form mindestens 24 Stunden vor der Beladung bei der US-Zollbehörde eingetroffen sind (für die anderen Transportmittel verhält es sich analog). Dies trifft im Übrigen auch für FROB (Foreign Cargo Remaining on Board) zu, also für Container, die nur in einem US-Hafen zur „Durchreise“ eintreffen und für einen anderen Zielhafen bestimmt sind. In Anhang E.1 sind alle Daten aufgeführt, die ein Manifest für einen Container enthalten muss.

Container Security Initiative

Im Rahmen der CSI (Container Security Initiative) werden die AMS-Daten eines jeden Containers automatisiert ausgewertet und ein statischer (Terrorismus-)Risikowert ermittelt. Als zu riskant eingestufte Container müssen noch vor dem Beladen im Ausfuhrhafen bereits mit einem Container-Röntgengerät durchleuchtet und wie Handgepäck bei der Flugzeugabfertigung überprüft werden (wenn die Häfen kein Gerät besitzen, wird dieses manuell durchgeführt) [98]. Die Devise der CSI lautet, die amerikanischen Häfen als letzte Verteidigungslinie zu etablieren und dementsprechend die Häfen außerhalb der USA als davor liegende [97]. Mittlerweile haben 20 internationale Häfen diese teuren Röntgen-Geräte angeschafft. Darüber hinaus fordert die CSI den Einsatz von fortschrittlichen intelligenten Containern. Die Einführung des elektronischen Siegels ist die erste Reaktion auf diese Forderung.

1.2.3 Kosten durch Terrorabwehrmaßnahmen

(in TEU)	Insgesamt	Export	Import
Führende 25 US-Häfen	23 493 957	7 887 924	15 606 033
Alle US-Häfen	23 850 523	8 045 045	15 805 478
Prozentualer Anteil der Top 25	98,5%	98,0%	98,7%

Tabelle 1.3: US-Container-Handel in TEU [22]

Laut [22] werden jährlich fast (äquivalente) 24 Millionen Standardcontainer (TEU – siehe Anhang B.1) in die USA eingeführt (siehe Tabelle 1.3). Neben den beim Durchleuchten anfallenden Kosten benötigt jeder Hafen Röntgen- und Gamma-Strahlen-Durchleuchtungsmaschinen. Nach [100] liegen die bloßen Anschaffungskosten bei über zwei Millionen Dollar. Neben den Wartungskosten fallen überdies Trainingskosten an.

Die Universität von Washington hat für den neuntgrößten Hafen der USA in Seattle eine Analyse durchgeführt, wie die Transportkosten durch die Anti-Terror-Maßnahmen wachsen [3]. Dabei haben sie zwei Rechnungsmodelle als Grundlage verwendet: Eine untere und eine obere Abschätzung für die entstehenden Kosten. Die wirklichen Beträge bewegen sich wohl in der Mitte. Zwischen der unteren und der oberen Abschätzung liegt ungefähr der Faktor zwei, Tabelle 1.4 zeigt den Mittelwert der errechneten Werte. Die Zahlen beziehen sich dabei auf den Dollarzuschlag, der pro 20-Fuß-Container aufzuwenden ist.

	C-TPAT	CSI	24-HMR	kombinierte Initiativen
Import	$ 12	$ 12	$ 55	$ 79
Export	$ 0	$ 0	$ 42	$ 42

Tabelle 1.4: Zusätzliche Kosten per TEU (nach [3])

Das World of Shipping Council hat Zahlen für das Jahr 2004 bekannt gegeben [22], nach denen der Hafen von Seattle jährlich 368 324 TEU exportiert und 680 780 TEU importiert. Basierend auf diesen

Werten und der Tabelle 1.4, zeigt Tabelle 1.5 die Gesamtkostenexplosion durch die beschriebenen Maßnahmen. Darüber hinaus wird die Annahme getroffen, dass die in Seattle entstehenden Kosten exemplarisch für die gesamten USA sind. Die Ergebnisse für den Staatenbund ergeben sich aus den Werten aus der Tabelle 1.3 hochgerechnet mit dem Ergebnis des Seattle-Modells.

	Export	Import	Gesamt
Seattle	$ 15 469 608	$ 53 781 620	$ 69 251 228
USA*	$ 337 891 890	$ 1 248 632 762	$ 1 586 524 652

* Unter der Annahme, dass die Daten von Seattle für die ganzen USA gelten.

Tabelle 1.5: Zusätzliche Kosten pro Jahr

Nach dieser Hochrechnung ergäben sich jährliche Kosten in Höhe von 1,6 Milliarden Dollar, die für die neuerlichen Maßnahmen von der Allgemeinheit zu tragen sind. In dem Modell der Universität von Washington wurde eine Container-Durchleuchtungsrate von 3% angenommen. Die Effektivität dieser Maßnahme ist fraglich. Eine wesentlich höhere Rate wäre in realistischen Anwendungen aber sicherlich aus Kosten-Nutzen-Sicht noch weniger sinnvoll.

1.2.4 Bewertung der aktuellen Terrorabwehrmaßnahmen

Die US-CBP hat ein Maßnahmenpaket verabschiedet, das auf dem ISPS-Code basiert. Die CSI geht zurzeit davon aus, dass sich der Status der Schiffe zwischen den Häfen nicht verändert – zumindest bezüglich der Container. Diese Annahme stützt sich auf den verschärften ISPS-Code-Regeln. Die aktuellen Maßnahmen überprüfen durch aufwändige und kostenintensive Kontrollen (AMS und Röntgen) an den statischen Endpunkten des Transports deren korrekten Status. Durch die Annahme, dass kein Unbefugter mehr in Container-Kontakt kommt, ohne dass zumindest eine Warnung ausgesprochen wird, reichen diese Maßnahmen womöglich aus.

Es ist allerdings fraglich, wie gut sich der ISPS-Code verlässlich anwenden lässt. Gerade Terroristen, die Mittel und Wege finden, Flugzeuge zu entführen und innerstädtisch zum Absturz zu bringen, werden auch die Möglichkeit haben, unerkannt auf Containerschiffe zu gelangen. Dieser Tatsache scheint sich auch die US-CBP bewusst zu sein, da sie in ihren Kernzielen zur CSI bereits weitergehende Ziele definiert. So spricht die Zollbehörde von einer notwendigen permanenten Überwachung durch intelligente Container [98].

Wären jedoch intelligente Container verfügbar, die es schaffen, durch technische Einrichtungen Warnungen auszusenden, wenn sich an ihrem Status etwas ändert, könnte der zurzeit lediglich aus den Manifestdaten des Containers bestehende statische Risikowert dynamisch gestaltet werden und so ein wesentlich effizienteres Maßnahmenpaket etablieren.

1.3 Aktuelle Lösungsansätze

Es haben sich schon mehrere Gruppen Gedanken darüber gemacht, wie eine höhere Transportqualität erzielt werden kann. Eine Maßnahme ist zum Beispiel das elektronische Siegel. Gemäß ISO 18185 („Bolt E-Seal“) können die elektronischen Siegel automatisiert berührungslos ausgelesen werden. Dabei können drei Siegel pro Sekunde aus vier Meter Entfernung gelesen werden. Aus dem Siegel sind der Siegelstatus, der Batteriestatus, die Siegelnummer und die Containernummer ersichtlich. Die Siegel sollen fälschungssicher sein. Der erhoffte Nutzen zeichnet sich bei einem Interchange ab, bei dem zurzeit das Siegel-Auslesen immer noch manuell geschieht. Ein elektronisches Siegel könnte automatisiert ausgelesen und Türöffnung im vorigen Transportabschnitt so detektiert werden. Die folgenden Abschnitte berichten über zwei exemplarische Ansätze, welche weiteren transportqualitätssteigernden Vorkehrungen aktuell getroffen werden. Beide Ansätze verfolgen den vielversprechenden Weg einer Transportüberwachung.

1.3.1 Cargobull Flottenmanagement

Im Bereich des LKW-Transports gibt es mittlerweile Speditionen, die interne „Tracking-und-Tracing“-Technologien einsetzen. T-Systems setzt zum Beispiel ein Flottenmanagement System namens „Cargobull Telematics“ bei der Spedition Schmitz Cargobull ein. Auf GSM-Technik basierend, wird so der Aufenthaltsort jedes Containers in ganz Europa überwacht. Die Zentrale von Cargobull kann auf diese Weise ihre Fracht zuordnen und zu jeder Zeit den Aufenthaltsort jedes LKW und den dazugehörigen Container bestimmen. Darüber hinaus wird der Status des Containers überwacht. Es werden dabei Sensoren am ganzen Container verwendet. Neben der Überwachung der Frachttür und der Kupplung zum Führerhaus werden auch Daten aus dem Containerinneren (wie Temperatur und Bewegung) permanent gemessen. Ein unbefugtes Abkoppeln oder Öffnen des Containers wird dem LKW-Fahrer per SMS-Alarmmeldung mitgeteilt. Das System ist aus technischer Sicht interessant, jedoch ist das gleichzeitige Überwachen des LKW-Fahrers (oder zumindest seines Mobiltelefons) aus Datenschutzgründen bedenklich. Dieses System ist nur auf Cargobull-LKWs verbaut. Die Container bilden eine Einheit mit dem Anhänger und lassen sich nicht abnehmen. Wegen dieser Tatsache und des Verwendens von GSM versagt dieses System auf hoher See. [83, 95]

1.3.2 Kühlkettenüberwachung

Im internationalen Transportwesen gibt es auch Versuche, die gesamte Transportkette in die Überwachung einzubeziehen. Bei der Kühlcontainer-Überwachung wird die Eigenschaft verwendet, dass Kühlcontainer bis auf kurze Unterbrechungen ständig mit Strom versorgt werden. Sensoren im Inneren des Containers ermitteln die Temperatur und zeichnen sie in elektronischen Logbüchern alle 15 Minuten auf. Die Innentemperatur des Kühlcontainers muss dabei immer $-18°C$ betragen. Mithilfe des Zeitstempels kann auf diese Weise im Nachhinein ermittelt werden, in wessen Zuständigkeitsbereich zu hohe Temperaturschwankungen aufgetreten sind. Für diese Systeme gibt es bereits internationale

Klassifizierungsrichtlinien und Überprüfungen, die von Klassifizierungsgesellschaften überprüft werden. Dieses System ist aber lediglich auf die Kühlcontainer und deren Innentemperatur beschränkt. Nach ISO-Standard 10368 [21] existieren bereits Systeme zur Fernauslesung. Über Powerline[4] kann das elektronische Logbuch von einer zentralen Stelle im Schiff ausgelesen werden.

1.3.3 Bewertung der Transportüberwachungssysteme

Die beiden Ansätze zeigen, dass bislang nur in engen Dimensionen gedacht wurde. Beides sind Insellösungen, die auf einen kleinen Bereich beschränkt bleiben. Bei „Cargobull Telematics“ wird eine komplette Containerinnenraumüberwachung durchgeführt, die permanent verfügbar ist und über intelligente Warnsysteme verfügt. Allerdings ist dieses interessante System auf einen Schmitz-Cargobull-LKW beschränkt. Wenn die Ware über Europa hinaus transportiert werden soll, kann einem dieses System nicht weiter helfen.

Bei den Kühlcontainern wird die gesamte Transportkette abgedeckt. Vom Zeitpunkt des Einladens bis zum Entnahmezeitpunkt wird die Kühlkette durchgehend überwacht. Im Hami-Melonen-Fall war die Schuldzuweisung leicht möglich, weil nur ein Transporteur involviert war. Wenn sich mehrere Transporteure den Frachttransport teilen, kann nur eine lückenlose Überwachung den Schuldigen sicher entdecken. Allerdings wird bei der Kühlcontainerüberwachung lediglich die Temperatur betrachtet. Die Kühlcontainer machen überdies nur einen Bruchteil des Containertransportwesens aus, eine erweiterbare Lösung wäre wünschenswert.

Parallel zu diesem Projekt forschen verschiedene andere Gruppen an übergreifenden Containerüberwachungen. Am weitesten ist ein Konsortium von IBM und Mærsk, das sich bereits im Feldversuchsstadium befindet. In Kapitel 7 werden das IBM-Projekt „ITL“ und das MASC-Projekt miteinander verglichen und die Unterschiede herausgearbeitet.

[4] Bei Powerline wird der Datentransfer über das normale Stromnetz geführt, in dem der Datenstrom in hohen Frequenzen auf die Stromnetzfrequenz moduliert wird. An jedem Punkt der Leitung können so die Kommunikationsdaten gelesen werden.

2 MASC – Monitoring and Security of Containers

Unter dem Akronym MASC versteht diese Arbeit ein Containerüberwachungssystem, welches die gesamte Transportkette umfasst und für alle Container einsetzbar ist. Die MASC-Anwendung adressiert dabei die angesprochenen Probleme und versucht einen adäquaten Lösungsansatz zu liefern. MASC steht dabei für „Monitoring and Security of Containers“.

2.1 Grundüberlegungen

Kapitel 1 hat gezeigt, dass der aktuelle Containertransport für viele Parteien Verbesserungspotenzial beinhaltet. Von entscheidender Bedeutung ist dabei, dass die gesamte Transportkette in den Überwachungsbereich einbezogen wird. Zur Terrorabwehr ist der Zeitpunkt interessant, zu dem Unbefugte sich Zugang zum Container verschaffen. Versicherungen müssen Zugang zu den Daten des gesamten Transportwegs haben, um genauer aus Tathergängen schlussfolgern zu können. Für die Versicherungen ist es dabei nicht so wichtig, dass der Container die gemessenen Daten sofort sendet – die Datenspeicherung in einem sicheren Datenspeicher ist relevanter. Terrorismusabwehr-Behörden und Shipper sind zur Schadensbegrenzung jedoch mehr an den „Live-Daten“ interessiert.

Die MASC-Anwendung versucht das gesamte Containertransportwesen zu verbessern. Der zentrale Ansatz einer Containerinnenraumüberwachung ist die Hauptinnovation des Systems, jedoch nicht der alleinige Verbesserungsansatz. Parallel zu der reinen Transportbedingungsüberwachung enthält die MASC-Anwendung ein System zur Automatisierung des Container-Manifests und der Bill-of-Lading. Außerdem ist es ein Leichtes, das System dahingehend zu erweitern, dass es die aktuellen Terrorabwehrmaßnahmen nachhaltig verbessern kann.

2.1.1 Containerüberwachung

Wie Abschnitt 1.1 aufzeigt, sind Containerdiebstahl und Schadensfälle ungelöste Probleme. Besonders die Entdeckung dieser Vorfälle geschieht meist erst am Ende oder nach Abschluss des Transports. Oftmals werden die Verursacher der Schäden nicht belangt, weil die Nachweisbarkeit nicht gegeben ist. Gerade dieser Zustand ist ungünstig, weil weder gutes Verhalten belohnt noch schlechtes Verhalten bestraft wird. Es zahlt sich nicht aus, besonders sorgfältig zu sein.

Sensorik

Das MASC-Projekt verfolgt den Ansatz, das Containerinnere mit Sensoren zu überwachen. Zur Zugriffsüberwachung und Diebstahlerkennung ist im Besonderen der Türöffnungssensor von Interesse. Als weiterer Sensor ist der RFID-Sensor interessant. Dieser erkennt, welche RFID-Tokens sich zurzeit im Container befinden. Wird die Ware gestohlen, enthält diese auch den eingewebten RFID-Token. Der RFID-Sensor liefert also ein Indiz dafür, welche Ware entwendet wurde. Allerdings ist es dazu erforderlich, dass sich ein RFID-Standard durchsetzt. Der weitere Vorteil eines RFID-Sensors wäre, dass die Containerüberwachung in ein Warenmanagementsystem eingebettet werden könnte.

Das Erkennen von Schadensfällen gemäß Abschnitt 1.1.5 lässt sich mit Hilfe von Temperatur- und Luftfeuchtigkeitssensoren bewerkstelligen. Auch Fusarien und Penicillium lassen sich über spezielle Sensoren (wie beschrieben) detektieren. Das Fallen eines Containern kann mit Hilfe von Bewegungs- respektive Kräftesensoren festgestellt werden.

Datenbereitstellung

Die Containerüberwachungseinheit beobachtet die Sensoren und erzeugt Ereignisse, wenn die Sensoren signifikante Veränderungen anzeigen. Diese Ereignisse können Temperaturwechsel, fallende Container, Feuchtigkeitsanstieg, aber auch Türöffnung oder Containerumschlag repräsentieren. Die Hauptaufgabe der MASC-Anwendung ist das sichere Speichern dieser Ereignisse in der Containereinheit.

Bei der Sofort-Benachrichtigung werden die generierten, in der Einheit sicher gespeicherten Ereignisse per Funk gesendet. Dieses Sofort-Benachrichtigungs-System benötigt daher eine Infrastruktur auf dem Transportmittel und dem Containerstellplatz, welche diese Funknachrichten annimmt und weiterleitet. Ziel dieses Benachrichtigungsdienstes ist die Möglichkeit, dass der Shipper zu jeder Zeit den Status seiner Ware begutachten und gegebenenfalls den Schaden abschätzen kann. Ebenso können die Frachtführer ihre Ware auf diese Weise fernüberprüfen und bei drohendem Schaden Gegenmaßnahmen einleiten. Meldet beispielsweise ein Containersensor Rauchentwicklung, kann der Frachtführer unmittelbar Maßnahmen einleiten, die einen Übergriff auf weitere Container erschweren.

2.1.2 Automatische Manifestierung

Jeder Transport benötigt eine Reihe von offiziellen Dokumenten. Neben der „Bill-of-Lading“ (BoL), die nachweist, welche Parteien sich vertraglich an dem Transport auf welche Weise beteiligen, gibt es weitere Frachtpapiere, Zolleinfuhrdokumente und auch das automatische Manifest. Es existieren zahlreiche verschiedene Arten von BoLs und jeder Staat hat unterschiedliche Einfuhrbedingungen. Etwas anders verhält es sich mit dem AMS, welches in Abschnitt 1.2.2 beschrieben ist. Hier gibt es jedoch einen genauen Standard, der einzuhalten ist (siehe Anhang E).

Die Schwierigkeit bei der Automatisierung von Frachtdokumenten besteht darin, alle Transportbeteiligten in ein System zu integrieren. Jeder Beteiligte müsste mit einem Zusatzaufwand dem System bekannt gemacht werden. Da dies für die an der Containerüberwachung teilnehmenden Parteien ohnehin

notwendig ist, sind diese dem System bereits bekannt. Das Erweitern eines Containerüberwachungsdienstes auf einen Manifestierungsdienst ist somit eine logische Fortführung. Ein integriertes System bringt den beteiligten Logistikern neben der Containerüberwachung folglich weitere Vorteile.

2.1.3 Terrorabwehrmaßnahmen

Die US-CBP setzt bei der CSI zurzeit ausschließlich auf die Auswertung der automatischen Manifeste. Dies reicht allerdings nicht aus. Der ermittelte Risikowert ändert sich während des gesamten Transportes nicht. Er ist lediglich durch die beim Transport involvierten Partner und Länder gekennzeichnet.

Die Einführung eines dynamischen Risikowertes ist wünschenswert. Wenn die Containertür nach der letzten Kontrolle geöffnet wurde, sollte sich der Risikowert drastisch erhöhen. Neben dem Türöffnungssensor sind weitere Sensoren für die Terrorismusabwehr interessant. Ein Radioaktivitätssensor könnte schmutzige Bomben erkennen. Das Verhältnis zwischen CO_2 und O_2 in einem Container kann ein Indiz dafür sein, dass sich (atmende) Menschen darin aufhalten.

Die Zollbehörden aller von einem Container angesteuerten Länder sollten Zugriff auf die Sensordaten haben, die Auskunft darüber geben, ob eine Bedrohung für die nationale Sicherheit von diesem Container ausgehen könnte.

2.2 Allgemeine Anforderungen

2.2.1 Zu- und Ableitungen

Das Containerüberwachungssystem benötigt zum Datentransport einen Uplink und zur Versorgung eine Stromzufuhr. Ein genormtes Kabel könnte sowohl Stromzufuhr als auch Datentransport ermöglichen. Dieses müsste vor dem Entladen und nach dem Beladen des Containers jedoch händisch gesteckt werden. Die Tatsache berücksichtigend, dass ein Containerschiff bis zu 10 000 Containerstellplätze haben kann, ist der für das Einstecken und Abklemmen der Kabel benötigte Zeit- und Kostenaufwand immens. Auch das Bearbeiten eines kompletten Containerzuges und das manuelle Einstecken aller Containerkabel ist ein hoher Aufwand. In Deutschland dürfen Containerzüge auf Grund begrenzter Gleislängen in Bahnhöfen maximal 700 Meter lang sein, in den USA sind sie häufig um ein Vielfaches länger. Bei Bahn und Schiff müssten also automatische Systeme nachgerüstet werden, die die Verbindung beim Beladen selbsttätig herstellen. Es bleiben jedoch zwei Probleme bestehen, die nicht so leicht zu lösen sind. Erstens gibt es weltweit unzählige auf der grünen Wiese gebaute Containeryards und -depots, die nicht den finanziellen Aufwand betreiben werden, alle Containerplätze mit Stecksystemen zu versehen. Zweitens wird sich das System niemals durchsetzen, wenn zunächst alle Schiffe und Bahnwagen aufwändig umzurüsten wären, bevor es überhaupt in Betrieb geht. Es ist ein Schlüsselkriterium, dass die Containerüberwachung mit der aktuellen Infrastruktur kompatibel ist und nur minimalen Mehraufwand im operativen Geschäft bedeutet.

Eine Möglichkeit wäre, den Containerstahl als Übertragungsmedium zu benutzen und Strom über Batterien bereitzustellen. Auf diese Weise müssten nur die übereinander stehenden Container auf einem Kanal zusammengeschaltet und jeder unterste Container mit einer Datenleitung zum Nachbarn versehen werden. Wesentlich besser ist die Idee, die Catwalks[1] als Übertragungsmedium zu verwenden. So werden lediglich eine Empfangseinheit pro Catwalk benötigt (Anzahl Catwalks = Anzahl Containerstapel -1). Hierbei würde die Kommunikation über Funk abgewickelt und die dazu benötigte Stromversorgung per Batterie erfolgen.

2.2.2 Vertragsstruktur

Damit die MASC-Anwendung eine realistische Chance hat, sich auf dem Markt zu beweisen, muss sie sich gut in bestehende logistische Abläufe einbetten lassen. Sie sollte in den täglichen Abläufen der damit Arbeitenden keinen unnötigen Mehraufwand erzeugen. Im Besonderen ist zu berücksichtigen, dass die Strukturen der Abhängigkeiten der Vertragspartner und des über Jahrhunderte gewachsenen Prozedere der Transportbeauftragung (wie Abschnitt 1.1.3 beschreibt) nicht aufgebrochen, sondern vom Zugriffssystem abgebildet und ausgenutzt werden. Eine völlig neue Abläufe bei der Vertragsabwicklung erfordernde Anwendung wird sich nicht durchsetzen.

Zurzeit nehmen am weltweiten Containertransport völlig unterschiedliche Firmen teil. Dabei gibt es Partnerschaften zwischen kleinen Betrieben mit wenigen Mitarbeitern und großen vollautomatisierten Unternehmen. Firmen, die bislang ohne computergestützte Auftragsverarbeitung gearbeitet haben, sollten die Anwendung leicht einsetzen können. Gleiches sollte für Unternehmen mit bestehenden Warenverfolgungssystemen gelten. Eine gute systematische Integrierbarkeit ist also ebenso gefordert wie eine einfache Oberfläche für die Standardprozeduren.

Neben den direkten Vertragspartnern, die dem transitiven Vertragsbaum aus Abschnitt 1.1.3 folgen, gibt es weitere Parteien: Lokale Zollbehörden und Verifizierungsunternehmen. Bei der Erstellung eines umfassenden Containerüberwachungssystems sollten diese in die Überlegungen einbezogen werden.

2.2.3 Managementkonsole

Begleitend zur Containerüberwachung muss die MASC-Anwendung also ein Softwaresystem bereitstellen, welches auf einzelnen PCs genauso leicht funktioniert, wie es sich in bestehende Transportverwaltungssysteme einklinken lässt. Nur so kann sichergestellt werden, dass kleine Logistiker (Wenige-Mann-Betriebe) mit global agierenden Konkurrenten mit existierenden IT-Infrastrukturen mithalten. Daher ist es notwendig, dass Schnittstellen offen sind und das Programm nach Möglichkeit frei verfügbar ist. Proprietäre Lösungen werden sich vielleicht als Nischenprodukt durchsetzen, aber keine Chance am Massenmarkt haben.

[1] Catwalks: Luftspalten zwischen den Containerstapeln

Das Programm sollte eine einfach benutzbare Managementoberfläche bereitstellen. Offene Schnittstellen müssten aber gleichermaßen beschrieben sein, damit die Integrierbarkeit dieses Systems in proprietäre Logistiksysteme möglich ist.

2.2.4 Transportbedingungen

Es gibt auf Containerschiffen nur wenige, über einen Stromanschluss verfügende Kühlcontainerstellplätze. Damit die Containerüberwachung schiffsunabhängig auf der gesamten Transportkette funktioniert, müssen die Container also stromanschlussautark arbeiten, sprich mit eigener Stromquelle versorgt sein. Da Container eine Laufzeit von bis zu acht Wochen haben und die Einheit nicht nach jedem Transport gewechselt werden soll, müssten die Einheiten mehrere Jahre arbeiten.

Des Weiteren ist der Container besonderen Transportbedingungen ausgesetzt: Ein Sturz aus wenigen Zentimetern Höhe wird nicht als Fallengelassen betrachtet, ein Transport durch Sibirien mag sich von den Außentemperaturen stark von einem Transport durch Death Valley unterscheiden. Auf hoher See ist die Salzkonzentration der feuchten Luft wesentlich höhen als auf einem kontinentalen Containerdepot. Selbst unter extremen Bedingungen muss die Einheit weiterhin funktionieren und alarmieren.

2.2.5 Sicherheit

Bei vielen Anwendungen wurde der Fehler begangen, dass sich die Entwickler während der Entwurfsphase keine Gedanken über Sicherheit gemacht haben. Ein prominentes Beispiel ist das Drahtlosnetzwerk. Die Entwickler wollten das drahtgebundene Netzwerk auf eine höhere Technologiestufe stellen, übersahen aber, dass sie sich damit neue Angreifer ins Boot holen. Durch die Verwendung der Funktechnologie haben nicht nur Personen Zugriff auf das Netzwerkmedium, die eine Steckdose erreichen können, sondern vielmehr alle Personen, die sich in Reichweite der Basisstation befinden. Obwohl der IEEE-Standard 802.11i mittlerweile verabschiedet wurde, der seit drei Jahren durch WPA und besonders WPA2 ein adäquat sicheres Zugriffsmanagement auf die Nutzdatenübertragung bereitstellt, sind zahlreiche Netzwerke immer noch nicht zureichend abgesichert. Hinzu kommt, dass die ISO/OSI-Schicht-2-Befehle immer noch unverschlüsselt übertragen werden, um kompatibel mit dem WLAN-Standard IEEE 802.11 zu sein. Daher ist es nach wie vor leicht möglich, jede beliebige Verbindung als Außenstehender mit einem „Deauthentication-Frame“ zu beenden. Dieser Missstand kann im Nachhinein nicht mehr beseitigt werden, ohne Kompatibilität mit dem bestehenden System zu verlieren.

Der Fokus dieser Arbeit liegt daher nicht auf dem Entwurf eines Containerüberwachungssystems, sondern, wie der Buchstabe S von MASC besagt, auf dem Entwurf eines SICHEREN Containerüberwachungssystems. Das Design der Anwendung wird aus diesem Grund in ein Sicherheitskonzept eingebettet, denn auch bei der Containerüberwachung wird eine neue Technologie eingesetzt. Es werden noch immer Faxkopien und an den LKW-Fahrer übergebene handschriftliche Bill-of-Ladings angefertigt. Wenn ein Angreifer an die Daten der BoLs herankommen möchte, muss er physisch auf die ausgedruckten Ladungspapiere zugreifen können. Die Grundvoraussetzung dazu ist, dass er weiß, wo

sich diese Papiere befinden. Erst dann kann er sich Zugang zu den Informationen verschaffen. Wenn das Internet genutzt wird, um die Frachtdaten zu übertragen, werden die Daten neuen Gefahren ausgesetzt. Angriffe können anonym und von überall auf der Welt auf diese Daten lanciert werden. Es reicht also nicht, nur das bestehende Verfahren zu automatisieren; vielmehr müssen adäquate Sicherheitsüberlegungen angestellt werden, um sich gegen das veränderte Risiko abzusichern.

2.3 MASC-Einheit

Herzstück des Containerüberwachungssystems ist die Containereinheit. Diese wird im Container implantiert und steuert die Containerüberwachung, indem sie die angeschlossenen Sensoren ausliest. Ihre Aufgaben bestehen darin, alle Unregelmäßigkeiten zu verzeichnen, diese sicher zu speichern und, wenn möglich, direkt und drahtlos zu übertragen. Dabei muss eine Vielzahl spezieller Anforderungen beachtet werden. Mit einer MASC-Einheit bestückte Container nennen sich MASC-Container.

2.3.1 Anforderungen an die MASC-Einheit

Transportbedingungen

Wie in 2.2.4 angesprochen, müssen Einheit und angeschlossene Sensoren überall auf der Welt funktionieren. Feuchtigkeit und Wärme (beziehungsweise Kälte) dürfen der Computereinheit nichts anhaben. Genauso wichtig ist eine Robustheit gegenüber äußeren Kräften, die auf den Container wirken. Natürlich gibt es Grenzen der Belastbarkeit von elektronischen Bauteilen, jedoch ist es wichtig, dass die MASC-Einheit erst aufhören darf zu arbeiten, wenn der Container so stark beschädigt ist, dass dies auch durch einen Interchange erkannt werden kann.

Positionierung

Die Containerüberwachungseinheit im Container sollte keinerlei Einschränkung an Packung und Verpackungsvorschriften bedeuten. Sie darf nicht in den Container hinein ragen und sollte sich von der Oberfläche nicht abheben. Da die Frachtcontainer aus Sicken[2] bestehen, bietet es sich an, die Einheit in einer dieser Vertiefungen unterzubringen. Der Sinn der Sicke darf durch diesen Einbau nicht beeinträchtigt werden. Da die Containerüberwachungseinheit die Sicke unbeeinträchtigt lässt und sich lediglich in dem entstehenden Freiraum ausbreitet, sind keine Einschränkungen an die Steifigkeit zu erwarten. Die Container sind im Regelfall mit der Tür zum Catwalk ausgerichtet, weswegen die Kommunikationsantenne in der Türregion positioniert sein sollte. Es bietet sich daher an, die MASC-Einheit in der ersten Sicke hinter der Containertür zu positionieren, damit die Einheit mit Antenne und Türöffnungssensor leicht verbunden werden kann.

[2] Bei einer Sicke handelt es sich um rinnenförmige Vertiefungen in der Containerdecke und -wand, die zur Stabilitätserhöhung in den Stahl gepresst wurde.

Lebensdauer

Von eminenter Bedeutung ist die Lebensdauer der MASC-Einheit. Da sie nicht nach jedem Transport abgesteckt werden soll, sondern fest mit dem Container verdrahtet ist, ist eine mehrjährige Lebensdauer erforderlich. Eine komplett autarke Energieversorgung (zum Beispiel über Vibrationsenergiegewinnung) wäre ideal.

2.3.2 Funktionsweise

Die MASC-Einheit liest in definierten Intervallen die Sensordaten aus und vergleicht sie mit den zuvor gemessenen. Weichen diese signifikant voneinander ab, wird ein Ereignis generiert. Dieses Ereignis wird sicher abgespeichert.

Das Abfrageintervall sollte dabei klein genug sein, um jede Änderung aufzeigen zu können, jedoch so groß gewählt sein, dass keine unnötige Energie aufgebracht wird. Dieser Zielkonflikt ist exemplarisch für die gesamte Konstruktion der MASC-Einheit. Jede Funktion und jeder Hardwarebaustein benötigt Energie und beeinflusst ihrerseits die Lebensdauer. Eine Lebensdauer von mehreren Jahren zu erreichen, ist eine schwierig zu erfüllende Anforderung – deswegen muss auf Energieeffizienz besonderer Wert gelegt werden, ohne dass die Funktionalität beeinträchtigt wird.

Die Signifikanz, mit welcher die Werte abweichen dürfen, ohne dass ein Ereignis erzeugt wird, sind Erfahrungswerte, die derzeit nicht bestimmt werden können. Auch hier spielt die Energieeffizienz eine Rolle. Jedes Ereignis verbraucht Strom: zur Erstellung, um im Speicher gehalten zu werden und beim Senden. Daher sollte die Reizschwelle überlegt gewählt werden.

2.4 Datenabtransport

Die Container stehen in Türmen angeordnet und in Stapeln zusammengefasst auf Schiffen, Containeryards oder Depots, einzeln auf LKWs und in einer Reihe auf Bahnwagen. Die Kommunikation mit einem Container ist problemlos möglich, wenn sich die Antenne in direkter Sichtlinie zum Sender befindet. Allerdings besteht der Container aus Stahl und dieser lässt keine Funkwellen passieren. Der Kommunikationsaufbau mit einem Container von der Seite oder von hinten kann daher nicht realisiert werden. Schwierig wird es ebenfalls, wenn die Container in Stapeln stehen. Zwischen den Containern ist auf Schiffen unterschiedlich viel Platz. Bis zu 15 Containertürme nebeneinander ergeben einen Stapel, der Containerturm ist dabei sieben Container hoch. Der Abstand der Containerrückwände beträgt dabei 25, 38 oder 80 Millimeter; die Containerseiten stehen 76 Millimeter auseinander. Vor jedem Container befindet sich ein Catwalk, der (mindestens) 60 Zentimeter Breite hat. Tests haben ergeben, dass 60 Zentimeter ausreichen, um drahtlose Kommunikation störungsfrei durchführen zu können. Die durch die Containerstapel entstehenden Stahlwände wirken mitunter sogar signalverstärkend.

Es wird pro Catwalk lediglich eine Antenne benötigt, die mit einer Basisstation verbunden ist. Diese sendet die empfangenen Daten über einen Gateway ins Internet. Auf Containerschiffen existiert bereits

eine Internetverbindung über das schiffseigene Satellitensystem, diese könnte also von dem Gateway verwendet werden. Auf Containeryards und -depots könnte dieser Uplink über eine normale Telefonleitung oder einen DSL-Anschluss realisiert werden.

Bei mobilen Überlandverkehrsmitteln wie LKW, Bahn und Binnenschiff bietet sich die Verwendung von UMTS oder anderen Mobilfunktechnologien an. Bei Bahn und LKW ist der Aufwand etwas größer, allerdings sind die Überlandverkehrsmittel bereits mit Mobiltelefonverbindungen ausgerüstet. Weitere Synergien ergäben sich beim Integrieren von Flottenmanagement-Lösungen wie bei Cargobull.

Der Nachteil dieses Systems ist die Notwendigkeit einer intakten Infrastruktur. Jeder MASC-Container benötigt eine Basisstation in Reichweite, wenn der Dienst verfügbar sein soll. Wenn allerdings diese Infrastruktur besteht, dann kann jeder Container seine Daten zu jeder Zeit absenden, ohne dabei von anderen Faktoren abhängig zu sein.

2.4.1 Meshing

Eine Alternative zur Positionierung einer Basisstation pro Catwalk wäre, die Container als Knoten eines Netzwerks zu betrachten und sie untereinander die Nachrichten weiterleiten zu lassen. Hierzu wäre es allerdings nötig, jeden Container mit zwei Antennen auszustatten. Containern wäre es möglich, über die Grenzen des eigenen Catwalks hinaus zu kommunizieren. Auf diese Weise könnten Basisstationen auf den Transportmitteln und Umschlagsstellen eingespart werden. Die Kehrseite der Medaille sind neue Probleme im Bereich der Sicherheit und Verfügbarkeit.

Die Knoten des auf diese Weise entstehenden Meshingnetzwerks organisieren Routing und Nachrichtenweiterleitung selbst. In einem derartigen Netzwerk bräuchte nur ein Container Kontakt zu einer Basisstation zu haben. Alle anderen Container würden die Basisstation transitiv über diesen einen Container erreichen. Damit dies funktioniert, müssen zwei Voraussetzungen erfüllt sein: Jeder Container erreicht einen anderen Container, über den ein transitiver Weg zu einer Basisstation möglich ist, und alle auf der Route liegenden Knoten beteiligen sich am Meshingnetzwerk.

Durch die Einführung von Meshing wird weniger infrastrukturelle Hardware benötigt, da nicht jeder Catwalk mit einer Basisstation ausgerüstet werden muss. Nur eine Basisstation pro Transportmittel wird benötigt. Dies geht aber auf Kosten des Energieverbrauchs der einzelnen Knoten. Jeder Knoten muss ständig die Umgebung überwachen und gegebenenfalls Nachrichten in Empfang nehmen. Diese müssten dann gemäß einer vorher zu errechnenden Route über einen Nachbarknoten weitergegeben werden.

Die Meshingfrage reduziert sich also auf die „globale“ Abwägung, ob die Energiekosten für jede Einheit eher zu verkraften sind, oder ob alle Containerstellplätze weltweit eine aufwändigere Hardwareausstattung benötigen. Allerdings sind nicht nur die Energiekosten für jede Containereinheit höher, sondern auch deren Lebensdauer geringer. Es wären also mehr Containereinheiten notwendig, um die gleiche Lebensdauer zu erzielen. Da eine Containereinheit wesentlich teurer ist als eine Basisstation, ist

es günstiger, auf Meshing zu verzichten (unter der Voraussetzung, dass die Durchsetzung der MASC-Container groß ist und es viele Containereinheiten gibt). Die folgenden zwei Unterabschnitte zeigen jedoch, dass diese globale Abwägung gar nicht entscheidend ist. Vielmehr gibt das knotenindividuelle Verhalten den Ausschlag, ob Meshing verwendet werden kann oder nicht.

Sicherheit

Ein grundsätzliches Problem beim Verwenden von Meshingnetzwerken ist die Tatsache, dass es Angriffe gibt, die nicht effizient verhindert werden können. Die *Sybil-Attacke* [28] ist ein Beispiel. Bei diesem Angriff erzeugt ein Netzwerkknoten mehrere virtuelle Netzwerkteilnehmer, die sich alle gleichberechtigt dem Netzwerk anschließen. Durch diesen Angriff umgeht ein Knoten durch Kooperation die implizite Heterogenität des Netzwerks. Ein Sybil-verwandter Angriff könnte auch im MASC-Szenario durchgeführt werden. Ein Containerbesitzer besitzt oftmals mehrere Container auf einem Stellplatz oder Schiff. Wenn dieser mehrere MASC-Container in einem Meshingnetzwerk stellt, kann er es ausnutzen, dass er mehr Informationen über den Datenfluss hat. Er kann die Routingeigenschaften des Netzwerkes zu seinen Gunsten verbessern. Es ist überdies möglich, dass nur Nachrichten von eigenen Knoten weitergeleitet und Konkurrenten so ausgesperrt werden. Durch *Selektive Weiterleitung* könnte eine MASC-Einheit zunächst die Gewissheit erlangen, dass eine Verbindung zum Server besteht. Anschließend können Knoten auf der Route Nachrichten gezielt unterbinden und somit die Verbindung zu beliebiger Zeit kappen. Vergleichbare Angriffe (zum Beispiel False-Basestation-Attack) sind auch bei Verzicht auf Meshing denkbar, allerdings bei weitem nicht so leicht durchzuführen. Alles, was ein Angreifer im Meshing-Fall benötigt, ist eine leichte Umkonfigurierung der Containereinheit – Konkurrenten könnten dies einfach bewerkstelligen.

Individuelles Mitspielverhalten

Viel schwerwiegender als diese recht akademischen Angriffe ist die Tatsache, dass der Knoten fürs „Fair-Spielen“ bestraft und fürs „Falsch-Spielen“ belohnt wird. Beim Meshing kann ein Knoten nur eine Nachricht an die Basisstation senden, wenn andere Knoten die Nachrichten weitersenden. Gerade bei den vorliegenden Knoten kostet Empfangsbereitschaft, Senden und Empfangen aber viel Energie. Wenn eine Containereinheit ihre Sende-/Empfangseinheit ausschaltet und sich somit nicht am Meshing beteiligt, wird diese mit einem geringeren Energieverlust belohnt. Dies ändert sich erst, wenn zwischen den Containereinheiten ein Vertrauensverhältnis bezüglich des „Fair-Spielens“ aufgebaut ist. Dies ist aber im MASC-Fall per se nicht gegeben: Verschiedene Shipper, Speditionen und Containerbesitzer haben kein Interesse daran, die Arbeit der Konkurrenten zu unterstützen.

Dieser Sachverhalt kann auch als Spiel bezüglich der Spieltheorie formuliert werden. Dazu wird das Kommunikationsverhalten modelliert und auf zwei Teilnehmer beschränkt. Übertragen auf den MASC-Fall wären A und B zwei Container, die jeweils für sich die Entscheidung treffen müssen, ob sie am Meshing teilnehmen. Durch die modellierte und verallgemeinerte Reduzierung auf zwei Teilnehmer ist das Resultat dieses „Spiels“, dass ein Meshing nur dann möglich ist, wenn A und B mitmachen. Sobald

ein Teilnehmer aussteigt, kann der andere keine Nachrichten mehr absenden. Dabei sind die Kosten für das Senden/Empfangen c, wobei c einen Wert zwischen 0 und 0,5 annimmt[3]. Jeder Knoten möchte also seine Nachricht zu einer Basisstation senden. Der Nutzen, der bei Erfolg eintritt, wird mit dem Wert 1 gewichtet. Wenn die Nachricht nicht verschickt werden kann, ist der Nutzwert 0.

Meshing	A spielt mit		A spielt nicht mit	
Individuelle Kosten für	A	B	A	B
B spielt mit	$1-c$	$1-c$	1	$-c$
B spielt nicht mit	$-c$	1	0	0

Tabelle 2.1: Spieltheorie im Meshing-Fall

Die Situation ist nun die Folgende: Es gibt zwei Knoten A und B. Beide haben die Wahl, die Nachricht des anderen entweder weiterzuleiten und dabei die Kosten c in Kauf zu nehmen oder sich an dem Weiterleiten nicht zu beteiligen. Tabelle 2.1 zeigt die vier Fälle, die durch diese zwei Freiheitsgrade entstehen können. Die Spieltheorie geht davon aus, dass alle Teilnehmer ihre persönliche Kostenfunktion abwägen und ihr Verhalten davon abhängig machen. Ein sich nicht am Weiterleiten von Nachrichten beteiligender Knoten (Knoten spielt nicht mit) hat den Nutzen 1, wenn der andere mitspielt, oder den Nutzen 0 für den Fall, dass der andere ebenfalls nicht mitspielt. Würde er sich jedoch am Spiel beteiligen, wäre seine Nutzenfunktion $1-c$, wenn der andere fair spielt, oder $-c$ bei einem nicht mitspielenden Kommunikationspartner. In dem Fall, dass er nicht mitspielt, sind seine Kosten also in beiden Fällen um c geringer. Mitspielen zahlt sich also nicht aus.

Das so genannte Nash-Gleichgewicht bezeichnet den sich einstellenden Zustand, wenn alle Teilnehmer ihre Kostenfunktion optimieren. Für das Zwei-Teilnehmer-Modell ist dieser, dass beide nicht mitspielen. Dieses Spiel kann auf viele teilnehmende Container erweitert werden, da die Entscheidung für jede Kommunikation mit einem anderen Knoten der getroffenen Abwägung dieses Modells entspricht. Auf Grund der Annahme, dass sich die Container nicht bezüglich des „Fair-Spielens“ vertrauen, ist Meshing konzeptionell ungeeignet.

Kein Meshing	A spielt mit		A spielt nicht mit	
Individuelle Kosten für	A	B	A	B
B spielt mit	$1-k$	$1-k$	0	$1-k$
B spielt nicht mit	$1-k$	0	0	0

Tabelle 2.2: Spieltheorie ohne Meshing

Als Gegenprobe sei der Fall ohne Meshing durchgespielt (Tabelle 2.2). Die Kostenfunktion sei k, wobei dieses k wesentlich kleiner ist als das c im vorhergehenden Fall. Das liegt daran, dass in diesem Fall kein ständiges Abhören des Datenkanals erforderlich ist – daher ist natürlich auch die Kostenfunktion für das „Fair-Spielen“ geringer. Das Nash-Gleichgewicht stellt sich ein, wenn beide mitspielen, da ein

[3] Der Nutzen, den ein Mitspieler durch das Spiel erlangt, ist 1. Die Kosten für das Spiel sollten also wesentlich geringer sein, weil sich das Spiel sonst nicht lohnt. Diese allgemeine Anforderung modelliert die Tatsache, dass die Sendekosten sehr viel kleiner sein müssen als der zu erwartende Nutzen der Anwendung, andernfalls würde die Anwendung auf Drahtlosübertragungen gänzlich verzichten.

Mitspielen sich immer auszahlt. Da die beiden Ereignisse (Knoten A spielt mit und Knoten B spielt mit) unabhängig voneinander sind, kann sich jeder Knoten überlegen, ob er den Nutzen $1-k$ fürs Mitspielen erhält, oder keinen Nutzen, aber auch keine Kosten fürs Nicht-Mitspielen. Mitspielen lohnt sich in diesem Fall (da $1-k>0$).

Zusammenfassend kann diese Diskussion darauf zurückgeführt werden, dass Senden und Empfangen sehr teuer ist. Wären die Kosten c in der Nähe von 0, lohnte sich Meshing und alle Beteiligten machten vermutlich mit. Da das Beteiligen teuer ist, lohnt sich das Meshing nicht für den EINZELNEN Teilnehmer. Das einfache Zurückführen auf die „globale“ Abwägung *Energieverbrauch pro Einheit contra Hardwareaufwand pro Catwalk* ist also hinfällig. Da die Containereinheiten autark sind, ist nur die individuelle Kostenoptimierung entscheidend – und die spricht gegen das Meshing.

Schlussbetrachtung Meshing

Wenn Meshing ERZWUNGEN werden sollte, wäre es unabdingbar, die Container regelmäßig auf korrekte Funktion zu überprüfen. Dazu wäre es notwendig, dass die Verifizierungsgesellschaften die Containereinheiten häufig unter die Lupe nehmen. Diese Prüfungen wären aber recht kostenintensiv, da sie jedesmal eine Softwareanalyse erforderten. Des Weiteren würden sie eine zusätzliche unerwünschte Bürokratiehürde in der MASC-Anwendung nach sich ziehen. Die globale Abwägung kann als Randbedingung dieser Diskussion gesehen werden. Nur wenn diese zu Gunsten von Meshing beantwortet würde, wäre es überhaupt sinnvoll, Meshing einzusetzen. Dieses ist bereits, wie angesprochen, sehr fragwürdig. Für die heterogene MASC-Anwendung, bei der die Containereinheiten unterschiedlichen Besitzern und Herstellern zugeordnet sind, zeigt die Spieltheorie, dass Meshing das falsche Konzept ist[4]. Der entstehende Bürokratieaufwand und die ebenfalls leicht veränderte Sicherheitslage sind weitere Nachteile, die zu Lasten von Meshing wirken. Diese stehen dem einen Vorteil entgegen, dass Hardware eingespart werden kann. Das Verwenden von Meshing im Umfeld der MASC-Anwendung scheint damit wenig sinnvoll.

[4] Bei der angesprochenen ITL-Lösung von IBM und Mærsk sieht dies anders aus, da alle Containereinheiten einer Partei gehören – diese Diskussion wird in Abschnitt 7.2 aufgegriffen.

3 MASC-Infrastruktur

In diesem Kapitel wird auf die Entwurfsentscheidungen eingegangen, die die Infrastruktur der MASC-Anwendung betreffen. Die getroffenen Abwägungen basieren dabei auf einer Abstimmung der zum Teil konträren Anforderungen aus Abschnitt 2.2.

3.1 Zugriffsstrategie

Die Ausgangslage ist wie folgt: Der Container speichert eine Vielzahl von Ereignissen in seinem lokalen Speicher. Shipper, Versicherer, verschiedene Logistiker und Zoll wollen auf diese Ereignisse zugreifen. Allerdings darf nicht jede Partei alle Daten lesen. So sollten nur Shipper und Versicherer Zugriff auf die RFID-Daten des Containers erhalten, weil diese mit der Information der im Container befindlichen Waren verknüpft sind. Gravierender verhält es sich mit den Terrorbekämpfungsinformationen: Diese sollten ausschließlich für die Zollbehörden der Länder lesbar sein, die auf der Transportroute liegen. Es gibt also viele unterschiedliche Zugreifende mit verschiedenen Rechten. Daher ist es notwendig, dass ein adäquates Zugriffskontrollsystem den Zugang zum System überwacht.

3.1.1 Zugriffskontrolle

Begrifflichkeiten

Definition 3.1 (Zugriffsrecht)**.** *Ein Zugriffsrecht (oder nur Recht) definiert, wie eine zugreifende Entität auf einen Datensatz zugreifen darf. Klassische Beispiele von Zugriffsrechten auf einen Datensatz sind* LESEN*,* EINFÜGEN*,* VERÄNDERN *und* LÖSCHEN*. Die letzten drei Rechte können auch zu* SCHREIBEN *zusammengefügt werden.*

Definition 3.2 (Richtlinie)**.** *Eine Richtlinie (engl. „policy“) definiert Regeln, unter denen ein Zugriffsrecht gilt. Sie bindet Bedingungen an ein Zugriffsrecht. Das Auswerten der Richtlinien führt unter den gegebenen Randbedingungen zu der Entscheidung, ob das enthaltene Zugriffsrecht gilt. Eine Richtlinie enthält somit das Zielobjekt, ein für das Objekt geltendes Zugriffsrecht und die Bedingungen, unter denen dieses gilt. Die Auswertung einer Richtlinie führt zur Entscheidung, ob der Zugriff gestattet oder verweigert wird. (Vgl.* [105]*)*

Ein Zugriffsrecht beschreibt, ob und wie Benutzer oder Dienste auf Programme, Dateien/Daten, Drucker, Dienste oder Netzwerke zugreifen dürfen. Richtlinien können dafür verwendet werden, Zugriffsrechten Bedingungen zuzuweisen. Auf diese Weise kann ein Zugriffsrecht nur für eine zeitliche Periode, spezielle Benutzer oder Benutzergruppen gelten. Es kann auch an andere Bedingungen gekoppelt

werden. Richtlinien müssen also zum Zugriffszeitpunkt ausgewertet und die Zugriffsentscheidung somit zeitnah getroffen werden. Nachfolgend wird beschrieben, wie bei einem Zugriffskontrollsystem die Richtlinienverarbeitung verteilt wird. Die anschließenden Definitionen beruhen auf denen des *Swedish Insitute of Computer Science* [93]. Weitere ähnliche Definitionen zu diesem Thema finden sich in [101, 105, 5].

Definition 3.3 (PAP)**.** *Der PAP (engl. „Policy Administration Point“) ist der Dienst, der die Richtlinien semantisch erstellt. Er bestimmt inhaltlich, welche Regeln bezüglich welcher Dienste gelten. Der PAP ist für die* ***Richtlinienerstellung*** *zuständig.*

Definition 3.4 (PIP)**.** *Der PIP (engl. „Policy Information Point“) ist der Dienst, der die Richtlinien abspeichert. In der Regel handelt es sich dabei um einen datenbankbasierten Dienst.*

Definition 3.5 (PDP)**.** *Der PDP (engl. „Policy Decision Point“) hat zwei Aufgaben, da er als einziger mit dem PIP kommuniziert. Er wertet die Richtlinien aus und entscheidet, ob das enthaltene Zugriffsrecht zurzeit gültig ist – der PDP ist der* ***Richtlinienentscheider****. Seine zweite Aufgabe ist das Überprüfen neu eingereichter Richtlinien. Sind diese syntaktisch korrekt und ist es dem PAP erlaubt, diese einzureichen, fügt der PDP die neue Richtlinie in den PIP ein.*

Definition 3.6 (PEP)**.** *Der PEP (engl. „Policy Enforcement Point“) ist der Dienst, der schlussendlich die Zugriffsentscheidung vollstreckt. An ihn muss sich gewendet werden, wenn der Zugriff auf einen Datensatz gewünscht wird. Der PEP ermittelt den Zugreifenden und den Zugriffswunsch und übersendet dem PDP diese Daten. Der PDP wertet die Regeln aus und entscheidet, ob der Zugriff richtlinienkonform ist. Dies wird dem PEP mitgeteilt, der die* ***Richtlinienvollstreckung*** *übernimmt.*

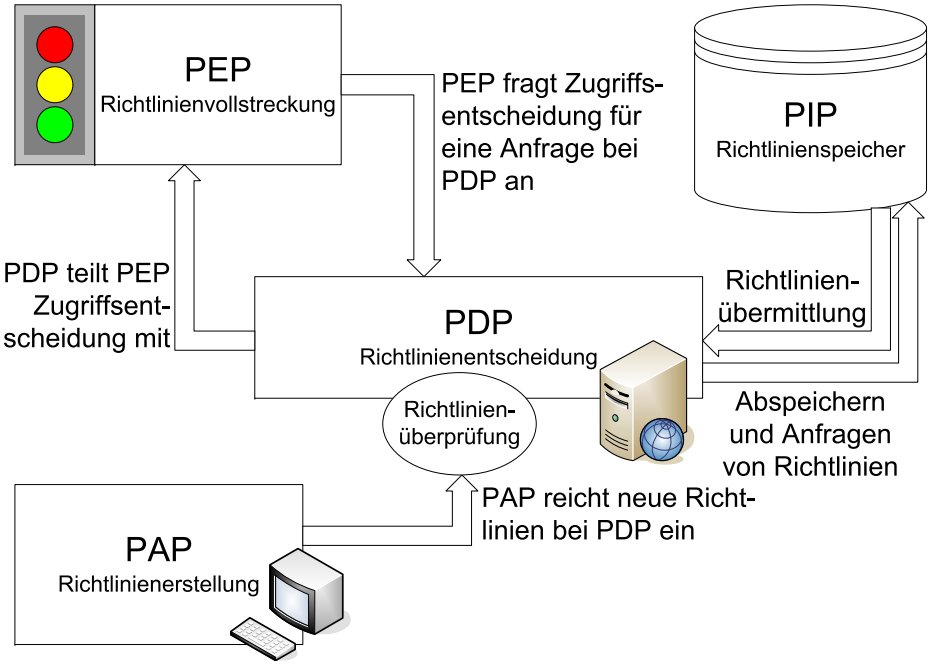


Abbildung 3.1: Richtlinienverwaltung

Abbildung 3.1 veranschaulicht wie die einzelnen Dienste untereinander kommunizieren. Der PIP ist ein einfacher Datenbankdienst, der lediglich wissen muss, wie die Richtlinien abgelegt werden. Der einzige Zugreifende ist der PDP. Dieser übermittelt Richtlinien, die der PIP abspeichert, oder fragt Richtlinien an. Die zu erstellenden oder zu ändernden Richtlinien werden vom PAP definiert. Dieser übermittelt die Richtlinien an den PDP, dessen Aufgabe die syntaktische Überprüfung ist. In speziellen Fällen können durch Richtlinien Bedingungen an das erlaubte Einreichen von Richtlinien des PAP gebunden werden. In diesem Fall ist es die Aufgabe des PDP, dieses ebenfalls zu überprüfen. Läuft die Überprüfung erfolgreich ab, reicht der PDP die Richtlinien bei dem PIP ein. Der PEP (auch „Wächter“ genannt) ist schlussendlich die beim PDP anfragende Exekutive, ob ein Zugriffsversuch richtlinienkonform ist. Schließlich vollstreckt der PEP die Entscheidung des Richtlinienentscheiders.

MASC-Zugriffskontrolle

In der MASC-Anwendung steht auf der einen Seite die Containereinheit, die batteriebetrieben eine möglichst große Lebensdauer haben soll. Die andere Seite bilden eine Vielzahl von Parteien, die sich Zugriff auf die Daten wünschen. Dabei soll aber zwischen den Rollen der Parteien beim Zugriff unterschieden werden und der Shipper soll (über den Spediteur) die Möglichkeit erhalten, Einfluss auf die Zugriffsrechte zu nehmen. Hinzu kommt die Tatsache, dass die Parteien variabel sind und sich gemäß des Vertragsbaums (siehe 1.1.3) transitiv und dynamisch zusammensetzen. Der Spediteur ist dabei der Startpunkt. Die besondere Situation ist hier, dass der PAP für die Richtlinien eines Transports auf mehrere Parteien verteilt ist. Zunächst bekommt der Spediteur das Recht, als PAP zu agieren. Er kann dann die Zugriffsrechte für Shipper, Versicherer und seine Vertragspartner erstellen und die Richtlinien dafür anlegen. Im Folgenden kann er seinen Vertragspartnern das Recht zusprechen, für bestimmte Zugriffsrechte selbst als PAP zu agieren. Auf diese Weise können alle Transportteilnehmer von ihrem Auftraggeber direkt berechtigt werden. Die Erstellung der benötigten Richtlinien folgt den Kanten des Vertragsbaums. Wie der PAP im Genauen aussieht, wird in Kapitel 5 beschrieben.

Es ist aus mehreren Gründen unklug, PEP, PDP und PIP in den Container zu verlagern. Erstens müsste bei jeder Richtlinienveränderung eine Kommunikation mit der MASC-Einheit aufgebaut werden. Jedes Verwenden der Sende-/Empfangseinheit kostet jedoch Energie und sollte auf ein Minimum reduziert werden. Zweitens müsste die Datenbanktabelle des PIP auch im Container gehalten werden. Externer Flashspeicher müsste aufgerüstet werden, um dem zusätzlichen Speicherbedarf gerecht zu werden. Das erhöht die Kosten, wäre aber grundsätzlich machbar. Drittens ist der PDP der algorithmisch und kommunikativ aufwändigste Teil der Richtlinienverwaltung[1] und benötigt seinerseits Arbeitsspeicher, Programmspeicher, Sende- und Rechenleistung. Ein Auslagern dieser Komponenten wäre eine Maßnahme, die die Containereinheit massiv entlastete. Das Auslagern von PDP und PIP würde das Zugreifen bei der Regelerstellung vermeiden und den Zugriff auf die Sensorabfragen beschränken.

[1] Der PDP ist der einzige, der mit allen anderen Teilnehmer kommuniziert, und nur er übernimmt die Überprüfungen der Rechtmäßigkeit.

Eine Grundvoraussetzung für das Auslagern von Diensten ist die Existenz einer Partei, der von allen Seiten (den Zugreifenden, den Containerbesitzern als Besitzer der MASC-Einheit und den Shippern, die ihre Regeln durchgesetzt bekommen möchten) dahingehend vertraut wird, dass sie die Aufgabe korrekt durchführt. Abschnitt 3.2 nimmt diesen Punkt auf und zeigt, dass diese vertrauenswürdigen Dritten im Transportwesen bereits existieren und eine Auslagerung der Dienste damit möglich ist.

3.1.2 Richtlinienvollstreckung

Es ist denkbar, den PEP beim Container zu belassen. Greift eine Partei auf diesen zu, muss die Containereinheit Kontakt mit dem ausgelagerten PDP aufnehmen, um die Zugriffsrechte überprüfen zu lassen. Diese Zusatzkommunikation kostet ebenfalls Energie, die eingespart werden könnte, wenn der PEP ebenfalls ausgelagert würde.

Definition 3.7 (Identifizieren)**.** *Einer Partei, einer Person oder einem Objekt wird ein Identifizierungsmerkmal bijektiv zugeordnet. Identifizieren bezeichnet den Vorgang der Vorlage des Identifizierungsmerkmals.*

Definition 3.8 (Authenti(fi)zieren)**.** *Das Glaubhaft-Machen einer Entität, dass ihr das eigene Identifizierungsmerkmal zugeordnet ist, nennt sich Authentisieren. Die erfolgreiche Überprüfung dieser Zuordnung wird als Authentifizieren bezeichnet. Die Herleitung von Authentisieren und Authentifizieren erfolgt in Anhang B.4.*

Die Basis der Identifizierung bildet das Identifizierungsmerkmal. Es kann bei Firmen ein Tupel aus Firmennamen, Firmensitz und Herkunftsland sein. Zur Authentisierung wäre die Vorlage eines Briefkopfes oder eine beglaubigte Bescheinigung einer offiziellen lokalen Behörde denkbar, die die Zuordnung vom Identifizierten zum Identifizierungsmerkmal bestätigt. Das Identifizierungsmerkmal muss dabei keine globale Gültigkeit haben (wie es bei Firmendaten oder Personalausweisnummer der Fall ist), sondern kann auch ein *temporäres Identifizierungsmerkmal* sein, das nur für den aktuellen Transport bestimmt ist. Solch ein temporäres Merkmal könnte über einen Namen wie „Frachtführer *x*“ definiert werden oder auf der Existenz eines eindeutigen öffentlichen Schlüssels[2] beruhen. Der Zugreifende müsste bei der Authentisierung dem Verifizierer glaubhaft machen, dass er der Frachtführer *x* beziehungsweise Besitzer des öffentlichen Schlüssels ist. Neben dieser expliziten Identifizierung wäre es auch denkbar, dass sich der Zugreifende *implizit identifiziert*. Dies könnte geschehen, indem er beim Authentisieren nachweist, ein ihm eigenes Geheimnis zu kennen.

Die Aufgabe des PEP ist das Ermitteln, auf welche Ressourcen der Zugriff einer Identität gewünscht wird, und das Anfragen bei dem PDP, ob der Zugriff gestattet ist. Dabei übermittelt der PEP dem PDP den Authentisierungsnachweis des Zugreifenden, der vom PDP überprüft wird. Es gibt verschiedene Ansätze, die Authentifizierung zu gestalten. Diese reichen von einem Benutzername-Passwort-Nachweis über geteilte Geheimnisse bis hin zu einer PKI (Erklärung siehe Anhang A.4). Die Vorteile einer PKI auf Basis von Zertifikaten ergeben sich, wenn viele Teilnehmer auf viele verschiedene

[2] Erklärung von öffentlichen Schlüsseln siehe Anhang A.3.2

Dienste zugreifen wollen, wie dies beim direkten Containerzugriff der Fall ist. Es müssten also nicht alle Container die Passwörter (oder Geheimnisse) aller potenziellen Teilnehmer kennen, sondern nur das Zertifikat (beziehungsweise den Verifizierungsschlüssel) eines vertrauten Dritten, der jedem Teilnehmer ein Zertifikat ausstellt. Bei Vorlage desselben kann sich jeder ausweisen und der Container kann die Unterschriften des vertrauten Dritten überprüfen. Allerdings ist die dafür benötigte Logik wegen der Verwendung asymmetrischer Kryptographie (vgl. A.3.2) vergleichsweise aufwändig (siehe Anhang A.3.3) und Testimplementierungen auf Sensorknoten haben ergeben, dass sie praktisch nicht realisierbar sind. Aber auch die Ausweichverfahren benötigen einen Zusatzaufwand, der bei Auslagerung des PEP entfiele. Da es bereits einen vertrauenswürdigen Server gibt, der PDP und PIP führt, ist es sinnvoll, auch den PEP dorthin auszulagern.

3.1.3 Ausgelagerte Richtlinienverarbeitung

Im Folgenden werden drei verschiedene Zugriffsmanagementsysteme für die MASC-Anwendung vorgestellt. Der Grad der Auslagerung der Richtlinienverarbeitung wird dabei schrittweise erhöht, um eine anschließende Evaluierung dieser Systeme durchführen zu können. Alle diese Dienste sind zeitlich unbeschränkt gültig und überdauern einzelne Transporte.

- **Proxyserver –** Der Container ist nur über einen Proxy zu erreichen. Der Proxy ist der PEP und vollstreckt die Erlaubnisentscheidung eines PDP, ob die Anfrage bezüglich der Richtlinien erlaubt ist. Bei Korrektheit leitet er die Anfrage an die Containereinheit weiter.
- **Kerberos-Server –** Nach der Anmeldung am Kerberos-Server (PDP) erhält der Zugreifende ein Ticket mit den Zugriffsrechten. Mit diesem Ticket kann er auf die MASC-Einheit (PEP) direkt zugreifen. Mit Hilfe von Kerberos werden PDP und PIP ausgelagert.
- **Spiegelserver –** Der Container kommuniziert nur mit dem Server (PDP, PEP, PIP) und übermittelt diesem alle Informationen. Der Server wird zum Datenbankspiegel der Containereinheit und regelt den Zugriff völlig selbstständig.

Proxyserver Das englische Wort „proxy“ bedeutet Bevollmächtigter. In der Informatik sind Proxyserver kontrollierende Rechner, über die der Datentransfer abläuft. Diese können Filter- oder Firewallfunktionen übernehmen oder zur Effizienzsteigerung dienen, weil sie einen Cache vorschalten. Jedem Container ist ein Proxyserver zugeordnet; beide teilen sich einen geheimen Schlüssel, mit dessen Besitznachweis sie sich dem anderen gegenüber ausweisen können. Der Proxyserver erhält die ankommenden Anfragen der Zugreifenden und leitet diese an einen ausgelagerten oder integrierten PDP weiter, der sie auf Rechtmäßigkeit prüft. Attestiert der PDP Richtlinienkonformität, leitet der Proxy die Anfrage des Zugreifenden an die Containereinheit weiter, wobei die Daten mit dem gemeinsamen Schlüssel verschlüsselt werden. Die Antwort geht über den Proxyserver zurück. Dieser kann zur Effizienzsteigerung Ergebnisse von Anfragen zwischenspeichern und muss somit nicht jede Anfrage weiterleiten, sondern kann kürzlich gestellte Anfragen selber beantworten.

Der Proxyserver könnte entweder nur PEP sein, PEP und PDP verbinden oder sogar PEP, PDP und PIP in einem vertreten. Diese Entwurfsentscheidung ändert wenig an der Funktionsweise für Zugreifende und Containereinheit. Wenn jedes Verkehrsmittel oder jeder Frachtführer seinen eigenen Proxy hat, dann ist es im Sinne des Shippers, dass dieser nicht PDP ist und die Erlaubnisentscheidungen selber trifft. Wenn aber Staaten oder Staatenbunde den jeweiligen Proxyserver stellen, kann es anders aussehen. Abschnitt 3.2 beschäftigt sich mit der Frage, an welcher Stelle die Richtlinienentscheidung idealerweise erledigt wird.

Kerberos-Server Kerberos ist ein häufig verwendetes „Single-Sign-On“-Verfahren, welches an vielen Universitäten und Firmen eingesetzt und in Abschnitt A.5 näher erklärt wird. Angewandt auf das Containerszenario melden sich die Zugreifenden ausschließlich an dem vertrauten Kerberos-Server (PDP und PIP) an. Dieser stellt dem Zugreifenden für den Containerzugriff ein Ticket aus, mit welchem er sich direkt an den Container wendet. Neben dem Ticket erhält der Zugreifende einen vom Server erstellten geheimen Schlüssel. Das Ticket enthält alle Zugriffsrechte des Zugreifenden und diesen geheimen Schlüssel und kann nur vom Container entschlüsselt werden. Wenn dieser das Ticket korrekt entschlüsselt hat, dient der enthaltene Schlüssel zur gesicherten Übertragung zwischen dem Zugreifenden und dem Container. Die Authentifizierung wird somit vom Kerberos-Server durchgeführt. Die Richtlinienvollstreckung wird letztendlich vom Container selbst erledigt, da der Kerberos-Server keine Kontrolle über den Kommunikationsfluss hat.

Der Kerberos-Server hat bereits die Authentifizierung übernommen und die Richtlinien bezüglich der Benutzer vom PIP angefordert. Diese extrahierten Zugriffsrechte werden so in das Ticket implantiert, dass nur der PEP (Containereinheit) diese lesen kann. Wenn sich der Zugreifende beim Container mit dem Ticket anmeldet, kann der PEP die Zugriffsrechte auslesen.

Spiegelserver Die dritte Variante ist der Spiegelserver, der die Verbindung zum Container transparent hält. Der Zugreifende kommuniziert ausschließlich mit dem Spiegelserver, der die Daten in der eigenen Datenbank speichert. Der Spiegelserver ist PEP, PDP und PIP und regelt den Zugriff selbständig. Die Containereinheit sorgt aktiv dafür, dass die Daten in der Datenbank aktualisiert gehalten werden. Jedes neue Ereignis wird dem Spiegelserver schnellstmöglich mitgeteilt.

Allen drei Varianten ist gemeinsam, dass sie auf einem ausgelagerten Server ausgeführt werden. Die Tatsache, dass dieser Server PDP, PEP und PIP ist, gibt dieser Partei eine zentrale Rolle im Sicherheitskonzept der MASC-Anwendung. Die ganze restliche Architektur kann beliebig sicher sein – wenn dieser Server den falschen Benutzern Zugriff auf die Daten gewährt, ist das gesamte Sicherheitskonzept unterwandert. In Abschnitt 3.2 wird dieses Problem aufgegriffen und gezeigt, wie dies in der Logistik gelöst werden kann.

3.1.4 Push oder Pull

Pull Englisches Wort für Ziehen. In der Informatik bezeichnet Pull das aktive Abholen von Daten. E-Mails sind ein klassisches Beispiel. Der E-Mail-Client fragt beim Server an und holt die E-Mails auf Kommando herunter.

Push Englisches Wort für Drücken, Schieben. Push bezeichnet in der Informatik einen Dienst, der die Daten asynchron zustellt, wenn sie anliegen. SMS sind ein Beispiel für einen Push-Dienst.

Shipper, Speditionen und Logistiker sollen Zugriff auf die Daten erhalten, wenn sie dies wünschen, daher ist eine Pull-Strategie wünschenswert. Der Proxyserver und der Kerberos-Server basieren auf dieser Strategie, bei der sich die Zugreifenden selbst bei der Containereinheit anmelden und die gewünschten Daten anfordern.

Pull-basiertes Hybridverfahren

Ein Ansatz für eine pull-basierte Zugriffskontrolle ist ein auf der Verwendung von Kerberos und dem Proxyserver fußendes Hybridverfahren (siehe [94]). Der Kerberos-Server übernimmt PDP und PIP und stellt den Zugreifenden Tickets aus. Mit diesen melden sich die Zugreifenden bei dem Proxyserver (PEP) an. Abbildung 3.2 veranschaulicht diesen Architekturentwurf.

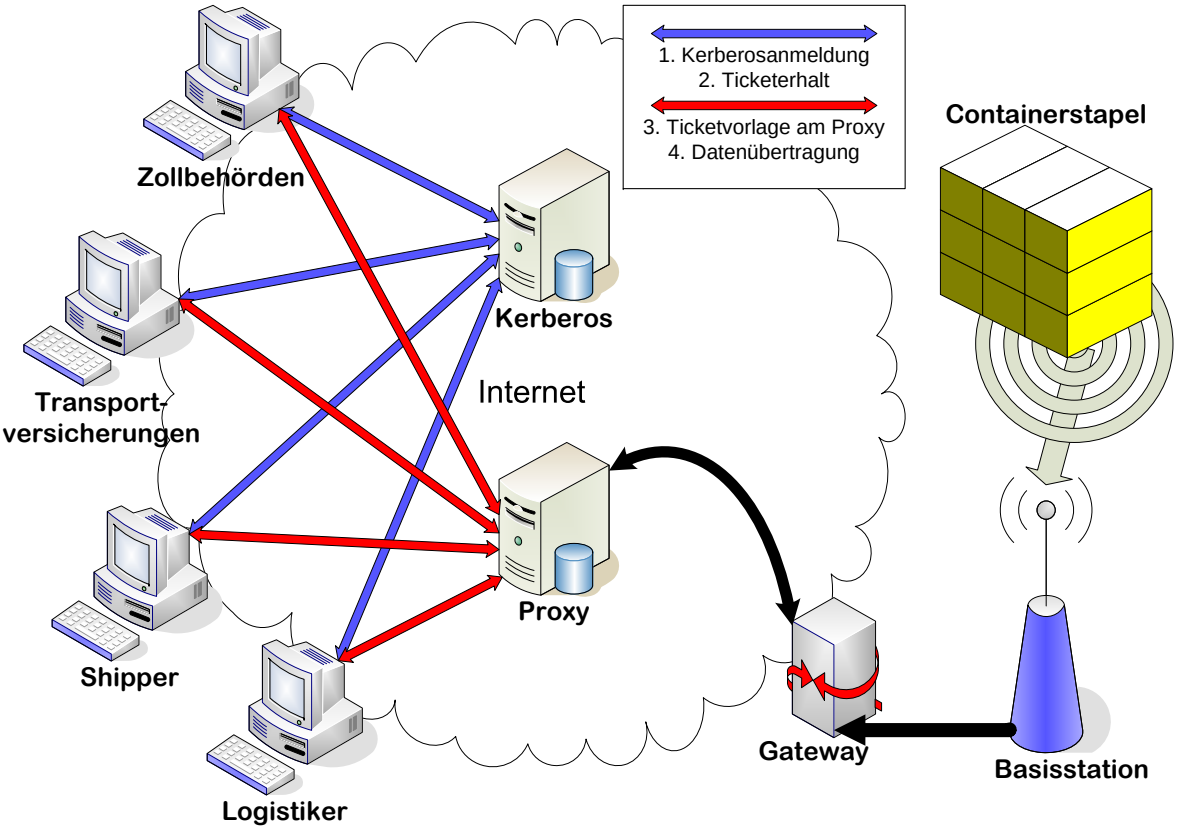


Abbildung 3.2: Kerberos-Proxy-Hybridverfahren

Der Proxyserver wertet das Ticket aus und aktualisiert den Ereigniszähler, der das letzte erzeugte Ereignis des Containers angibt. Zusammen mit dem Ereigniszähler übermittelt der Container (via Basisstation und Gateway) die betreffenden Sensoren, deren Daten im Ereignis enthalten sind. Wenn der Container keine Verbindung zu einem Proxy hat, wird der letzte Stand im Proxy belassen. Der Proxyserver teilt daraufhin dem Zugreifenden mit, welche Ereignisnummern zu Ereignissen gehören, die mit seinen Zugriffsrechten lesbar sind. Im Anschluss fordert der Zugreifende die genauen Ereignisse an. Wurde auf diese (kürzlich) bereits von anderen Teilnehmern zugegriffen, sind sie noch im Speicher des Proxy vorgehalten. Der Container muss nicht angefragt werden, sondern der Proxy kann die Anfragen selber aus seinem Speicher beantworten. Ist das Ereignis nicht zwischengespeichert, fragt es der Proxy beim Container an und legt die Antwort in seinem Zwischenspeicher ab, aus dem dann der Zugreifende bedient wird.

Es kann vorkommen, dass ein Ereignis aus mehreren Sensorschwankungen besteht, der Zugreifende aber nicht alle lesen darf. In dem Fall ist es die Aufgabe des Proxyservers, die Inhalte so zu filtern, dass nur die berechtigten Daten übertragen werden.

Durch die Einführung des Proxyservers entsteht ein einstellbares Mischsystem zwischen der Verwendung des Spiegelservers und des direkten Zugriffs. Je mehr Nachrichten vom Proxy zwischengespeichert werden, desto ähnlicher wird das Konzept dem des Spiegelservers. Werden gar keine Nachrichten zwischengespeichert, entspricht das System dem direkten Containerzugriff mit ausgelagertem PEP. Kerberos ist sowohl PDP als auch PIP und garantiert in allen Fällen die Datenintegrität und Vertraulichkeit.

Push-basierter Spiegelserver

Der Nachteil von pull-basierten Verfahren ist die Datenaktualität. Nur zum Zeitpunkt eines Zugriffsversuches wird der Container angefragt. In dem Fall, dass der Container sich nicht in Reichweite einer MASC-Basisstation befindet, kann der Proxy keine Aktualisierungen empfangen. Sollte sich die Einheit nur selten in MASC-Reichweite befinden, wäre es sinnvoll, wenn sie dann alle bis dahin angefallenen Ereignisse übertragen kann.

Nur bei Verwendung eines push-basierten Verfahrens ist es möglich, einen Niedrig-Energie-Modus zu aktivieren, wenn keine Ereignisse auftreten. Mit einem deaktivierten Empfänger kann die Containereinheit keine Nachrichten in Empfang nehmen. Weil die Sende-/Empfangseinheit der größte Stromverbraucher der Containereinheit ist, kann durch ein Abschalten der Einheit deren permanenter Energieverbrauch stark gesenkt werden.

Die MASC-Architektur besteht also aus einem zentralen Spiegelserver, der als einziger Vermittler PDP, PEP und PIP übernimmt, um den Zugriff auf die Daten zu verwalten. Dieser zentrale Server wird folgend MASC-Gegenstelle genannt. Der Zugriff steuernde Dienst heißt MASC-Informationsdienst. Eine darstellende Abbildung der Spiegelserverarchitektur zeigt Abbildung 3.3.

3.2 MASC-Gegenstelle

Die Aufgabe der MASC-Gegenstelle ist es, die Daten der Container für die Zugreifenden bereitzustellen. Gegenüber den Zugreifenden bietet es den MASC-Informationsdienst (MASC-ID) an, der die Zugriffskontrolle regelt. Auf der anderen Seite teilt sich die Gegenstelle einen geheimen Schlüssel mit der ihr eindeutig zugeordneten Containereinheit. Die Annahme ist, dass keine weitere Partei diesen Schlüssel kennt (und auch nicht kennen darf). Jeder Statusreport oder jede Alarmierung, die die MASC-Einheit aussendet, hat die Gegenstelle zum Ziel. Auf diese Weise dient die Gegenstelle als Informationssenke für die MASC-Einheit. Die MASC-Gegenstelle nimmt somit die zentrale Rolle zwischen den Zugreifenden und der Containereinheit ein und ist nicht zuletzt durch Beherbergung von PEP, PDP und PIP ein wesentlicher Bestandteil des Sicherheitskonzeptes.

Der Frachtführer könnte unter Umständen kein Interesse daran haben, den Shipper von einem vermuteten Schaden in Kenntnis zu setzen. Sollte sich kein Schaden in der Ware finden, wäre eine Alarmierung „viel Wirbel um nichts“, die den Transportablauf stören und unnötige Kosten verursachen würde. Falls dabei kein sichtbarer Schaden entstanden ist, würde der nächste Interchange reibungslos verlaufen. Sein Interesse ist es nicht, sich selbst unter Umständen zu belasten. Er scheidet als vertrauenswürdiger Dritter somit aus. Der Containerbesitzer könnte als Informationsverteiler fungieren, wenn er mit keinem der anderen Partner in Zusammenhang steht. Oftmals sind aber Containerbesitzer und Reeder ein und dieselbe Partei. Gleiches gilt für die Spedition: Sie könnte ebenfalls im Interessenkonflikt stehen, da sie oftmals auch der Frachtführer einer Teilstrecke ist. Es fehlt auch hier die benötigte Unabhängigkeit.

Die Versicherungen könnten auch die Gegenstelle beherbergen. Bei Consolidation-Containern ist häufig zu beobachten, dass mehrere Versicherungen die Ladung versichern – es kann aber nur eine Gegenstelle geben. Die Versicherungen müssen sich in diesem Fall einigen. National wäre eine Konsortiumsbildung denkbar, international sind die Absprachen der Versicherungen aber noch nicht ausgebildet genug. Zu klären bleibt die Frage, wie die Versicherungen den geheimen Schlüssel in der Containereinheit platzieren können. Die Versicherungen sind nicht in die Transportabläufe involviert und bieten sich somit nur bedingt zur Bereitstellung der Gegenstelle an. einhei Die Zollbehörden könnten ebenfalls eine Gegenstelle anbieten, allerdings gibt es viele von ihnen – welche soll für welchen Container zuständig sein? Abhilfe würde ein zentraler Gegenstellen-Server bei der IMO schaffen. Dieser wäre für die Zuordnung von Gegenstelle zu Transport zuständig. Allerdings muss dieser eine große Datenlast aushalten und wäre ein Single-Point-Of-Attack[3].

Es erscheint also sinnvoll, einen unabhängigen Partner zu wählen, der bestenfalls bereits ein vertrauenswürdiges Verhältnis mit den meisten Partnern pflegt. Im Bereich des Schiffs- und Containerwesens gibt es diese vertrauenswürdigen Dritten bereits. Es sind die RSOs, die überprüfen, ob die internationalen Bestimmungen bezüglich des ISPS-Codes eingehalten werden (siehe Abschnitt 1.2.1). Diese RSOs

[3] Single-Point-Of-Attack beschreibt die Gefahr, dass es in dem System einen Punkt gibt, der im Angriffsfall das ganze System lahmlegt.

überprüfen derweil die Terrorabwehrmaßnahmen der Häfen und Schiffe, was die Vertrauensstellung gegenüber den Zollbehörden ebenfalls erhöht, da sie für genau diesen Bereich bereits eingesetzt werden. Es ist nur logisch, die Gegenstelle und den integrierten MASC-ID bei den RSOs unterzubringen. Da jeder Container von einer Zertifizierungsstelle abgenommen werden muss, können die RSOs die korrekte Implementierung der MASC-Einheit überprüfen, den gemeinsamen Schlüssel festlegen und die Zuständigkeit für den MASC-Container somit übernehmen.

3.3 MASC-Anwendung

Dieser Abschnitt fasst noch einmal alle beschriebenen Einzelteile zusammen, um das Komplettbild der MASC-Anwendung kompakt darzustellen. Anschließend wird die Netzwerkinfrastruktur genauer beschrieben, damit eine bessere Risikobetrachtung durchgeführt werden kann.

3.3.1 Anwendungsbeschreibung

Um den Container auf seiner gesamten Transportkette zu überwachen, enthält dieser bereits ein implantiertes Gerät (MASC-Einheit), mit dessen Hilfe das Geschehen im Innenraum verfolgt werden kann (siehe Abschnitt 2.3). Jedes Ereignis wird im Datenspeicher der Containereinheit abgelegt, um mit Hilfe dieser Informationen die Verantwortlichkeiten im Schadensfall besser zuordnen zu können.

Der Transport beginnt und der MASC-Container wird von verschiedenen Frachtführern und Transportmitteln befördert. Während des gesamten Transportes sollten bestenfalls MASC-Infrastrukturen vorhanden sein – da dies aber besonders in der Anfangsphase utopisch erscheint, reicht es aus, wenn der Container immer wieder in Bereiche gelangt, in denen eine Infrastruktur ihm ermöglicht, seine Ereignis-Historie an die MASC-Gegenstelle zu übertragen.

Bei der Datenübertragung gilt es allerdings vieles zu beachten, weswegen das Datenübertragungsprotokoll in einem eigenen Kapitel behandelt wird. Die Bezeichnung für dieses Protokoll ist MASC-Secure-Tunnel und wird in Kapitel 4 beschrieben.

Die Gegenstelle (Abschnitt 3.2) spiegelt den kompletten Datensatz der Containereinheit und hält die Daten auch nach dem Transport verfügbar. Da Schadensentdeckung und Regressforderungen oftmals lange dauern, sind die Containerereignisse für die zugreifenden Parteien noch eine ganze Weile von der Gegenstelle abrufbar.

Der Zugriff auf die Daten geschieht dabei über den MASC-Informationsdienst (MASC-ID), der das Zugriffsmanagement für die Zugreifenden regelt. Da der Spediteur die Transportkette erst nach und nach zusammenstellt und Unterauftragnehmer eigenständig Teile organisieren, muss das Zugriffsmanagement diese Transportbeauftragung gut abbilden können. Weil diese Vertragsbildung extrem dezentral abläuft, sollte das Zugriffsmanagement ebenfalls diese Dezentralität unterstützen. Zur Verbesserung der formellen Prozesse unterstützt der MASC-ID auch die Automatisierung der Ladungspapiere und

Zolldokumente. Kapitel 5 beschreibt die detaillierte Funktionsweise und die genauen Anforderungen an diesen Dienst.

3.3.2 Netzwerkinfrastruktur

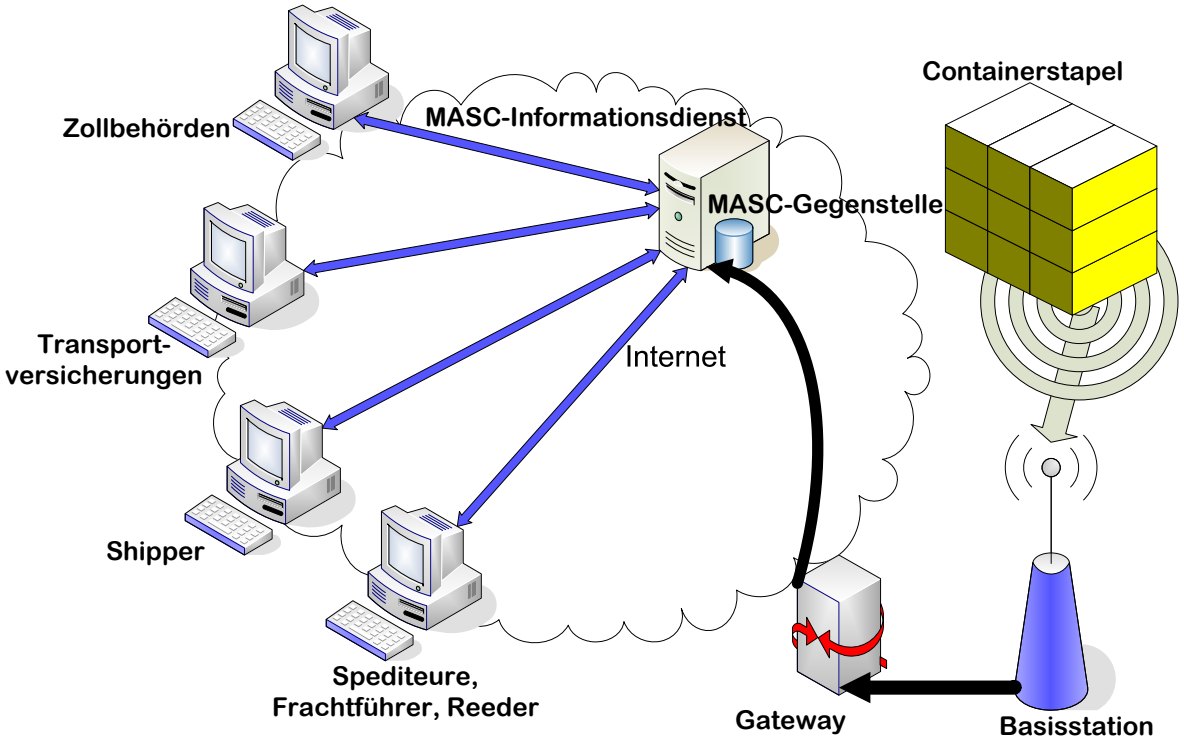


Abbildung 3.3: MASC-Infrastruktur

Bild 3.3 veranschaulicht die resultierende Infrastruktur. Der exemplarisch im Stapel stehende MASC-Container ist dabei die Quelle des Datenflusses. Der Container kann sich dabei gestapelt auf Schiff, Containerdepot oder -yard oder alleine auf einem LKW oder Eisenbahnwaggon befinden. Die Sofortbenachrichtigung funktioniert bei MASC nur dann, wenn eine MASC-Basisstation in Reichweite ist. Diese sind dabei nur Sende-/Empfangseinheiten, die Nachrichten entgegennehmen, senden oder an den Gateway weiterleiten. Jede Basisstation ist mit einem Gateway verbunden, allerdings wird nur ein Gateway pro Transportmittel oder Stellplatz benötigt.

Die MASC-Einheit sendet eine Nachricht[4] und versieht diese dabei mit der DNS-Adresse der betreffenden MASC-Gegenstelle. Die Basisstation leitet diese Nachricht blind an den Gateway weiter. Der Gateway speichert die Absender-ID und die Basisstation, von welcher die Nachricht gekommen ist. Die Empfänger-DNS-Adresse wird durch eine DNS-Abfrage in eine IP-Adresse umgewandelt. Der Gateway erzeugt einen TCP-Datenstrom mit der IP-Adresse des Empfängers und seiner IP-Adresse als Absender. Der Datenstrom erhält die Container-ID, die verschlüsselte Nachricht, einen Zeitstempel

[4]Das genaue Nachrichtenformat und das zu Grunde liegende Protokoll werden in Kapitel 4 erläutert.

und die aktuellen GPS-Koordinaten des Gateways. Letztere sind notwendig, um eine exakte Positionierung des Container(stellplatze)s durchführen zu können. Kommt die Antwort der MASC-Gegenstelle, ist ebenfalls die Container-ID im Klartext beigefügt, damit der Gateway die Nachricht an die korrekte Basisstation weitergeben kann. Auf diese Weise wird das Routing sichergestellt.

Die MASC-Gegenstellen sind bei den RSOs platziert und somit gibt es nicht nur eine. Allerdings hat jeder MASC-Container genau eine, die ihm zugeordnet ist. Für jeden Container verwaltet dieses Computersystem bei der RSO eine Datenbank mit mehreren Datenbanktabellen. Eine Datenbanktabelle beinhaltet die Containerereignisse und ist somit der Spiegel der MASC-Einheit. Auf diese Tabelle hat nur die MASC-Gegenstelle Schreibrechte und füllt sie entsprechend der ankommenden Daten. In den weiteren Datenbanktabellen sind die geheimen Schlüssel, die Richtlinien und die Informationen für das elektronische Containermanifest enthalten. Richtlinien und Manifestinformationen sind wiederum für den MASC-Informationsdienst von Bedeutung, der dafür sorgt, dass die zugreifenden Parteien Zugriff auf die Containerdaten erhalten.

Der MASC-Informationsdienst sollte nicht unbedingt auf dem gleichen Server wie die MASC-Gegenstelle beheimatet sein. Er steht aber definitiv bei der gleichen Partei und benötigt Zugriff auf dieselbe Datenbank. Der MASC-Informationsdienst beginnt seine Tätigkeit bei der Anmeldung des MASC-Transportes durch die organisierende Spedition. Er sorgt auch dafür, dass die initialen Richtlinien erstellt werden, und regelt den Zugriff auf die verteilten Daten. Die Spedition, die den Transport anmeldet, ist dabei diejenige Partei, die in erster Instanz die Rechtezuordnung durchführt. Genaueres enthält Kapitel 5.

3.4 Sicherheitsbetrachtung

Aus Sicherheitsgründen ist es wichtig, Risikofaktoren schon zur Entwurfszeit zu berücksichtigen. Gefahren, die erst später deutlich werden, sind meist nicht mehr adäquat zu bekämpfen. Die folgenden Risikobetrachtungen sollen demonstrieren, welche Angriffsszenarien denkbar sind und wogegen sich die MASC-Anwendung schützen muss.

Bild 3.4 zeigt die Bereiche, in denen Angriffe erfolgen können. Angriffspunkt 1 ist das Container-Innere. Zwei Arten von Angriffen können aus dem Inneren lanciert werden: Zum einen kann ein Angreifer ein Gerät zur Ware hinzulegen. Dieses könnte das korrekte RFID-Auslesen stören oder bei drahtloser innerer Sensoranbindung die Datenübermittlung angreifen. Zum anderen ist ein direkter Angriff auf die Hardware in der MASC-Einheit denkbar. Angriffspunkt 2 bezieht sich auf den Datentransport von Nachrichten von der MASC-Einheit zur MASC-Gegenstelle. Der Datentransport durchläuft dabei die Luftschnittstelle zwischen Container und Basisstation, die Basisstation-Gateway-Schnittstelle und die Internetroute zwischen Gateway und MASC-Gegenstelle. Überall sind verschiedene Angriffe denkbar. Angriffspunkt 3 ist die RSO, wo Gegenstelle und MASC-ID-Server beheimatet sind. Angriffspunkt 4 ist die Kommunikation der Gegenstelle mit den Zugreifenden über den MASC-Informationsdienst. Auf

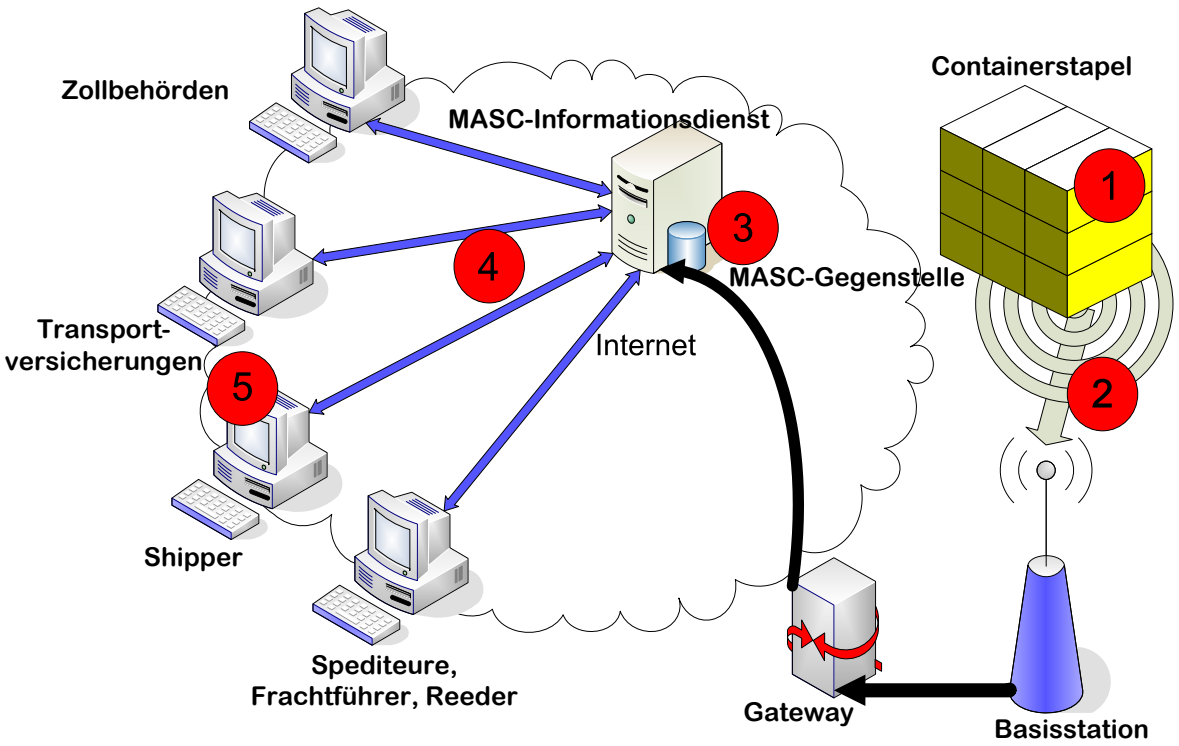


Abbildung 3.4: Angriffspunkte

Angriffspunkt 5, die Workstations der Interessenten, wird in Abschnitt 6.3 eingegangen. Das Rechnersystem der Teilnehmer sollte zwar gegen einen externen Angriff abgesichert sein, allerdings liegt dies im Verantwortungsbereich der einzelnen Unternehmen. Die Gefahr, dass indiskrete Mitarbeiter sensible Informationen weitergeben, kann durch technische Sicherungsmaßnahmen nicht ausgeschlossen werden.

3.4.1 Container-Inneres

Durch die Stahlwände des Containers ist zwar das Container-Äußere von dem Container-Inneren getrennt, jedoch können Angriffe aus dem Container-Inneren nicht ausgeschlossen werden. Es gibt bereits heute so genannte Hamster, die einem Container zur Innenraumüberwachung der Fracht beigegeben werden. So wird diese elektronische Einheit zum Beispiel einem Bananen-Karton zur ständigen Überwachung des Temperaturverlaufs hinzugefügt. Ferner wäre es zu Vertuschungszwecken für Frachtführer und Einbrecher interessant, die Spuren zu verwischen, wenn es zu Diebstahl oder Beschädigung der Fracht gekommen ist. Ebenfalls könnte es für Terroristen von Interesse sein, die Radioaktivitäts- und Zugriffserkennung des Containers zu deaktivieren.

Sensoranbindung

Bei einer drahtlosen Sensorenanbindung im Container-Inneren müssen weitere Angriffe berücksichtigt werden. Die Sensoren verfügen im Normalfall über einen analogen Ausgang, der den Wert der Sensoren zeigt. Dieser wird per Kabel an die MASC-Einheit übertragen, dort per Analog-Digital-Wandler digitalisiert und weiterverarbeitet. Wenn die Sensoren drahtlos im Container verteilt werden sollen, müssen diese ebenfalls mit einem Analog-Digital-Wandler ausgerüstet werden, um die digitalen Informationen verschlüsselt zur MASC-Einheit senden zu können. Sollten diese nicht verschlüsselt werden, wäre es ein Leichtes, die Sensorinformationen durch die Verwendung von modifizierten Hamstern auszulesen und darüber hinaus sogar falsche Sensordaten zu senden. Bei Verwendung drahtloser ungeschützter Sensorkommunikation kann die MASC-Einheit nicht erkennen, von welchem Sensor die digitalisierten Messwerte kommen. Ferner steigt der Stromverbrauch durch das Einführen der drahtlosen Sensoren eminent. Die MASC-Einheit benötigt eine innere Antenne und jeder Sensor muss zusätzlich drahtlose Sendeleistung aufbringen und gleichzeitig die Daten verschlüsseln.

Wenn die Sensoren durch zugängliche Kabel verbunden sind, laufen die analogen Sensorsignale ebenfalls ungeschützt über diese Datenleitung. Ein Angreifer könnte das Kabel durchtrennen und einen eigenen Sensor dort anschließen. Es wäre daher optimal, wenn alle Sensoren direkt in die MASC-Einheit integriert werden können. Wenn dies nicht mit allen Sensoren durchgeführt werden kann, sollte es zumindest für die Wichtigsten erfolgen.

MASC-Einheit

Die MASC-Einheit generiert Ereignisse, wenn signifikante Sensorschwankungen gemessen werden. Diese Ereignisse werden sowohl im lokalen Speicher als auch auf dem Spiegelserver (der MASC-Gegenstelle) gespeichert. Allerdings dürfen diese Daten nicht in die Hände von Angreifern gelangen, da sie vertrauliche Informationen über die Ware und deren Zustand enthalten. Da das Auslesen von Speicherzellen für einen Angreifer, der physischen Kontakt zur Hardware erhält, ein Leichtes ist, muss die MASC-Einheit adäquat gegen Fremdauslesen gesichert werden. Des Weiteren sollte jedweder Zugriffsversuch entdeckt werden, damit Manipulationen ausgeschlossen werden können.

Um ein unbefugtes Auslesen der Hardware zu verhindern, müssen die Ereignisse im Speicher verschlüsselt und der Schlüssel gut gesichert werden. Der Ablauf der Verschlüsselung wird in Abschnitt 4.3 erläutert. Die Sicherungsmaßnahmen der MASC-Einheit gegen Fremdzugriff werden in Abschnitt 6.1 behandelt. Das benötigte Initial-Schlüsselmaterial wird bereits vor der Implementierung der Einheit in den Container auf die Einheit geschrieben. Dies geschieht unter Kontrolle der RSOs, die den Container verifizieren. Auf diese Weise ist sichergestellt, dass nur die RSO das Schlüsselmaterial in der Einheit kennt.

3.4.2 Container-Gegenstelle-Schnittstelle

Ein offensichtlicher Angriffspunkt ist die drahtlose Übertragung zwischen Container und Basisstation. Alle Alarmierungen und Statusmeldungen des Containers gehen über diesen Kanal. Diese Informationen sind für die Konkurrenten von hohem Interesse, weil die Kenntnis dieser Informationen zu einem Marktvorteil führen kann. Der Konkurrent könnte das Verhalten des Angegriffenen abschätzen und entsprechend reagieren. Die Daten müssen also gegen **Abhören** gesichert sein.

Der Konkurrent hätte sogar einen noch größeren Nutzen, wenn er die Informationen verfälschen beziehungsweise komplett falsche Daten senden kann. Wenn es also der Konkurrent schafft, sich mit einem seiner Container als der Container des Angegriffenen auszugeben, könnte er gefälschte Schadens-Alarmierungen durchführen und so den Konkurrenten zu teurem Fehlverhalten zwingen. Es muss also sichergestellt werden, dass der Absender **authentisch** ist.

Weitere Angriffsszenarien sind denkbar, bei denen Terroristen oder einfach nur involvierte Frachtführer die potenziellen Angreifer sind. **Nachrichtenintegrität** und **Vertraulichkeit** müssen auf dieser Schnittstelle gewahrt bleiben.

In Abschnitt 4.3 wird erläutert, wie die Absicherung dieser Schnittstelle in der MASC-Anwendung realisiert wird.

Denial-of-Service

Mit Denial-of-Service (DoS) wird ein Angriff bezeichnet, der das Lahmlegen eines Dienstes zum Ziel hat. Im betrachteten Fall liegt ein erfolgreicher DoS-Angriff vor, wenn der Container keine Mitteilungen mehr an die Gegenstelle senden kann oder die Batterien leer sind und die MASC-Einheit ausfällt.

Die Kommunikation auf der Luft-Schnittstelle zu stören, ist dabei ziemlich leicht, da der Zugang zum Übertragungsmedium unbeschränkt ist[5]. Jeder, der in Reichweite ist, kann Signale auf das Medium senden. Die Reichweite ist dabei ebenfalls nicht beschränkt, da sie lediglich von der Sendeleistung des Senders abhängt. Mit einem Störsender (engl. *Jammer*) ist es daher ohne Weiteres möglich, das verwendete Frequenzband so zu stören, dass keine Kommunikation mehr möglich ist. Dieser Angriff ist nicht zu verhindern. Des Weiteren sind Angriffe gegen Basisstation, Gateway und Gegenstelle denkbar, die zu einem DoS führen.

Sollte das MASC-System zur Anwendung kommen, werden besonders in der Anfangsphase nur wenige Schiffe und Containeryards an die MASC-Infrastruktur angeschlossen sein. Es ist davon auszugehen, dass sich ein Container häufiger außer Reichweite einer Basisstation befindet, als dass er seine Daten senden kann. Dazu kommt, dass die Luftschnittstelle besonders rauschanfällig ist und deswegen trotz angeschlossener Infrastruktur kein korrekter Datenempfang möglich sein kann. Es sollte davon ausgegangen werden, dass die Verbindung keinesfalls permanent besteht. Dadurch ist das Entdecken

[5] Die Zugänglichkeit hängt vom Containerstellplatz ab. Ein Schiff auf hoher See ist schwerer zu erreichen als ein Schiff in Hafennähe oder ein LKW auf der Autobahn.

von DoS-Angriffen relativ schwierig, weil niemand weiß, ob die Einheit einfach keine Netzwerkverbindung hat oder ob ein Angriff auf die Datenübertragung vorliegt.

Angriffe gegen die Batterie sollten hingegen nicht möglich sein. Es sollte vermieden werden, dass der Angreifer die MASC-Einheit zu massiver unnötiger Arbeit bringt. Natürlich könnte sich ein Angreifer als Basisstation ausgeben und damit die Container dazu bringen, eine Kommunikation über die falsche Basisstation aufzubauen. Es darf aber nicht dazu führen, dass eine falsche Basisstation die MASC-Einheit zwingt, so viele Rechenoperationen durchzuführen, dass die Lebensdauer merklich sinkt. Abschnitt 4.3 greift beim Protokollentwurf diesen Punkt auf.

3.4.3 Gegenstelle

Die Gegenstelle an sich kann durch moderne Servertechniken gegen die meisten Angriffe gut geschützt werden. Es handelt sich bei der Gegenstelle lediglich um einen Internetdienst, der gleichermaßen abzusichern ist wie ein HTTP-Server. Die Gefahr eines verteilten DoS-Angriffs ist natürlich nicht auszuschließen. Um die Auswirkungen eines solchen Angriffs zu minimieren, sollten die Gegenstellen für die MASC-Einheiten dezentralisiert werden.

Die Internet-Schnittstelle zu den Zugreifenden (MASC-ID) ist gleichermaßen angreifbar. Die Vertraulichkeit der Datenbankabfragen muss auch hier gewährleistet bleiben. Auch sollte die Integrität gewahrt werden, sowohl die Identität der Zugreifenden als auch die Datenkorrektheit müssen stimmen. Der MASC-ID sollte auch verfügbar sein, allerdings gelten für ihn die Gesetze des Internets, da es sich um einen normalen Internetdienst handelt – auf die Verfügbarkeit lässt sich also nur begrenzt Einfluss nehmen.

4 MASC-ST – Erstellung des sicheren Tunnels

MASC-SecureTunnel (MASC-ST) ist die Bezeichnung für das Datenaustauschprotokoll zwischen der MASC-Einheit im Container und der MASC-Gegenstelle im Internet. Zwischen den beiden Kommunikationsendpunkten liegen die Basisstation und der Gateway. Letzterer ist dafür zuständig, die Umsetzung des lokalen Datentransportes auf das Internetprotokoll zu übernehmen und Statusinformationen hinzuzufügen. Jedoch ändert er nichts an den gesendeten Nutzdaten der Containereinheit, der Hauptteil des Datenstroms bleibt also unverändert.

MASC-ST basiert gemäß Abschnitt 3.1 auf einer symmetrischen Verschlüsselung. Das Initial-Schlüsselmaterial ist ausschließlich RSO sowie Container bekannt. Basierend auf diesem geteilten Geheimnis ist es die Aufgabe des MASC-ST, eine Transportverschlüsselung herzustellen.

4.1 Problemstellung

Die Anforderungen an dieses Protokoll sind ein niedriger Energieverbrauch der MASC-Einheit (siehe 2.3), ein vertraulicher Datenaustausch, bei dem die Datenintegrität gewahrt bleibt, und ein Verhalten, welches DoS-Angriffe nicht unterstützt (siehe 3.4.2). Wie bereits erwähnt, können DoS-Angriffe nicht verhindert werden (siehe 3.4.2), aber sie sollten dem Angreifer möglichst erschwert werden.

Der Informationsfluss ist quasi unidirektional. Zwar gibt es Statusmeldungen, Anfragen, Protokoll-Handshakes und Bestätigungen, die auch zur MASC-Einheit übertragen werden, dennoch haben alle Nutzinformationen von der MASC-Einheit die Gegenstelle zum Ziel. Ausnahmen bilden die Managementanfragen der Gegenstelle, die zur Erstellung neuer Sitzungsschlüssel oder zum Zurücksetzen von Zählervariablen führen. Datenintegrität ist beidseitig erforderlich. Vertraulichkeit ist für das Übermitteln der Ereignisse sowie für die Anfragen der Gegenstelle vonnöten.

4.2 Analyse bestehender Protokolle

Sensornetzwerke

Sensornetzwerke sind seit mittlerweile einem Jahrzehnt ein Thema, welches weltweit viele Forschungsbereiche beschäftigt. Seit Jahren gibt es immer neue Protokolle und auch die Hardware ändert sich fortlaufend. Angestoßen wurde die Forschung durch eine militärische Anwendung, bei der miniaturisierte Knoten über Feindesgebiet abgeworfen werden sollen, wobei die Forschung mittlerweile eine gewisse Eigendynamik entwickelt hat. Die Aufgabe der militärischen Knoten ist der Aufbau eines autarken

Netzwerks, welches zur Verteilung der gemessenen Daten genutzt wird. Das auf diese Weise gemeinsam erstellte Bild der Lage kann vom Militär abgerufen werden, indem es zu einem Sensorknoten Kontakt aufnimmt. Probleme, die mit der Miniaturisierung auftreten, beschäftigen die Forschung sehr. Die Sorge vor eingefügten falschen Knoten oder weitaus potenteren Angreifern, die den ungeschützten Datenverkehr manipulieren, hat die Forschung der Sicherheit in Sensornetzwerken angetrieben. Ein immer noch ungelöstes Problem ist das sichere Routing von Nachrichten durch diese Netzwerke (ISO/OSI-Schicht 2). Alle bekannten Sicherheitsprotokolle wirken sich auf die darüber liegenden Schichten aus und bekämpfen nur die Symptome. Aber auch die neuen Gefahren, die durch das Verwenden von Meshing (vgl. Abschnitt 2.4.1) eingeführt werden, sind weitere Ansätze für neue Sicherheitsprotokolle.

Inzwischen gibt es einige wenige Anwendungen, die es sich zu Nutze gemacht haben, dass immer mehr Hardware kostengünstig zu erwerben ist. Umweltmessungen [87] in Wäldern, Gletschern [65] oder an Vulkanen [84] werden vielerorts mit Sensornetzwerken durchgeführt. Erste Innenraumüberwachungs- und Haushaltssteuerungsprojekte [86] werden heute mit Sensorknoten realisiert.

Ein weiterer Forschungsbereich ist in den letzten Jahren aufgekommen: Die Auto-Netzwerke [33, 35]. Dabei sollen Autos zu Netzwerkknoten werden, die Staumeldungen, Meldungen über das eigene Fahrverhalten, aber auch Dienste wie Internet und Verkehrsleitung anbieten sollen. Wie die anderen angesprochenen Anwendungstypen basiert dieses ebenfalls auf Meshing (siehe Abschnitt 2.4.1). Einige existierende Algorithmen und Protokolle können dabei von der „Militäranwendung“ übernommen werden, andere müssen neu entworfen werden.

Gleichermaßen verhält es sich mit der Containerüberwachungsanwendung, wie sie von dieser Arbeit vorgestellt wird. Es handelt sich dabei ebenfalls um ein Sensornetzwerk, allerdings nicht um die „typische“ Militäranwendung, sondern um einen Vertreter mit sehr unterschiedlichen Voraussetzungen und Anforderungen. Die sich ergebende Frage ist, wie gut die vielen Sicherheitsprotokolle für Sensornetzwerke für die MASC-Anwendung geeignet sind.

Schlüsselverteilung in Sensornetzwerken

Um verschlüsselte Verbindungen in Sensornetzwerken aufzubauen, ist es wichtig, dass jede Partei den korrekten Schlüssel hat. Daher sind die Schlüsselverteilungsalgorithmen in Sensornetzwerken grundlegender Bestandteil der sicheren Kommunikation. Die folgenden Verfahren sind die bekanntesten Schlüsselverteilungsverfahren der Sensornetzwerkforschung und beschreiben exemplarisch einen sicherheitsrelevanten Forschungsschwerpunkt. (Vgl. [66])

Das *Key Management Scheme for Distributed Sensor Networks* von Eschenauer und Gligor [31] ist ein probabilistischer Ansatz, um bei einer Vielzahl von Knoten paarweise Schlüssel zu verteilen. Gleiches gilt für das *Random Key Predistribution Schemes for Sensor Networks* von Chan, Perrig und Song [19]. Zhu, Xu, Setial und Jajodia beschreiten mit ihrer Arbeit *Establishing Pair-wise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach* [107] einen ganz ähnlichen Weg, wobei

wenig Speicher verwendet wird, was allerdings auf Kosten einer Zufallszahlgenerierung pro Nachricht geht. Hwang und Kim haben mit *Revisiting Random Key Pre-Distribution for Sensor Networks* [41] mehr Wert auf Energiesparen gelegt, indem sie eine variable Reichweitensteuerung eingeführt haben.

Andere Papiere stellen deterministische Schlüsselverteilungsverfahren vor. Blundo, Santis, Herzberg, Kutten, Vaccaro und Yung arbeiten bei ihrem *Perfectly-secure key distribution for dynamic conferences* [12] mit zweidimensionalen Polynomen so, dass jeder Knoten mit jedem anderen einen Schlüssel basierend auf der Parameter-Variation dieser Polynome teilt.

Liu and Nings Arbeit *Establishing pairwise keys in distributed sensor networks* [58] ist ein Hybrid-Schema, welches den probabilistischen Ansatz mit den Polynomen kombiniert.

Law, Corin, Etalle und Hartel bauen mit *A Formally Verified Decentralized Key Management Architecture for Wireless Sensor Networks* [55] verschiedene Cluster auf, von denen einige überwacht sind und einen Anführer haben und andere wiederum gleichberechtigt sind. Die Sicherheit basiert hier auf dem Finden sicherer Routen.

Alle diese Arbeiten versuchen die Schwächen des Meshing auszugleichen. Das Hauptaugenmerk basiert auf der Kommunikation der Knoten untereinander. In der MASC-Anwendung muss allerdings die Kommunikation zwischen einem Server außerhalb des Netzwerks mit den Knoten im Netzwerk abgesichert werden. Die Knoten untereinander müssen nicht kommunizieren. Die zahlreichen Schlüsselverteilungsalgorithmen können also nicht als Basis für den sicheren Tunnel herhalten. Die Schlüsselverteilung bei der MASC-Anwendung ist auch bereits implizit erfolgt, allerdings muss ein effizientes Protokoll gefunden werden, welches in dem Sensornetzwerk benutzt werden kann.

Kommunikationsprotokolle für Sensornetzwerke

Einen Schritt weiter als die reinen Schlüsselverteilungsalgorithmen gehen die Kommunikationsprotokolle, die die Schlüsselverteilung sowie die verwendeten Protokolle zusammenfassen. Aus der VPN-Welt ist bereits LEAP bekannt, was von Zhu, Setia und Jajodia in *LEAP: Efficient Security Mechanisms for LargeScale Distributed Sensor Networks* [107] angepasst wurde. Dieses Sicherheitssystem wäre auf die MASC-Anwendung übertragbar, jedoch teilen sich alle Knoten einen gemeinsamen Schlüssel, auf dem die weitere Verbindungssicherheit basiert. Da beim Containerszenario aber von einem heterogenen, nicht-vertrauenswürdigen Umfeld auszugehen ist, passt dieser Ansatz nicht.

Khalil und Bagchi teilen bei *SECOS: Scalable and Energy Efficient Crypto On Sensors* [46] das Netz in Gruppen ein, die wiederum hierarchisch aufgeteilt werden. Datenaustausch wird immer von der übergeordneten Hierarchie-Stufe kontrolliert. Auch hierbei basiert die Organisation des Netzwerkes auf einem kollaborativen Ansatz, was ebenfalls ungeeignet ist, um im heterogenen Umfeld der Containerüberwachung eingesetzt zu werden.

Die Arbeit *SPINS: Security Protocols for Sensor Networks* von Perrig, Szewczyk, Tygar, Wen und Culler besteht aus zwei Protokollen: SNEP und μTESLA. SNEP regelt die sichere Kommunikation mit

der Basisstation und μTESLA sorgt für ein authentifiziertes Broadcasting. Einzig SNEP löst also genau die Aufgabe, die der MASC-ST übernehmen muss. Dies rechtfertigt eine genauere Betrachtung.

4.2.1 SNEP

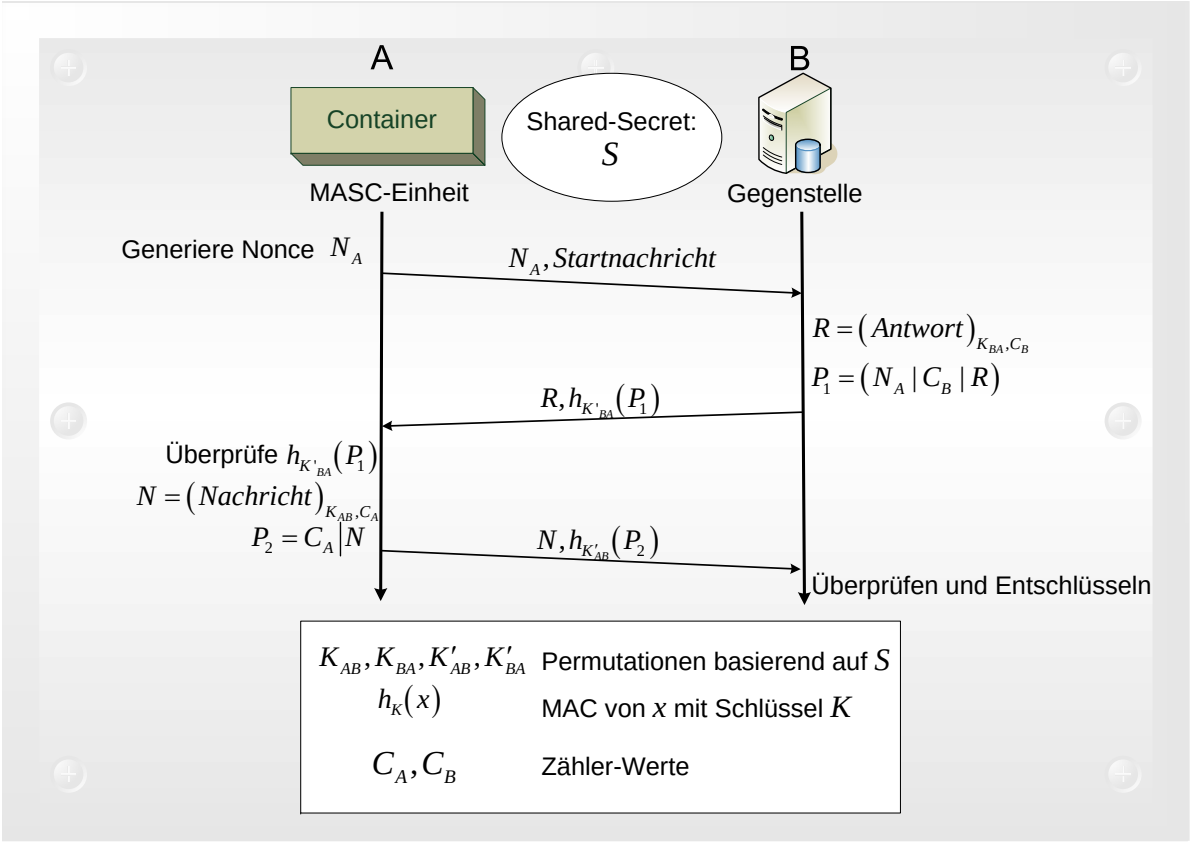


Abbildung 4.1: SNEP (Baustein von SPINS)[68]

Bei SNEP gibt es einen zentralen Server, der mit jedem Sensorknoten das gemeinsame Geheimnis (engl. *shared secret*) S teilt. Für das Meshing und andere Statusmeldungen ist es wichtig, dass die Knoten auch untereinander kommunizieren können, weswegen ein authentifiziertes Broadcast-Sendeverfahren benötigt wird. Die gesamte Architektur nennt sich SPINS (Security Protocols for Sensor Networks) [68]. Sie besteht aus zwei Teilblöcken: Der eine sorgt für die verschlüsselte Kommunikation mit dem Server und nennt sich SNEP (Secure Network Encryption Protocol), der andere ermöglicht den authentifizierten Broadcast, ist eine vereinfachte Variante von TESLA [67] und nennt sich μTESLA. Für Meshing-Zwecke wäre dieser zweite Teil interessant, für den Aufbau des Tunnels jedoch nur der erste Teil. Abbildung 4.1 skizziert dieses Verfahren. Hinzu kommt, dass bei SNEP der Server und die Basisstation die gleiche Partei sind. Dies ist bei MASC nicht der Fall – Broadcasting kann also gar nicht von dem Server mit den geheimen Schlüsseln durchgeführt werden.

SNEP benutzt für den Datentransport von S abgeleitete Schlüssel. Die Schlüssel sind aber nicht temporär, sondern allgemein gültig. Für einen besseren Schutz werden für jede Richtung verschiedene

Schlüssel verwendet. Es gibt daher vier Schlüssel. K_{AB} und K_{BA} werden für die Nutzdatenverschlüsselung benötigt, K'_{AB} und K'_{BA} sind die jeweiligen richtungsgebundenen Schlüssel für den MAC (Message Authentication Code, Erklärung in Anhang A.2). Der Nonce dient lediglich dem Nachweis der Aktualität der Nachricht (engl. „data freshness“). Angewendet auf MASC wäre die Containereinheit A und die Gegenstelle B. A sendet einen Nonce an B, wobei die folgende Antwort von B denselben Nonce wieder enthalten muss. Auf diese Weise werden alte wieder eingespielte Nachrichten erkannt (Schutz vor Replay-Attacken).

Bei SNEP werden keine **Sitzungsschlüssel** erzeugt, sondern die erzeugten Schlüssel für jede Verbindung verwendet. Jedoch gibt es nach [60, 12.2.2] vier Gründe für das Verwenden von Sitzungsschlüsseln: Die Beschränkung der verfügbaren verschlüsselten Nachrichten zur Erschwerung eines kryptanalytischen Angriffs, die Minimierung der Auswirkungen eines aufgedeckten Schlüssels, das Vermeiden des unnötigen Speicherns vieler Langzeitschlüssel bei gleichzeitiger Rückführung auf ein geteiltes Geheimnis und um eine Unabhängigkeit zwischen den Kommunikationssitzungen zu erzeugen. Dass SNEP keine Sitzungsschlüssel verwendet, disqualifiziert das Verfahren bei der Wahl der Tunnelgenerierung für die Containeranwendung, bei der die Lebensdauer der Einheiten einige Jahre beträgt und sich daher die Auswirkungen aufgedeckter (Sitzungs-)Schlüssel extrem unterscheiden.

4.2.2 AKEP2

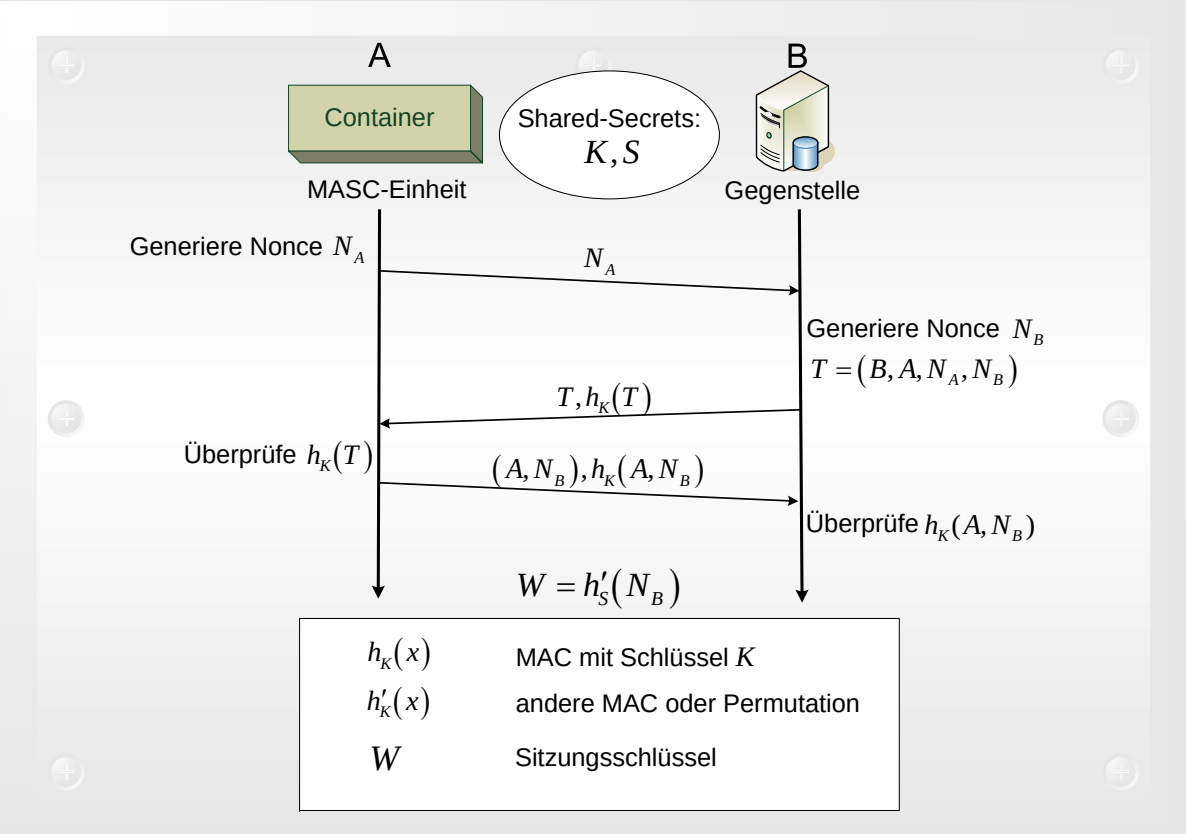


Abbildung 4.2: Authenticated Key Exchange Protocol 2 (AKEP2)

Als Alternative zu den speziellen Sensornetzwerk-Protokollen wird mit AKEP2 ein Verfahren für ein dynamisches und authentifiziertes Schlüsseletablierungsprotokoll gemäß den Definitionen aus Abschnitt A.6 behandelt. Das Handbook of Applied Cryptography [60] stellt mit AKEP2 (abgeleitet aus [7]) ein dynamisches und authentifiziertes Schlüsseletablierungsprotokoll vor. Die schematische Funktionsweise ist in Abbildung 4.2 zu sehen. Das Verfahren generiert einen temporären Sitzungsschlüssel W. Nach zwei Nachrichten ist der Sitzungsschlüssel bereits generiert – die dritte Nachricht dient der Authentifizierung des Senders (Container). Das Verfahren basiert auf zwei geteilten Geheimnissen, die beiden Kommunikationspartnern initial bekannt sind. Das eine entspricht dem Schlüssel für die MAC-Funktionen und wird K bezeichnet. Das andere (S) dient lediglich der Berechnung des temporären Sitzungsschlüssels. Auf diese Weise wird sichergestellt, dass ein Aufdecken des Schlüssels für die MAC-Funktionen nicht zur Folge hat, dass der Schlüssel für die Verschlüsselung aufgedeckt wird. Laut [60, Definition 12.20] kann der Sitzungsschlüssel dabei entweder über eine andere MAC-Funktion oder über eine pseudozufällige Permutation der Eingabewerte berechnet werden.

AKEP2 ist ein Protokoll, welches für zwei gleichberechtigte Partner entworfen wurde. Die Containeranwendung ist hingegen asymmetrisch. Ein potenter Server, der sehr viel Strom verbrauchen kann und viel Prozessorleistung besitzt, kommuniziert mit Containereinheiten, die mit einem Prozessor bestückt sind, der eine sehr geringe Leistungsaufnahme hat und dementsprechend wenig leistungsfähig ist. Für diesen speziellen Anwendungsfall bietet AKEP2 noch Verbesserungspotenzial. Die letzte Nachricht, die der Containerauthentifizierung dient, kann weggelassen und implizit durch die erste Nutzdatennachricht ersetzt werden. Dies entlastet den Energieverbrauch, weil auf diese Weise eine ganze Nachricht nicht gesendet werden muss und die Sendeeinheit bekanntlich der größte Energieverbraucher ist.

4.3 Protokollentwurf

Das MASC-SecureTunnel-Protokoll (MASC-ST) ist eine Abwandlung des AKEP2-Protokolls mit einigen Optimierungen für den Containertransport. Dabei gibt es drei wesentliche Änderungen: Die dritte, der Containerauthentifizierung dienende Nachricht wird aus Gründen der Energieeffizienz ausgelassen. Stattdessen geschieht die Authentifizierung implizit mit der ersten verschlüsselten Nutznachricht (M). MASC-ST verwendet obendrein zwei verschiedene Sitzungsschlüssel: Einen für die schnellen MAC-Berechnungen zur Authentifizierung und Nachrichtenintegrität, einen zur Verschlüsselung. Dieses Verfahren hat sich als „Best-Practice“ durchgesetzt, da auf diese Weise Angriffspunkte, wie in [60, 9.88ff] dargestellt, ausgeschlossen werden können. Der dritte Unterschied ist die Verwendung nur eines geteilten Geheimnisses, aus dem beide Schlüssel (vergleichbar mit SNEP) berechnet werden.

4.3.1 Schlüsselverwaltung

Basis der Schlüsselverwaltung ist das geteilte Geheimnis (*Shared-Secret*) S, welches ausschließlich die beiden Kommunikationspartner Containereinheit und zugeordnete Gegenstelle kennen. S wird von der

RSO bei der Abnahmeprüfung des Containers in die Containereinheit gebracht und gleichzeitig in der Gegenstelle gespeichert.

Aus in Abschnitt 4.2.1 genannten Gründen sollten alle übertragenen Nachrichten mit Sitzungsschlüsseln verschlüsselt übertragen werden. Das Geheimnis S basiert auf einer RSO-innerbetrieblichen Vertrauensrelation, die durch Schlüsselableitung auf alle Sitzungsschlüssel übertragen werden kann. Allerdings sind die Ereignisse bereits verschlüsselt im Speicher der Einheit abgelegt. Diese zu entschlüsseln und neu zu verschlüsseln, nur damit ein Sitzungsschlüssel verwendet werden kann, erscheint wenig sinnvoll. Daher wird beim MASC-ST neben der normalen kurzen Sitzung, die aufgebaut wird, um ein Ereignisupdate zu übertragen, eine logische längere Sitzung eingeführt, für die der Nutznachrichtenschlüssel definiert wird.

Bei dem MASC-ST wird zwischen der **Kurzsitzung** und der **Langsitzung** unterschieden. Die Kurzsitzung startet bei jedem neuen Verbindungsaufbauversuch des Containers mit dem Handshake. Dabei wird jedes Mal ein neuer Kurzsitzungsschlüssel K_M berechnet, der bereits im aktuellen Handshake für die MAC-Berechnung verwendet wird. Bei dem Langsitzungsschlüssel handelt es sich um den Verschlüsselungsschlüssel K_E. Dieser wird nur auf Kommando der Gegenstelle verändert und gilt erst für alle später eintretenden Ereignisse. Die MASC-Einheit verwendet für das verschlüsselte Ablegen der Ereignisse im Nicht-Flüchtigen-Speicher den bei Ereignisgenerierung gültigen Langsitzungsschlüssel K_E. Wenn die Gegenstelle den Wechsel des Langsitzungsschlüssels anordnet, was sie aus Sicherheitsgründen nach einigen Kurzsitzungen tun sollte, gilt dies nur für die zukünftig auftretenden Ereignisse. Vergangene Ereignisse werden nicht neu verschlüsselt.

Beispiel 4.1 (Ereignisse)**:**

Die MASC-Gegenstelle sendet zu Beginn eines neuen Containertransports ein *Reset* an die MASC-Einheit, die daraufhin den Ereigniszähler auf Null setzt[1]. Das *Reset* impliziert auch die Verwendung eines neuen Langsitzungsschlüssels. Der Container generiert folgend die Ereignisse 1–10. Wenn der Container **online** ist, aktualisiert er die Ereignisse fortlaufend an der Gegenstelle. Mit online ist dabei gemeint, dass eine MASC-Infrastruktur vorhanden und die MASC-Einheit mit dem Internet verbunden ist. Liegt die Verbindung nicht vor, ist die Einheit sinngemäß **offline**. Nachdem die MASC-Einheit Ereignis 10 übertragen hat, entscheidet die Gegenstelle, den Langsitzungsschlüssel zu ändern – ein entsprechender Befehl fordert die MASC-Einheit dazu auf. Diese ändert daraufhin den Langsitzungsschlüssel. Der neue Schlüssel wird für die zukünftig auftretenden Ereignisse verwendet, beginnend mit Nummer 11. Es ist allerdings möglich, dass die Gegenstelle die Ereignisse 1–9 zu diesem Zeitpunkt noch nicht erhalten hat, weil die Einheit vorher nicht online war. Fordert sie folgend zur Übermittlung dieser Nachrichten auf, erhält sie die mit dem alten Langsitzungsschlüssel verschlüsselten Nachrichten 1–9. Da die Gegenstelle alle Langsitzungsschlüssel speichert und nicht wie die MASC-Einheit nur den aktuellen, kann die Gegenstelle auch die Nachrichten entschlüsseln,

[1] Dieser Schritt ist optional, da die MASC-Einheit vom Zeitstempel der Ereignisse auf deren Auftreten zurückschließen kann.

die noch mit den alten Schlüsseln generiert wurden.

4.3.2 Handshake

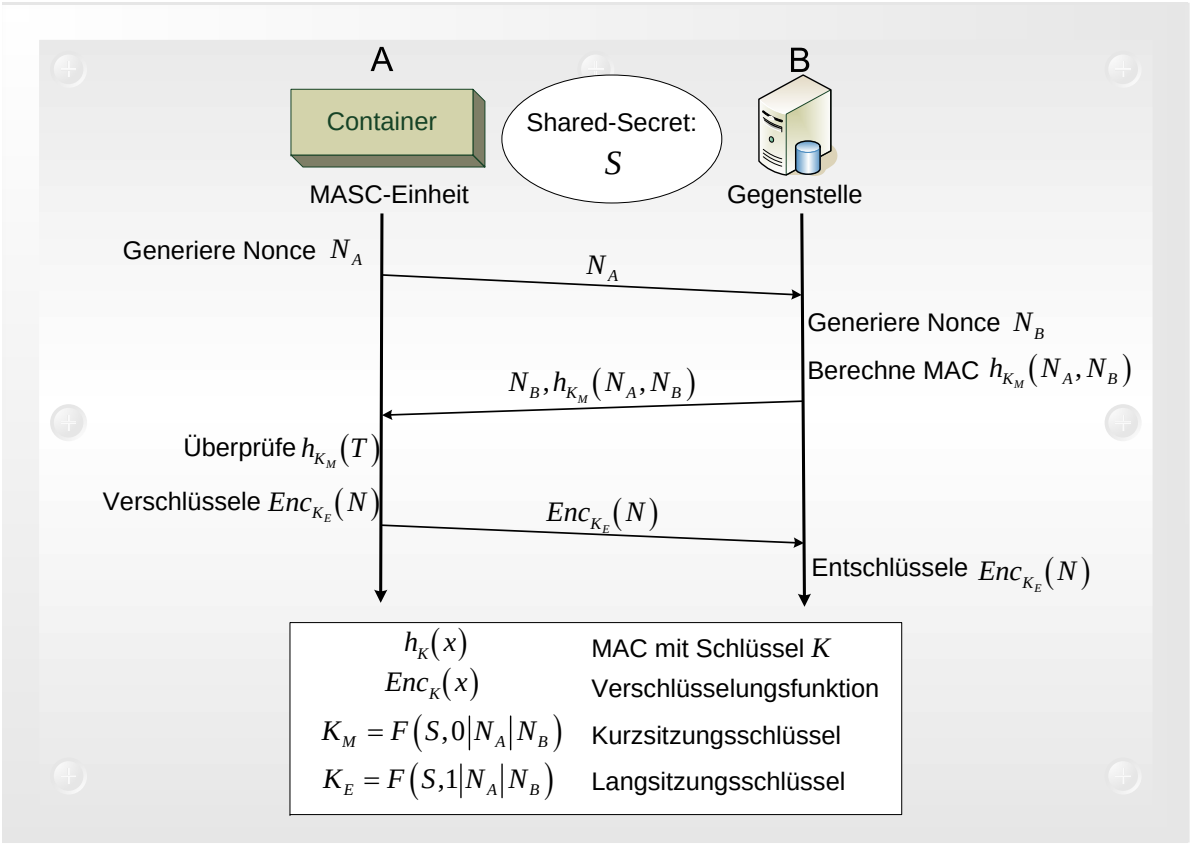


Abbildung 4.3: MASC-SecureTunnel, Tunnelgenerierung

Jede Kurzsitzung beginnt damit, dass die MASC-Einheit A einen Nonce (Einmal-Zufallszahl) an die Gegenstelle B sendet.

$$A \rightarrow B: \; N_A \tag{4.1}$$

Bei AKEP2 enthielt diese Nachricht noch die Absender-ID, die aber, wie bereits in 3.3.2 angesprochen, im Protokollrahmen jeder Nachricht übertragen wird. Wenn diese Nachricht von der Gegenstelle empfangen wird, ist die MASC-Einheit online, was sie aber selber noch nicht weiß. Die Gegenstelle generiert ein eigenes Nonce N_B und ist somit im Besitz beider Nonces. Die Gegenstelle berechnet den Kurzsitzungsschlüssel wie folgt, wobei das | für die Konkatenation steht.

$$K_M = F(S, 0|N_A|N_B)$$

Bei der Funktion F handelt es sich um eine pseudozufällige Einwegfunktion. Es bietet sich im MASC-Szenario an, die MAC-Funktion (h_K – MAC-Funktion mit Schlüssel K) für die Berechnung dieser Einwegfunktionen heranzuziehen, da diese bereits implementiert ist. In diesem Fall wäre S der Schlüsselparameter und $0|N_A|N_B$ der Eingabewert der MAC-Funktion: $F(S,0|N_A|N_B) \equiv h_S(0|N_A|N_B)$.

Die Gegenstelle berechnet den MAC mit dem Kurzsitzungsschlüssel und den beiden Nonces als Eingabewert und sendet diesen zur MASC-Einheit. Da diese aber N_B noch nicht kennt, wird N_B im Klartext zugefügt.

$$B \rightarrow A:\ N_B, h_{K_M}(N_A, N_B) \tag{4.2}$$

Die MASC-Einheit erhält N_B und berechnet ebenfalls den Kurzsitzungsschlüssel K_M. Anschließend berechnet sie den MAC beider Nonces mit Hilfe des Schlüssels. Stimmt der errechnete Wert mit dem empfangenen überein, weiß die MASC-Einheit, dass die Verbindung mit der korrekten Gegenstelle aufgebaut wurde: Die MASC-Einheit ist online und die Gegenstelle authentifiziert.

4.3.3 Datenübertragung

Angenommen, MASC-Einheit und Gegenstelle haben sich in einer vorherigen Kurzsitzung auf einen Langsitzungsschlüssel K_E geeinigt. Mit diesem Schlüssel hat die MASC-Einheit alle folgend aufgetretenen Ereignisse im Speicher abgelegt. Nachdem die ersten beiden Nachrichten des Handshakes erfolgreich abgelaufen sind, überträgt die Containereinheit das jüngste Ereignis (M) an die Gegenstelle.

$$A \rightarrow B:\ Enc_{K_E}(M), h_{K_M}(Enc_{K_E}(M)) \tag{4.3}$$

$Enc_{K_E}(M)$ ist das verschlüsselte Ereignis inklusive Ereignis-ID, welches direkt (bereits verschlüsselt) aus dem Ereignisspeicher gelesen werden kann. Gleichzeitig wird ein MAC-Wert dieser Nachricht angehängt. Die Gegenstelle überprüft zunächst den MAC; ist dieser korrekt, ist erstens die MASC-Gegenstelle authentifiziert und zweitens die Integrität von M gewahrt. Bei Erfolg entschlüsselt die Gegenstelle M und erhält die Ereignis-ID und das Ereignis selbst. Bei nicht korrektem MAC wird die Verarbeitung abgebrochen und die Nachricht als nicht gesendet betrachtet.

Die Gegenstelle kann jetzt verschlüsselte Nachrichten an die MASC-Einheit senden, um zum Beispiel fehlende Ereignisse anzufordern oder eine neue Langsitzung zu beginnen. In diesem Fall würde eine Aufforderung dazu an die MASC-Einheit mit dem alten Schlüssel verschlüsselt übertragen werden. Die Berechnung des neuen Langsitzungsschlüssels läuft auf beiden Seiten wie folgt ab.

$$K_E = F(S, 1|N_B|N_A)$$

Die MASC-Einheit löscht daraufhin den alten Langsitzungsschlüssel und verwendet für alle zukünftigen Ereignisse nur noch den neuen. Allerdings kann die Gegenstelle nicht sicher sein, ob dieses Paket angekommen ist. Sie geht im Folgenden davon aus, dass alle Ereignisse mit einer ID größer als das zuletzt erhaltene mit dem neuen Schlüssel verschlüsselt werden.

Allerdings können jetzt zwei abweichende Fälle auftreten: Erstens könnte die Containereinheit die Nachrichten nicht empfangen haben und weiterhin alle Pakete mit dem alten Schlüssel verschlüsseln. Zweitens könnten zwischen dem Handshake und der Einigung auf einen neuen Sitzungsschlüssel weitere Ereignisse aufgetreten sein, die noch mit dem alten Schlüssel verschlüsselt wurden. Der neue Schlüssel wird erst für die zukünftigen Ereignisse verwendet.

Die Gegenstelle weiß also nicht, ob die (für sie) nachfolgenden Ereignisse bereits mit dem neuen Schlüssel verschlüsselt wurden. Sie probiert diesen aber zunächst aus. Kommt bei der Entschlüsselung keine sinnvolle Nachricht heraus, die mit einer Ereignis-ID beginnt und nachfolgende Ereignisse korrekt auflistet, wird der nächstältere Langsitzungsschlüssel für die Entschlüsselung herangezogen. Nach der erfolgreichen Entschlüsselung fügt die Gegenstelle zu jedem Schlüssel das erneuerte Ereignisintervall in der Datenbank ein, damit sie weiß, welche Schlüssel für welche Ereignisse verwendet wurden. Wird jetzt ein neues Ereignis übermittelt oder ein altes Ereignis auf Anforderung nachgetragen, kann die Gegenstelle herausfinden, ob der Schlüssel für die Nachricht bekannt ist oder mehr als ein Schlüssel für die Entschlüsselung in Betracht kommt. Der Regelfall wird sein, dass der Schlüssel bekannt ist. Nur für die zwei obengenannten Ausnahmefälle ist diese Sonderbearbeitung notwendig.

Vor dem Einbau der MASC-Einheit kennt diese bereits das geteilte Geheimnis. Unmittelbar nach dem Einbau ist die Einheit im Initialisierungsmodus – es werden noch keine Ereignisse generiert. Erst während des ersten erfolgreichen Handshakes mit der RSO werden die beiden Sitzungsschlüssel angelegt. Bei diesem Handshake wird keine Nutznachricht übertragen, sondern ein Initialisierungsmuster, was der Gegenstelle anzeigt, dass die Initialisierung abgeschlossen ist. Jetzt befinden sich Einheit und Gegenstelle im Betriebsmodus und können im Folgenden das MASC-ST wie beschrieben ausführen.

Das Initialisierungsmuster ist die einzige Nachricht, die bestätigt werden muss. Kommt die Bestätigung nicht an, versucht die Einheit die erneute Übertragung des Musters. Wenn die Gegenstelle ein Initialisierungsmuster zugesendet bekommt, weiß sie, dass die vorherige Bestätigung nicht ankam, und wiederholt die Bestätigung. Erst wenn die Bestätigung ankommt, wechselt die MASC-Einheit vom Initialisierungs- in den Betriebsmodus. Um Probleme bei der Initialisierung zu vermeiden, lohnt es sich, den Einbau der Einheit und damit die folgende Initialisierung in direkter Nähe einer MASC-Basisstation durchzuführen, um so die Übertragungsfehlerwahrscheinlichkeit zu minimieren.

4.3.4 Protokollbetrachtung

Angreifer könnten beliebige Übertragungen abhören oder blockieren. Deswegen wurde das MASC-ST so entworfen, dass es zu keiner Zeit von einer verlässlichen Datenübertragung abhängt. Hinzu kommt,

dass Datenübertragungsfehler zwischen Gateway und Container nicht unwahrscheinlich sind. Das Protokoll ist daher so ausgelegt, dass jede Übermittlung fehlerhaft sein kann, ohne dass ein Schaden entsteht. Sollte die erste Nachricht (4.1) geblockt werden, entspricht dies einem Nicht-Online-Sein. Erst wenn das Nonce mit dem MAC der Gegenstelle bei der MASC-Einheit ankommt und diese den MAC verifizieren konnte, betrachtet die MASC-Einheit sich als online. Andernfalls geht sie davon aus, nicht online zu sein, und sendet in größer werdenden Abständen Handshake-Initialisierungs-Nonces aus, um zu überprüfen, ob sich der Status geändert hat.

Zwischenzeitlich ist die Gegenstelle zwar an der Containereinheit authentifiziert, umgekehrt gilt dies jedoch nicht. Erst wenn die verschlüsselte Nutznachricht (4.3) bei der Gegenstelle ankommt und der zugehörige MAC verifiziert wurde, ist dies der Fall. Es kann also passieren, dass die MASC-Einheit ein Ereignis an die Gegenstelle sendet, dieses dort aber nicht ankommt. Aus der Sicht der Gegenstelle ist die MASC-Einheit dann noch immer offline und das jüngste Ereignis wurde nicht übertragen. Beim nächsten Handshake kann dieses allerdings von der Gegenstelle angefordert werden.

Das MASC-ST verwendet keine Bestätigungspakete, da diese nur unnötigen Energieaufwand erzeugen und die Situation nicht verbessern. Das Nicht-Erhalten eines Bestätigungspaketes würde nämlich keineswegs sicherstellen, dass ihr Ereignis nicht bei der Gegenstelle ankam. Die Einheit könnte in diesem Fall nicht unterscheiden, ob ihre Nachricht bei der Gegenstelle nicht angekommen oder die Antwort der Gegenstelle einem Übertragungsfehler zum Opfer gefallen wäre. Träte dies ein, könnte die Containereinheit entweder mittlerweile offline sein (äquivalent zu der Situation, dass ein Angreifer die Nachrichten blockierte) oder es wäre zu einem temporären Übertragungsfehler gekommen, der durch erneute Sendung wahrscheinlich überwunden werden könnte. Nähme sie den Energieaufwand einer Neu-Sendung in Kauf und übertrüge das Ereignis nochmals? Wie oft wiederholte sie die Neu-Sendung? Durch das Verzichten auf Bestätigungspakete betrachtet die Gegenstelle die Einheit (bis zum nächsten erfolgreichen Handshake) als offline.

Die Einheit fängt dabei nicht unmittelbar nach jedem aufgetretenen Ereignis mit dem neuen Verbindungsaufbau an. Da viele Ereignisse nicht isoliert auftreten[2], wartet sie nach Sensorschwankungen zunächst ein definiertes Zeitintervall ab, um nicht mit der aufwändigen Übertragung zu beginnen, während noch weitere Verschlüsselungen anstehen, falls parallel neue Ereignisse auftreten. Des Weiteren muss bei neuen Ereignissen nicht jedes Mal eine neue Kurzsitzung aufgebaut werden. Sollte die alte Sitzung noch aktiv sein, kann diese weiter benutzt werden. Die Inaktivität einer Sitzung tritt erst nach einem festgelegten Zeitintervall ein, welches auf einer Abstimmung aus Energiesparen und Sicherheit beruht. Viele kurze Sitzungen sind aus Sicherheitsgründen besser, erfordern aber jedes Mal den Handshake und die Schlüsselberechnung, was Energie verbraucht. Für den Fall, dass über einen langen Zeitraum kein Ereignis auftritt, generiert die Containereinheit nach einem bestimmten Zeitintervall ein Statusereignis, welches einen neuen Verbindungsversuch anstößt.

[2] Das Fallen eines Containers wirkt sich auf mehrere Bewegungssensoren aus und erzeugt mehrere Schwankungen. Wird die Fracht in den Container geladen, treten ebenfalls viele Sensorschwankungen zusammen auf.

Die Alternative zum Verwenden der kommandobasierten Änderung des Langsitzungsschlüssels ist ein automatisiertes Verfahren, dass bei jedem Handshake sowohl Kurz- als auch Langsitzungsschlüssel geändert werden. Da die auftretenden Ereignisse nach wie vor mit dem aktuellen Langsitzungsschlüssel verschlüsselt würden, ändert dieses Verfahren nichts an dem Problem der potenziellen Schlüsselasynchronität. Bevor die Gegenstelle weiß, ob sie wirklich mit der korrekten Containereinheit verbunden ist (bevor Nachricht (4.3) bei der Gegenstelle ankommt), könnten bereits eine Vielzahl von Schlüsseln erzeugt worden sein. Diese müssten alle auf Verdacht von der Gegenstelle gespeichert werden, da sie nicht weiß, welche dieser Schlüssel von der MASC-Einheit übernommen worden sind. Dies wäre zudem ein neuer Angriffspunkt. Der Angreifer gibt sich als Container x aus und sendet andauernd Nonces. Die Gegenstelle müsste die Tatsache berücksichtigen, dass jeder Versuch von der korrekten MASC-Einheit stamme, und den Handshake jeder Anfrage beantworten. Es müssten bei jedem Versuch beide Schlüssel berechnet werden. Die Langsitzungsschlüssel würden dabei alle langfristig gespeichert, weil es sein könnte, dass genau dieser bei der MASC-Einheit angekommen wäre. Jeder Schlüssel müsste beim Empfang der Nachricht (4.3) durchprobiert werden. Wenn der Angreifer sich dann auch noch als verschiedene Container ausgibt und den Angriff parallelisiert, hat die Datenbank nach kurzer Zeit keinen Speicherplatz mehr zur Verfügung. Daher sollte nur ein neuer Langsitzungsschlüssel berechnet werden, wenn beide Seiten korrekt authentifiziert wurden.

4.3.5 Protokollverifikation

Die Sorge beim Entwurf neuer Protokolle sind Sicherheitslücken, die sich wegen eines falschen Designs ausnutzen lassen. Die meisten Sicherheitsprotokolle haben das Vertrauen ihrer Benutzergemeinde der Tatsache zu verdanken, dass sie seit vielen Jahren existieren und kein effizientes Verfahren zum ungewollten Ausnutzen von Sicherheitslücken bekannt geworden ist. Ein alternativer Ansatz zur Vertrauenserhöhung ist eine auf formalen Methoden basierende Analyse des Protokolls. Speziell gibt es Modellprüfungsverfahren, die Protokollmodelle auf die Resistenz gegenüber einer Vielzahl bekannter Angriffe hin untersuchen können. Bezüglich des erstellten Modells und der getroffenen Annahmen des Modellprüfers können die Verfahren durch Verwenden von Logik beweisen, dass die getesteten Angriffe nicht auf das Modell anwendbar sind.

Ein verbreitetes Modellprüfungswerkzeug ist **ProVerif** von Bruno Blanchet [9]. ProVerif dient der Protokollverifikation und kennt bereits kryptographische Primitive. Während des Modellierungsverfahrens werden die Primitive dem System über eine Textdatei bekannt gemacht. Dieses Verfahren ist vergleichbar mit Quellcodedateien bei Programmiersprachen, in denen Funktionen mit speziellen Parametern verwendet werden. Die Funktionen entsprechen bei ProVerif der Verwendung der Primitive und Kommunikationskanäle. Es wird dabei angenommen, dass die verwendeten Primitive perfekt sind. Eine Hashfunktion wird beispielsweise als eine kollisionsresistente pseudozufällige Einwegfunktion angenommen. Welche (gegebenenfalls schwache Implementierung) das Verfahren in der Praxis repräsentiert, wird nicht berücksichtigt.

ProVerif kann vier Sicherheitseigenschaften eines Protokollmodells nachweisen: Die Geheimhaltung der verschlüsselten Nachricht im Rahmen des Protokolls, die korrekte Authentifizierung der Schlüssel(besitzer), die starke Geheimhaltung[3] einer Nachricht und das Vorhandensein einer Äquivalenzrelation zwischen zwei Prozessen gemäß [10].

Zur besseren Lesbarkeit wurde die Darstellung des ProVerif-Quelltextes zweigeteilt. Die Definitionen und Systemanfragen des ersten Quellcodeteils werden in Listing 4.2 behandelt. Die Modellierung des Protokolls basiert auf dem zweiten Teil des Codes, welches in Listing 4.1 dargestellt ist.

Protokollmodellierung

Listing 4.1: ProVerif Quellcode 2

```
1  (*Protocol*)
2  let processCon =                          (* Modellierte Containereinheit *)
3    new na; (* Erstellung Nonce A *)
4    out (e,na);                               (* Senden Nachricht 1 *)
5
6    in (e,msg);                               (* Empfang Nachricht 2 *)
7    let x = fst(msg) in (* Extrahierung Nonce B *)
8    let y = snd(msg) in (* Extrahierung MAC *)
9    let km = f(S,(c0,na,x)) in (* Schlüsselberechnung KM *)
10   let t1 = concat(x,na) in (* Konkatenierung Nonces A und B *)
11   let hash1 = keyhash(t1,km) in (* Berechnung MAC *)
12   if hash1 = y then (* Überprüfung MAC *)
13   (
14     event endCon(hash1);
15     let ke = f(S,(c1,x,na)) in (* Schlüsselberechnung KE *)
16     let enc = sencrypt(m,ke) in (* Verschlüsselung Nachricht *)
17     new counter; (* Nachrichtidentifizierung *)
18     let henc = keyhash((enc,counter),km) in (* MAC Nachricht *)
19     event beginSvr(henc);
20     out (e,(enc,counter,henc))              (* Senden Nachricht 3 *)
21 ).
22
23 let processServer =                        (* Modellierte Gegenstelle *)
24   in (e,na);                                (* Empfang Nachricht 1 *)
25   new nb; (* Erstellung Nonce B *)
26   let km = f(S,(c0,na,nb)) in (* Schlüsselberechnung KM *)
27   let t2 = concat(nb,na) in (* Konkatenierung Nonces A und B *)
28   let hash2 = keyhash(t2,km) in (* Berechnung MAC *)
29   let msg = concat (nb,hash2) in (* Zusammenstellung Sendung *)
30   event beginCon(hash2);
```

[3] Mit starker Geheimhaltung ist gemeint, dass Dritte bei Betrachtung der verschlüsselten Übertragung nicht in der Lage sind zu sehen, ob sich die Nachricht verändert hat.

```
31   out (e,msg);                                    (* Senden Nachricht 2 *)
32
33   in (e,(msg2,msg3,msg4));                        (* Empfang Nachricht 3 *)
34   let ke = f(S,(c1,nb,na)) in (* Schlüsselberechnung KE *)
35   let hash3 = keyhash((msg2,msg3),km) in  (* Extrahierung MAC *)
36   if msg4 = hash3 then (* Überprüfung MAC *)
37   (
38     event endSvr(hash3);
39     let m2 = sdecrypt(msg2,ke) in (* Entschlüsselung *)
40     let ack = keyhash(msg3,km) in (* Bestätigungspaket erstellen *)
41     out (e, ack)                                   (* Senden Bestätigung *)
42 ).
43
44
45 process                                        (* Starten beider Prozessmodelle *)
46         new S;
47 ((!processCon) | (!processServer))
```

Listing 4.1 zeigt die formale Darstellung des Protokollentwurfs. Das Modell besteht aus zwei Prozessen, die Containereinheit und Gegenstelle entsprechen. Das Modell der Einheit wird `processCon`, das der Gegenstelle `processServer` bezeichnet. Die beiden Prozesse bilden das Verhalten der jeweiligen Kommunikationspartner während eines Protokollablaufes ab.

Containereinheit Zwischen Zeile 51 und 70 wird der Prozess der Containereinheit beschrieben. Zeile 52 beschreibt die Nonce-Erstellung und in 53 wird das Nonce (per `out`) an den unsicheren Kanal `e` gegeben – dies entspricht der ersten Nachricht des MASC-ST. In Zeile 55 erhält das Nonce (per `in`) die mutmaßliche Antwort der Gegenstelle und extrahiert in 56 und 57 die Nachricht. Zeile 58 bis 60 beschreibt das Überprüfen des Gegenstellen-MAC und in Zeile 61 geschieht die Abfrage, ob die Verifizierung erfolgreich ist. Nur im Erfolgsfall werden die Zeilen 63 bis 69 durchgeführt. In den Zeilen 64 bis 67 wird die dritte Nachricht des MASC-ST erstellt. Aus Gründen der Modellierbarkeit wird nicht berücksichtigt, dass die Verschlüsselung der Nachricht bereits vorher erfolgte (dies bedeutet für das Protokoll keinen Unterschied. Ebenfalls wird in Zeile 66 die ID der Nachricht extrahiert, um die implizite Ereignis-ID darzustellen. In Zeile 69 wird die letzte Nachricht des MASC-ST übertragen.

Gegenstelle Der Gegenstellenprozess besteht aus drei Blöcken. Der erste Block beginnt in Zeile 73 mit dem Empfang der Container-Nonces und endet in Zeile 80 mit dem Senden der Antwort. In ihm wird der eigene Nonce generiert, der MAC-Schlüssel berechnet, der MAC über die Nonces berechnet sowie Nonce und MAC zur Übertragung zusammengefasst. Der zweite Block beginnt in Zeile 82 mit dem Empfang der dritten Nachricht des MASC-ST und beschreibt das Extrahieren der Nachricht und das Überprüfen des MAC. In Zeile 85 erfolgt die Abfrage seiner Korrektheit. Der dritte Block wird nur

ausgeführt, wenn diese Abfrage positiv beantwortet wurde. In Zeile 88 wird die Nachricht entschlüsselt, in Zeile 89 ein Bestätigungspaket erstellt und in Zeile 90 zurückgesendet. Dieser letzte Schritt ist optional und beschreibt die Variante, dass Bestätigungspakete (ack) gesendet werden sollen. Wenn die Pakete nicht bestätigt werden sollen, würde Zeile 89 entfallen. Da diese Nachricht vom Containerprozess auch nicht mehr angenommen wird, hat diese Zeile keinen Einfluss auf die Ergebnisse von ProVerif.

In den Zeilen 94 bis 96 werden die beiden Prozesse aufgehängt und in Zeile 96 wird ProVerif deutlich gemacht, dass die beiden Prozesse miteinander kommunizieren und so das Protokoll definieren.

Sicherheitseigenschaften

Listing 4.2: ProVerif Quellcode 1

```
1 (*param traceDisplay = long.*)
2
3 (* Public key cryptography *)
4 fun pk/1.
5 fun encrypt/2.
6 reduc decrypt(encrypt(x,pk(y)),y) = x.
7
8 (* Keyed hash function *)
9 fun keyhash/2.
10
11 (* Shared-key cryptography *)
12 fun sencrypt/2.
13 reduc sdecrypt(sencrypt(x,y),y) = x.
14
15 (* F functions *)
16 fun f/2.
17
18 (* T functions *)
19 fun t/2.
20
21 (* concatenation *)
22 fun concat/2.
23
24 (* fst returns the first atomic component of a concatenation *)
25 reduc fst(concat(x,y)) = x.
26
27 (* snd returns a concatenation without its first atomic component *)
28 reduc snd(concat(x,y)) = y.
29
30 (* Declarations of free names *)
31 private free m. (* Nachricht m *)
```

```
32 free e. (* Öffentlicher Kanal e *)
33
34 (* Constants *)
35 data c0/0. (* 0 für Kurzsitzungsschlüssel *)
36 data c1/0. (* 1 für Langsitzungsschlüssel *)
37
38 (* Secrecy assumptions *)
39 not attacker:S. (* Shared secret *)
40
41 (* The query, for protocol analysis *)
42 query ev:endCon(x) ==> ev:beginCon(x). (* Authentifizierung Server *)
43 query ev:endSvr(x) ==> ev:beginSvr(x). (* Authentifizierung Container *)
44 (* Injective queries to verify the absence of replay attacks *)
45 query evinj:endCon(x) ==> evinj:beginCon(x).
46 query evinj:endSvr(x) ==> evinj:beginSvr(x).
47 query ev:endCon(x). (* Erreichbarkeitstest Protokollende *)
48 query ev:endSvr(x).
49 query attacker:m. (* Geheimhaltung Nachricht m *)
```

Das Listing 4.2 beschreibt die Definitionen (Zeilen 1 bis 40) und die zu prüfenden Anfragen (Zeilen 41 bis 49). Die Definitionen deklarieren Namen und weisen ihnen die Primitive ProVerifs zu. Der Wert hinter dem Schrägstrich beschreibt die Anzahl der Eingabeparameter des jeweiligen Primitivs. In Zeile 31 wird die Nachricht als geheim (private) deklariert und der Kanal `e` als offen zugänglich (`free`). In den Zeilen 35 und 36 werden die als 0 und 1 definierten Werte zur Berechnung der Sitzungsschlüssel als beliebige (aber feste) Konstanten definiert (eine feste Wertzuweisung ist mit ProVerif nicht möglich). In Zeile 39 wird die Sicherheitsannahme formuliert, dass ein Angreifer nicht das geteilte Geheimnis kennt.

Neben Prozessen und Primitiven kennt ProVerif ebenfalls Ereignisse. Diese werden verwendet, um Authentifizierungsprozesse darzustellen. In den Modellen von Containereinheit und Gegenstelle wurden daher Ereignisse definiert, die von ProVerif ausgewertet werden, um das Ausbleiben von Angriffen bezüglich des Modells zu beweisen.

Authentifizierung In Zeile 79 (Listing 4.1) wird vor dem Senden der Antwort der Gegenstelle das Ereignis `BeginCon(hash2)` generiert. Die Variable `hash2` ist dabei der vorher berechnete MAC-Wert. Die Antwort wird anschließend an die Containereinheit gesendet und dort extrahiert (55 bis 57). Wenn der MAC verifiziert wurde, ist die Gegenstelle authentifiziert und das Ereignis `EndCon(hash1)` wird erzeugt. Die Variable `hash1` entspricht dabei dem durch die Containereinheit berechnet MAC-Wert. Wenn kein Angriff und kein Übertragungsfehler vorliegen, sind `hash1` und `hash2` in einem Protokolldurchlauf identisch. Angenommen, ein Angreifer verändert den übertragenen MAC, gilt nicht `hash1` = `hash2`. Nur wenn beide Seiten die gleichen Nonces haben, sich die daraus berechneten MAC-

Werte (hash1 und hash2) gleichen und der übertragene MAC-Wert mit diesen übereinstimmt, werden in einem Protokolldurchlauf erst beginCon(x) und dann endCon(x) mit gleichem x erzeugt. Die Implikation in Zeile 42 (Listing 4.2) bedeutet: Wenn das Ereignis endCon(x) erfolgte, dann muss ihm in der gleichen Protokollinstanz ein beginCon(x) vorhergegangen sein. Wenn dies der Fall ist, dann ist die Authentifizierung der Gegenstelle an der Containereinheit erfolgt. In Zeile 42 wird dem ProVerif-Werkzeug somit die Anweisung erteilt, diese Implikation nachzuweisen. Gleiches geschieht für die Authentifizierung der Containereinheit an der Gegenstelle in Zeile 43. Die entsprechenden Ereignisgenerierungen erfolgen in der Modelldefinition in Zeile 68 (beginSvr) und 87 (endSvr) des Listings 4.1.

Replay-Angriff Unter einem Replay wird das erneute Übermitteln oder Abspielen verstanden. Ein Replay-Angriff ist der Versuch, ein bereits gesendetes oder abgefangenes Datenpaket zu einem späteren Zeitpunkt erneut einzuschleusen[4]. Es muss also sichergestellt sein, dass ein Authentisierungsversuch (das entsprechende begin-Event) eines vorherigen Protokolldurchlaufes nicht zu einer Authentifizierung (end-Event mit gleichem Parameter x) in einem anderen Durchlauf führt. Die Zeilen 45 und 46 (Listing 4.2) repräsentieren die entsprechenden injektiven[5] Anfragen.

Geheimhaltung und Vollendung Die Anfragen der Zeilen 47 und 48 veranlassen ProVerif zur Überprüfung, ob die beiden Prozesse bei jedem Durchlauf korrekt beendet werden. Auf diese Weise kann ausgeschlossen werden, dass sich ein Prozess in einem undefinierten Zustand befindet. Die letzte Anfrage (Zeile 49) bezieht sich auf die Geheimhaltung der Ereignisse: Ist es einem Angreifer möglich, bei diesem Modell an die Nachricht m zu gelangen?

ProVerif beantwortet die gestellten Anfragen auf Kommandozeile wie folgt:

```
//The secrecy of message m is achieved:
RESULT not attacker:m[] is true.

//These two queries are to prove protocol completion:
RESULT not ev:endSvr(x_250) is false.
RESULT not ev:endCon(x_635) is false.

//These two queries are to prove non-injective agreement (Authenticity):
//Authenticity of container to the server is achieved:
RESULT ev:endSvr(x_1433) ==> ev:beginSvr(x_1433) is true.
//Authenticity of server to container is achieved:
RESULT ev:endCon(x_1662) ==> ev:beginCon(x_1662) is true.
```

[4] TAN-Nummern beim Online-Banking sind nicht gegen Replay geschützt, da diese immer gültig sind. Die iTAN-Nummern, bei dem das System die exakte TAN-Nummer der Liste vorschreibt, hingegen schon.

[5] Eine injektive Relation beschreibt eine genaue Eins-zu-eins-Beziehung.

```
//These two queries are to prove injective agreement
//Authenticity including the absence from replay attack:
RESULT evinj:endSvr(x_969) ==> evinj:beginSvr(x_969) is true.
RESULT evinj:endCon(x_1205) ==> evinj:beginCon(x_1205) is true.
```

ProVerif konnte für alle Anfragen unter den getroffenen Annahmen die geforderten Sicherheitsanforderungen nachweisen. Kein Angreifer ist in der Lage, an die Nachricht `m` zu gelangen. Die beiden Prozesse enden nicht in einem undefinierten Zustand. Eine Authentifizierung findet nur statt, wenn ein korrekter Authentisierungsversuch vorausging. Schließlich ist dies auch nur innerhalb einer Sitzung (Protokolldurchlaufsinstanz) möglich. Die Ziffern hinter den `x` stehen für die modellierten Werte, die ProVerif zur Identifizierung der Instanzen angenommen hat.

4.4 Ergänzungen

4.4.1 Verschlüsselungsverfahren

Wie bereits ausführlich diskutiert, ist für die Lebensdauer der MASC-Einheit ein minimaler Stromverbrauch von besonderer Bedeutung. Gleichzeitig muss eine ausreichende Sicherheit gewährleistet sein. Interessant ist bei diesem Punkt, dass das MASC-System einige Jahre bis Jahrzehnte in Verwendung sein soll. Es ist davon auszugehen, dass sich Angriffsverfahren in dieser Zeit verbessern und Computer leistungsfähiger werden. Natürlich kann niemand voraussehen, welche Verfahren in zehn Jahren als noch sicher gelten, allerdings hat eine Gruppe der führenden Kryptographie-Forscher Europas eine Studie durchgeführt, die alle Algorithmen auf potenzielle Schwachstellen untersucht. Unter dem Namen NESSIE[6] wurde dieses Projekt von der Europäischen Union gefördert. Der *NESSIE Security Report* [70] analysiert die untersuchten Algorithmen auf deren Sicherheit hin, das Dokument *Performance of Optimized Implementations of the NESSIE Primitives* [71] vergleicht deren Geschwindigkeiten.

Bei der NESSIE-Initiative gab es im Bereich Blockchiffren 22 Kandidaten, von denen lediglich die Hälfte als ausreichend sicher eingestuft wurde. Die elf empfohlenen Chiffren sind: Rijndael (AES), RC6, *IDEA*, Camellia, Safer++, *MISTY1*, *Triple-DES*, *Khazad*, SHACAL-1, SHACAL-2. Die kursiv gesetzten Kryptosysteme verwenden als Schlüssellänge 64 Bit[7]. Diese Schlüssellänge ist aus heutiger Sicht gerade ausreichend, jedoch ist fraglich, ob dies in zehn Jahren noch gilt. Allein das zu erwartende Voranschreiten der Leistungsfähigkeit der Computer führt dazu, dass 64-Bit-Kryptosysteme nicht in die engere Wahl einbezogen werden sollten. SHACAL-1 und SHACAL-2 sind ausschließlich mit 256 Bit Schlüssellänge zu verwenden und daher bezüglich ihres Stromverbrauchs überdimensioniert.

[6] „New European Schemes for Signatures, Integrity and Encryption“

[7] Die Schlüssellänge bei Triple-DES ist zwar länger als 64 Bit, jedoch ist die interne Schlüssellänge bei den einzelnen Verarbeitungsschritten sogar nur 56 Bit lang. Bei Triple-DES wird der gebrochene Algorithmus DES dreimal hintereinander ausgeführt.

Neben der NESSIE-Initiative gab es im Jahr 2000 eine weitere bedeutende Blockchiffren-Analyse. Die amerikanische Standardisierungsbehörde NIST suchte einen Nachfolger für den Kryptographiestandard DES, der nicht mehr modernen Sicherheitsanforderungen entspricht. Im Rahmen der AES[8]-Challenge wurden Kryptographen eingeladen, Blockchiffren als Kandidaten einzureichen. Der Abschlussbericht [61] gibt Auskunft darüber, welche Kriterien bei der Wahl den Ausschlag gegeben haben. Aus 15 Kandidaten wurden lediglich fünf Finalisten gewählt, aus denen dann Rijndael als Sieger hervorging. Daher werden im Folgenden AES und Rijndael synonym verwandt.

Die weiteren vier Finalisten sind MARS, RC6, Serpent und Twofish. Alle diese Verfahren erfüllen die Sicherheitsanforderungen und können in die engere Wahl gezogen werden.

Laut [61] wurde Rijndael zum Sieger bestimmt, weil er sowohl in Hard- als auch Software außergewöhnlich schnell ist und sich gut optimieren lässt. Besonders die Fähigkeit, in Systemen mit beschränktem Speicher zu laufen, führte schlussendlich zum Sieg. Allein dieses Urteil der Experten zeigt, dass vermutlich AES die geeignete MASC-Blockchiffre ist.

RC6 ist eine Blockchiffre, die ebenfalls von beiden Gremien als ausreichend sicher erachtet wird. Es hat den Vorteil, dass es aus einem besonders kurzen Quelltext besteht und nicht sehr komplex aufgebaut ist. Zwei frei übersetzte Zitate aus der AES-Challenge:

1. *RC6 [...] ist allgemein betrachtet der Einfachste der Finalisten. Lediglich die modulare Multiplikation ist wesentlich komplexer als typische Chiffrieroperationen.* [61, Seite 26 oben]
2. *RC6 scheint einen ausreichenden Sicherheitsspielraum zu haben, dennoch gab es Kritik von vielen, die einen geringeren Spielraum als bei den Konkurrenten beanstandeten. Zur Entlastung muss hinzugefügt werden, dass die vielgelobte Einfachheit von RC6 natürlich auch die Analyse vereinfacht hat, zumal nur ein kurzer Zeitrahmen zur Verfügung stand.* [61, Seite 30]

Exemplarisch wurden AES und RC6 für eine beispielhafte Zielplattform optimiert und implementiert (siehe Abschnitt 4.5). Optimierungskriterium Nummer eins ist der Stromverbrauch. In diesen muss mit eingerechnet werden, dass ein größerer Speicheraufwand dazu führt, dass mehr Speicher benötigt wird. Flüchtiger Speicher muss permanent aufgefrischt werden, damit die Daten nicht verloren gehen.

ZigBee ist ein Standard für Sensornetzwerke, der durch die OFDM-Technologie sehr wenig Strom verbraucht. Die norwegische Firma Chipcon stellt einen ZigBee-Chip her, der unter der Bezeichnung CC2430 geführt wird. Dieser benötigt für das Speicherauslesen, das anschließende Verschlüsseln mit AES, das Verpacken in Datenpakete und das abschließende drahtlose Versenden lediglich 25 mA bei 3,6 V. Für das Empfangen und Entschlüsseln werden 27 mA benötigt [20, Seite 11]. Das entspricht einer Leistung von ca. 90 mW. Der CC2430 enthält einen AES-Koprozessor, um diesen geringen Stromverbrauch zu erreichen. Chipcon gibt den Verbrauch des AES-Moduls mit nur 50 $\mu A/MHz$ beim Ver- und Entschlüsseln an. Der ZigBee-Chip läuft im Stromsparmodus mit 16 MHz und im Sende-/Empfangsmodus mit 32 MHz. Das bedeutet, dass im Sende-/Empfangsmodus der AES-Koprozessor nur 1,6 mA

[8] AES steht für *Advanced Encryption Standard*

der 25 *mA* (27 *mA*) benötigt. Dies macht deutlich, dass die Wahl und Optimierung des Verschlüsselungsalgorithmus nur einen kleinen Einfluss auf die Gesamtleistung haben.

Aus Gründen der Stromverbrauchseffizienz ist es sinnvoll, die Verschlüsselungsfunktionalität auf einen speziellen Koprozessor auszulagern und nicht im Prozessor durchzuführen. Da mittlerweile AES-Koprozessoren von vielen Herstellern angeboten werden, ist davon auszugehen, dass die Effizienz dieser Chips entsprechend groß ist. Allerdings geht durch eine Festverdrahtung des Verschlüsselungsalgorithmus die Fähigkeit des späteren Änderns verloren. Es ist also nicht mehr möglich, bei Schwachstellen die Wahl des Algorithmus zu ändern. Inwieweit dies für die Praxis relevant ist, kann zu diesem Zeitpunkt nicht ermittelt werden. Je nach Koprozessor ist es möglich, durch neue Firmware-Versionen die konkrete Implementierung marginal zu verändern, allerdings werden die freien Ressourcen im Chip dieses voraussichtlich sehr einschränken. Auf diese Weise könnten zumindest Implementierungsfehler korrigiert werden. Es ist also abzuwägen, ob eine „Festverdrahtung“ gewünscht wird. Der vermutete Stromminderverbrauch spricht dafür, die Variabilität bei der Wahl des Verschlüsselungsverfahrens dagegen.

4.4.2 Operationsmodi

Das Verwenden einer Blockchiffre mit einem festen Schlüssel führt zu einem pseudozufälligen Chiffrat, das ohne Kenntnis des Schlüssels nichts über seinen Klartext preisgibt. Da eine (deterministische) Blockchiffre[9] bei festem Schlüssel eine eindeutige Funktion ist, werden immer gleiche Klartexte auf immer gleiche Chiffrate abgebildet. Wiederholungen gleicher Klartextblöcke können so im Chiffrat entdeckt werden. In der Praxis haben sich die so genannten Operationsmodi durchgesetzt, die dieses Problem lösen. Dabei werden Informationen vorhergehender Verschlüsselungen verwendet und nachfolgenden Blöcken hinzugefügt. Als Grundlage wird die symmetrische Bit-weise Entweder-Oder-Funktion verwendet, weshalb der Klartext nicht verlängert werden muss.

Anhang A.7 erklärt die fünf Operationsmodi ECB, CBC, CFB, OFB und CTR detailliert. Neben der Erläuterung der Verfahren werden diese in A.7 auf Fehlerfortpflanzung und Paketverlustverhalten analysiert. Weil die Route zwischen MASC-Einheit und Gateway nicht verlässlich ist, Bitfehler oder -verluste also auftreten können, ist das Verhalten bei fehlerhafter Übertragung relevant.

Der ECB, die Variante, keine Abhängigkeiten zwischen Blöcken zu schaffen und jeden Block unabhängig mit dem festen Schlüssel zu verschlüsseln, sollte aus oben genannten Gründen, nicht verwendet werden. Alle weiteren Verfahren eignen sich zur Verwendung im MASC-Szenario.

Die drei Operationsmodi CFB, OFB oder CTR verschlüsseln den Klartext nicht direkt mit der Blockchiffre, sondern berechnen einen Zwischenschlüssel, der für eine Verschlüsselung des Klartextes mit Entweder-Oder-Funktion benutzt wird. Für die Berechnung des Zwischenschlüssels werden die Blockchiffre, der Schlüssel und bei OFB und CFB ein rückgekoppelter Wert aus den Vorberechnungen verwendet. Weil die Blockchiffre nicht zum Verschlüsseln des Klartextes herangezogen wird (welches die

[9] Nicht-deterministische Blockchiffren spielen für diese Arbeit keine Rolle.

Entweder-Oder-Funktion erledigt), sondern lediglich eine pseudozufällige Abbildung unter Verwendung eines Schlüssels darstellt, wird wahlweise nur die Implementierung der Ver- oder der Entschlüsselungsfunktion der Blockchiffre benötigt.

OFB und CTR haben drei weitere Vorteile: Die Rückkopplung ist unabhängig vom Klartext, weswegen das Schlüsselmaterial für die Entweder-Oder-Verschlüsselung vorausberechnet werden kann, das Fehlertoleranzverhalten ist optimal und die zu übertragende Blocklänge ist variabel. Letzteres gilt nicht für die internen Berechnungen und Verknüpfungen auf beiden Seiten, aber für Datenpakete auf dem Übertragungsmedium.

OFB und CTR unterscheiden sich in der Generierung des Zwischenschlüssels. Während beim OFB jeder Zwischenschlüssel durch Rückkopplung mit seinen Vorgängern und der Anwendung der Blockchiffre berechnet wird, wird beim CTR ein vorher übertragener Initialisierungsvektor (IV) schrittweise hochgezählt. Bei OFB steht der vorherige Zwischenschlüssel bei der Berechnung seines Nachfolgers bereits in einem Register. Bei CTR muss hingegen der vorherige Zählerwert immer erst um eins erhöht werden, bevor er im nächsten Schritt verwendet werden kann. Bei der Verwendung des OFB wird also eine Rechenoperation eingespart (dies könnte allerdings abhängig von der Hardwareplattform sein).

Der benötigte IV muss beim MASC-ST ebenfalls nicht zusätzlich übertragen werden. Es können die beiden Nonces oder die ersten Bit der beidseitig bekannten MAC-Funktion der Nonces verwendet werden.

4.4.3 Paddingverfahren

Da jedes per Funk zu übertragende Bit bei Sender und Empfänger Strom verbraucht, sollte die Länge der Nachrichten minimiert werden. Jedes Ereignis der MASC-Einheit besteht aus verschiedenen Sensorschwankungen und jeder Sensor kann eine unterschiedliche Anzahl an Zuständen eingehen. Die Entropie des Ereignisses sollte daher unter der Bedingung maximiert werden, dass eine gewisse Struktur, bestehend aus Ereignis-ID und den Informationen über enthaltene Sensordaten, beibehalten wird. Dies kann nur geschehen, wenn die Sensorschwankungen komprimiert werden. Daraus ergibt sich eine variable Ereignislänge. Ein Türöffnungsereignis besteht aus einer ID, einer Identifizierung des Sensors und einem Bit Information. Dies lässt sich in wenigen Bytes ausdrücken. Die RFID-Nummern aller eingeführten Waren benötigen bei einem Beladevorgang eine Vielzahl von Bytes. Das resultierende Ereignis ist demnach um entsprechende Vielfache länger als das der Türöffnung. Treten zum Beispiel Temperaturschwankungen und Beladevorgänge gleichzeitig auf, verlängern sich diese Ereignisse weiterhin.

Wenn allerdings Operationsmodi wie CBC, ECB, unter Umständen auch CFB, verwendet werden, ist eine feste Blocklänge des Klartextes vorgegeben. Die Konsequenz ist, dass der letzte entstehende Block auf Blocklänge verlängert werden muss. Der Empfänger weiß beim Nachrichtenempfang nicht, wie lang das enthaltene Ereignis ist, das heißt wie lang die Nutzdaten des letzten Blocks sind. Daher gibt es zwei Möglichkeiten, dieses Problem zu lösen. Die erste ist, dass ein zusätzliches Feld im

Nachrichtenkopf verwendet wird, um dem Empfänger mitzuteilen, wie lang die Nachricht ist, und diese dann mit zufälligen Daten aufzufüllen. Dies führt jedoch zu einer festen Datenexpansion, da jedes Paket dieses Feld beigefügt bekommt. Die zweite Variante besteht darin, die Paddingfelder selber zu verwenden, um dem Empfänger mitzuteilen, dass es sich um Padding handelt.

Bei diesem inhärenten Padding sind ebenfalls mehrere Strategien denkbar. Wenn es ein Zeichen gibt, welches nie (beziehungsweise selten) in der Nachricht vorkommt, könnte dieses als Symbol für den Paddinganfang dienen. Falls es doch in der Nachricht enthalten ist, müsste ein Alternativsymbol kreiert werden, welches dieses ersetzt (zum Beispiel zweimal das Zeichen). Wenn aber alle Zeichen gleich wahrscheinlich sind, führt dieses Verfahren ebenfalls zu einer Datenexpansion. Ein anderes häufig verwendetes Schema ist die Verwendung des letzten Bytes für den Wert, der angibt, wie viele Paddingbyte angefügt wurden. Diese werden dann mit einem vorgegebenen Wert beschrieben. Bei diesem Verfahren geht ein Byte pro Block verloren.

Die Wahl des Paddingverfahrens ist durchaus sicherheitsrelevant. Es gab in der Vergangenheit einige Angriffe gegen verschiedene Implementierungen von kryptographischen Algorithmen, die deren Paddingstruktur ausnutzten (zum Beispiel [104, 11, 48]). Die Angreifer haben sich die Eigenschaften des Paddingverfahrens zu Nutze gemacht und so Informationen über den Klartext erhalten. Diese Informationen haben gereicht, um das Verfahren in dieser Implementierung zu brechen.

Bei den Operationsmodi ECB und CBC gehen die Teilblöcke der Nachricht direkt in die Verschlüsselungsfunktion ein. Bei CBC wird vorher noch ein Entweder-Oder durchgeführt. Es ist bei diesen beiden Verfahren erforderlich, dass alle Blöcke die richtige Länge haben, also Padding verwendet wird.

Bei CTR und OFB geht der Klartext (gemäß Anhang A.7) nicht in die Verschlüsselungsfunktion ein. Der letzte Klartextblock muss nicht mit einem Paddingverfahren aufgefüllt werden, da OFB und CTR eine variable Blocklänge unterstützen.

Beim CFB geht der Klartext ebenfalls nicht in die Verschlüsselungsfunktion ein, jedoch das letzte Chiffrat. Wenn die Länge dieses Chiffrats variabel gehalten werden soll, muss es bei der Rückkopplung auf Teilchiffratlänge erweitert werden. Dies kann durch ein Paddingverfahren geschehen. Allerdings kann beim CFB Padding des Klartextes umgangen werden, indem Padding in der Rückkopplungsschleife durchführt wird. Dies hätte den Vorteil, dass beide Seiten wissen, wie lang das Chiffrat ist und wie viele Byte Padding anzuhängen sind. Es fände also keine Datenexpansion statt. Im Gegensatz zu OFB/ CTR würde das Verfahren aber komplexer.

Es empfiehlt sich also die Verwendung von OFB ohne Padding. Sollte dennoch CBC verwendet werden oder aus Hardwaregründen ein Verfahren mit variabler Chiffratlänge unerwünscht sein und somit eine konstante Blocklänge notwendig werden, empfiehlt sich die Verwendung von Padding. Das Paddingverfahren aus Standard PKCS #5 [75] funktioniert wie nachfolgend beschrieben: Das letzte Byte geht als Nutzfeld verloren und enthält immer P, die Anzahl der Paddingbyte. Wird kein Padding benötigt,

weil der Block voll ausgenutzt wird, enthält das letzte Byte den Wert 0. Wird ein Byte Padding angehängt, stehen in den letzten beiden Byte eine 1. Auf diese Weise werden beim Padding nach PKCS #5 die letzten $P+1$ Felder immer mit der Zahl P aufgefüllt.

4.5 Machbarkeitsstudie

Im Rahmen einer Machbarkeitsstudie wurden entscheidende Blöcke des MASC-ST mit ScatterWeb-Sensorknoten[10] implementiert. Diese Knoten sind eine erste Annäherung an speziell zu erstellende Hardware, die für die MASC-Anwendung notwendig ist. Allerdings geben diese Standard-Sensorknoten einen guten Einblick, ob die jeweilige Funktion realisierbar ist und welche Probleme dabei zu erwarten sind.

4.5.1 RC6 und MASC-ST

Um die Komplexität des Tunnelgenerierungsprotokolls zu zeigen, wurde ein in seinem Umfang eingeschränktes MASC-ST auf den Sensorknoten unter Verwendung von RC6 als Verschlüsselungsfunktion implementiert. Das für 32-Bit-Rechner entworfene RC6 ließ sich nach der Implementierung von Salvo Salasio[11] ohne größere Probleme auf die 16-Bit-Knoten portieren.

Die Analyse des vom Compiler erzeugten Assembler-Codes hat ergeben, dass die Registerrotationen auf dem MSP430 sehr aufwändig sind. Während ein x86-Prozessor Registerrotationen um beliebige Bitanzahlen mit einem Befehl ausführt, kann der MSP430 Registerinhalte nur um jeweils ein Bit schieben. Eine Registerrotation muss daher aufwändig durch Schleifen dargestellt werden, wozu mehrere Register erforderlich sind. Der Versuch, den Assemblercode dieser Rotationsmakros zu optimieren führte dazu, dass sich der Umfang des nötigen Codes drastisch verringern ließ. Allerdings war dadurch kein Geschwindigkeitszuwachs (und keine Stromverbrauchsreduktion) messbar.

Die Analyse der Geschwindigkeit spielt für die MASC-Anwendung in erster Linie keine Rolle, da es unwichtig ist, wie schnell der Algorithmus läuft. Es hat sich allerdings herausgestellt, dass es für den Energieverbrauch sehr wohl relevant ist. Messungen haben ergeben, dass der MSP430-Prozessor einen nahezu konstanten Stromverbrauch hat. Es spielt also keine Rolle, welche Funktionen ausgeführt werden. Allein die Ausführungsdauer ist entscheidend.

Tabelle 4.1 zeigt, dass der Mikrocontroller der Sensorknoten unter Volllast immer die annähernd selbe Stromaufnahme besitzt und die Art oder Implementierung des Algorithmus keinen Einfluss auf den Energieverbrauch hat. Der MSP430 hat eine mittlere Stromaufnahme von ungefähr 1,15 *mA* – ausschlaggebend für den Energieverbrauch ist also die Rechenzeit, nicht die Art der bearbeiteten Algorithmen. Zu einem identischen Schluss kam die Gruppe der Universität Twente, die mit dem gleichen Mikroprozessor arbeitet [56].

[10] Eine genaue Erklärung der verwendeten Hardware ist im Anhang D zu finden.
[11] `http://www.ussrback.com/crypto/misc/rc6.c`

	Leerlaufstrom in *mA*	Laststrom in *mA*	Δ*mA*
MD2	15,40	16,60	1,20
MD4	15,38	16,56	1,18
MD5	15,40	16,56	1,16
RC6 (Schlüsselberechnung)	15,45	16,60	1,15
RC6 (Verschlüsselung)	15,45	16,56	1,11
RC6-CBC (Schlüsselberechnung)	15,45	16,59	1,14
RC6-CBC (Verschlüsselung)	15,45	16,56	1,11
HMAC	15,45	16,57	1,12
Mittelwert			$\approx 1{,}15$
Standardabweichung			$\approx 0{,}03$

Tabelle 4.1: Stromaufnahme der ESBs bei der Ausführung verschiedener Algorithmen [80]

Die Geschwindigkeitsmessungen für RC6 sind Tabelle 4.2 zu entnehmen.

	Schlüsselberechnung	Entschlüsselung	Verschlüsselung
1 block RC6	24,6 *ms*	3,6 *ms*	3,6 *ms*
3 blocks RC6-CBC	24,6 *ms*	10,8 *ms*	10,9 *ms*

Tabelle 4.2: Geschwindigkeitsmessungen RC6

Aus Komplexitätsgründen wurde in dieser Implementierung auf die Unterscheidung zwischen Langsitzung und Kurzsitzung verzichtet, des Weiteren kam CBC als Operationsmodus zum Einsatz. Daher war es auch nötig, ein Paddingverfahren einzuführen. Es wurde das PKCS-#5-Verfahren [75] (gemäß Abschnitt 4.4.3) verwendet.

Eine interessante Eigenschaft des RC6-CBC-Verfahrens ist, dass dieses auch als MAC-Funktion verwendet werden kann. Speicherplatzsparend wäre also der Einsatz des RC6-CBC-MAC, da keine MAC-Funktion neu implementiert werden muss. Allerdings benötigt diese für die Berechnung zweier Blöcke $31,8\ ms$[12]. Da der Vorteil einer MAC-Funktion deren schnelle (und damit stromsparende) Berechenbarkeit gegenüber einer Verschlüsselung ist und darauf auch das MASC-IS-Protokoll ausgelegt ist, wurde eine Verwendung dieser Variante verworfen. Stattdessen wurde ein HMAC[13] auf Basis der Hashfunktion MD5 implementiert. HMAC-MD5 kommt bei Berechnung zweier Blöcke auf eine Ausführungsdauer von 7,3 *ms*.

Die Hashfunktion MD5 gilt mittlerweile als nicht mehr sicher gegenüber Kollisionsangriffen. Die Eigenschaft der Kollisionsresistenz (und eingeschränkt der 2nd-Preimage-Resistance) ist bei ihr nicht mehr gegeben. Jedoch spielt das für die Anwendung keine Rolle. Weder eine falsche MASC-Einheit noch eine falsche Gegenstelle zögen einen Vorteil daraus, eine MAC-Kollision zu berechnen und dem Gegenüber eine gefälschte Information zu unterstellen, da keine spätere Nachweisbarkeit (*Non-Repudiation*) einer Nachricht eingeholt wird. Die MAC-Funktionen werden ausschließlich zur Authen-

[12] Schlüsselberechnung $+ 2 *$ Verschlüsselung $= 24{,}6\ ms + 2 * 3{,}6\ ms = 31{,}8\ ms$

[13] MAC-Funktion auf Basis einer Hashfunktion, siehe Anhang A.2.

tisierung und Integritätswahrung der gesendeten Nachricht verwendet. Einzig ein Mann-in-der-Mitte könnte eine Nachricht abfangen und mit verändertem Inhalt weitersenden. Damit diese von Empfänger angenommen wird[14], muss eine Kollision zur abgefangenen Nachricht generiert werden (Preimage-Resistance). In dieser Hinsicht gilt MD5 aber noch als ausreichend sicher. Selbst wenn MD5 die Preimage-Resistance oder die Nicht-Umkehrbarkeit in einigen Jahren nicht mehr gewährleisten kann, wäre der Angreifer höchstens in der Lage, den Kurzsitzungsschlüssel zu berechen und sich als falsche Gegenstelle auszugeben. Er wird es allerdings nicht schaffen, falsche Informationen an die Gegenstelle zu senden, da dafür die Kenntnis des Langsitzungsschlüssels erforderlich ist. Wenn die Zielhardware ausgelegt wird, sollte dennoch die Verwendung moderner Hashfunktionen als Basis für die MAC-Berechnung bevorzugt werden.

Aus der Arbeit mit den ScatterWeb-Knoten können für die MASC-Anwendung gute Erkenntnisse gezogen werden, die später in die Spezifizierung und Konstruktion speziell angepasster Hardware einfließen können. Aufgrund der Stromaufnahmemessungen mit den Knoten ist es fraglich, ob andere Standardhardware die gestellten Anforderungen überhaupt erfüllen kann, oder ob exakt für den gewünschten Einsatzzweck Hardware erstellt werden muss. Die Auslagerung der Kryptographiefunktionen in speziell angepasste Hardware kann sehr wohl sinnvoll sein, um die beiden Anforderungen von geringem Energieverbrauch und guter Leistung zu vereinigen. (Detailliertere Informationen sind [80] zu entnehmen.)

4.5.2 AES

Nachdem Rijndael die AES-Challange gewonnen und sich zunehmender Kryptanalyse unterzogen hat, entwickelte sich die Blockchiffre zum Standard-Verfahren. Im Bereich der Sicherheit und Effizienz halten Experten diesen Algorithmus für den besten (siehe dazu Abschnitt 4.4.1). Im Rahmen einer Testimplementierung (siehe [106]) wurde für die Feldtestknoten die Effizienz von AES untersucht.

Eine der Stärken von AES ist, dass viele aufwändige Operationen vorausberechnet und in Tabellen abgelegt werden können. Dies gilt nicht nur für alle Funktionen, sondern auch für Teilfunktionen. Auf diese Weise sind 18 verschiedene Implementierungen entstanden, die jeweils einen unterschiedlichen Speicherbedarf haben.

Tabelle 4.3 vergleicht verschiedene AES-Varianten. Gemessen wurden der Speicherbereich und die Zeiten für Schlüsselgenerierung, Verschlüsselung und Entschlüsselung. Dabei wurde ein Speicherbereich mit willkürlichen 16 Byte 1 000 Mal rekursiv verschlüsselt und anschließend entschlüsselt. Die Zeit für eine einzelne Entschlüsselung eines Blockes ist somit ein Tausendstel des Tabellenwertes. Die jeweiligen Durchläufe unterscheiden sich in der Parameterwahl, der durchgeführten AES-Implementierung. Normalerweise durchläuft eine AES-Verschlüsselungsrunde vier Stufen: Vertauschung (*Substitution*), Reihenverschiebung (*ShiftRow*), Spaltenvermischung (*MixColumn*) und Schlüsseladdition (*KeyAddition*). Es gibt aber auch die Variante, AES in nur einer Stufe ablaufen zu lassen. Hierfür

[14] MAC-Funktion der ursprünglichen Nachricht ist ebenfalls anwendbar auf neue Nachricht.

AES-Variante	Speicherverbrauch in Bytes	Schlüsselgenerierung in Millisekunden	Verschlüsselung in Millisekunden	Entschlüsselung in Millisekunden
1	9 040	5 076	4 065	4 107
2	9 058	4 922	4 066	4 083
3	9 068	4 843	4 064	4 106
4	9 118	4 794	4 042	4 107
5	9 254	4 767	4 067	4 108
6	8 586	4 860	8 517	4 106
7	8 604	4 754	8 574	4 100
8	8 622	4 796	8 563	4 107
9	8 800	4 772	8 557	4 092
10	8 366	221 446	198 603	4 101
11	7 750	228 300	689 341	4 105
12	7 768	119 077	361 065	4 103
13	7 778	64 601	196 718	4 102
14	7 828	12 722	41 500	4 100
15	7 964	8 644	28 818	4 101
16	8 114	5 230	18 379	4 098
17	14 732	2 764	1 670	1 623
18	13 744	2 764	2 199	1 624
Durchschnitt:	9 122	39 396	88 489	3 826

Tabelle 4.3: Messungen verschiedener AES-Implementierung [106]

werden vier Tabellen (FT_0, FT_1, FT_2 und FT_3) benötigt, aus denen die entsprechenden Werte in einem Schritt abgelesen werden können, ohne dass die jeweilige arithmetische Operation durchgeführt werden muss. Da die Tabelle FT_i aus ihrem Vorgänger FT_{i-1} berechenbar ist, braucht nur FT_0 vorzuliegen. Die benötigten Teile der anderen können zur Laufzeit einfach berechnet werden. Es ist auch möglich, die Tabelle FT_0 zur Laufzeit zu berechnen, allerdings ist dies etwas aufwändiger. Werden die vier Stufen bei jeder Verschlüsselungsrunde durchgeführt, erfolgt für jede Vertauschungsoperation eine Verknüpfung mit einer S-Box. Alle 256 Einträge derselben können entweder vorher im Speicher abgelegt oder iterativ berechnet werden. Es ist auch möglich, nur 64 Einträge (jeden vierten) zu speichern, um auf diese Weise höchstens drei Iterationsschritte für die Berechnung jedes S-Box-Eintrages zu benötigen. Sinngemäß kann mit 16, vier oder nur zwei Einträgen der durchschnittliche Iterationsaufwand schrittweise um den Faktor zwei vergrößert werden.

Die Tabelleneinträge eins bis fünf gehören zu AES-Implementierungen, bei denen AES nur mit einer Stufe berechnet wird und die S-Box mit einem, zwei, vier, 64 beziehungsweise 256 Einträgen vorausberechnet wurde. Dies kann in der Tabelle gut nachvollzogen werden, da der Speicherverbrauch schrittweise entsprechend größer wird, die Geschwindigkeit aber konstant bleibt. Die Vier-Stufen-Version ist um 554 Byte kleiner, was beim Vergleich zwischen Version eins und sechs zu sehen ist. Die Versionen sechs bis neun entsprechen AES-Implementierungen mit vier Stufen, bei denen eins, zwei, 16 und 256 Iterationen vorausberechnet wurden. Da die S-Box bei den Versionen eins bis neun aber komplett im ROM abgelegt wurde, wirkt sich die Wahl der Iterationen nicht auf deren Berechnung aus. Dies ändert sich bei den AES-Versionen zehn bis 15. Hier wurde die S-Box nicht im ROM des

Sensorknotens abgelegt, sondern iterativ berechnet. Version zehn entspricht dabei Version sechs ohne S-Box im ROM und Version elf der Version eins. Schlüsselgenerierungs- und Verschlüsselungsdauer steigen stark an. Bei den nachfolgenden Versionen wird die S-Box-Vorausberechnung schrittweise erhöht. Auf diese Weise entsprechen die Versionen elf bis 15 den Versionen eins bis fünf – mit dem Unterschied dass sich die Iterationen diesmal auswirken, weil die S-Box nicht im ROM abgelegt ist. Der Speicherverbrauch steigt leicht, die Rechenzeit sinkt drastisch. In Version 16 wurde, anders als bei den Versionen eins bis zehn, der FT_0 aus dem ROM entfernt. Die Version 16 entspricht somit Version fünf ohne FT_0 (im Gegensatz zu 15 wurde bei 16 die S-Box wieder hinzugefügt). Beim Vergleich zwischen 15 und 16 wird deutlich, dass die Tabelle FT_0 knapp 200 Byte größer als die S-Box, Variante 16 aber deutlich schneller ist. Bei Variante 17 wird alles in den Speicher geladen, was möglich ist. Neben der S-Box und FT_0 werden auch FT_1, FT_2 und FT_3 gespeichert. Gleiches passiert mit den entsprechenden Äquivalenten für die Entschlüsselung, die bislang nicht variiert wurden. In Version 18 wurde FT_3 aus dem Speicher entfernt und muss zur Laufzeit berechnet werden.

Die 18 verschiedenen Versionen zeigen deutlich, wie die Parameter an die Wünsche angepasst werden können. Die Extreme sind dabei die Versionen elf (minimaler Speicherbedarf) und 17 (maximale Geschwindigkeit). Ein guter Kompromiss scheint die Version eins zu sein, bei der die Tabelle FT_0 und die S-Box im Speicher sind, die Tabellen FT_1, FT_2 und FT_3 aber zur Laufzeit berechnet werden müssen.

Um die Aussagen in der Tabelle zu verdeutlichen, wurde der Mittelwert über alle Messungen genommen und deutliche Abweichungen farblich abgesetzt. Positive Abweichungen wurden grün dargestellt und negative rot. Da die ScatterWeb-Sensorknoten verwendet wurden, gilt auch hier, dass sich die Laufzeit der Algorithmen proportional zum Stromverbrauch verhält.

Bei RC6 wurde die Berechnung eines Blocks mit 3,6 *ms* angegeben. Bei AES dauert die Verschlüsselung eines Blocks zwischen 1,7 *ms* und 700 *ms*.

4.5.3 Zufallszahlengenerator

Um die benötigten kryptographischen Operationen durchführen zu können, wird ein Zufallszahlengenerator (RNG – engl. *random number generator*) benötigt, der unvorhersagbare Zahlenwerte generiert. Die Annahme, dass etwas nur entweder vorhersagbar oder nicht vorhersagbar sein kann, ist allerdings falsch. Viele Angriffe gegen RNG basieren auf statistischer Vorhersagbarkeit und der Tatsache, dass einige Werte wahrscheinlicher sind als andere, so dass eine GEWISSE Vorhersagbarkeit besteht [45]. Erschwerend kommt hinzu, dass die als Annäherung an die Containereinheit betrachteten ScatterWeb-Sensorknoten über wenig Rechen-, Speicher- und Energieressourcen verfügen. Ein kryptographisch verlässlicher Pseudozufallszahlengenerator (PRNG)[15] benötigt aber rechenintensive Verfahren.

[15] Ein PRNG ist ein Softwarebaustein, der einem RNG nahe kommt. Allerdings benötigt er eine externe zufällige Quelle (Saat).

Pseudozufallszahlengenerator Eine sehr einfache Methode, einen PRNG zu realisieren, ist die Verwendung einer kryptographischen Hashfunktion [60, Abschnitt 5.3]. Bei einer zufälligen Saat[16] erzeugt zum Beispiel SHA-1 unvorhersagbare Werte. Da SHA-1 als Standard-Hash-Algorithmus bereits für die Sensorknoten existiert, wurde dieser für eine Testimplementierung als PRNG verwendet. Damit wird die Güte des resultierenden RNG allein aus der Güte des Saatwertes bestimmt.

Zufällige Saat Das Erstellen einer zufälligen Saat ist unter Zuhilfenahme perfekter analoger Sensoren (theoretisch) einfach. Die Nachkommastellen der Temperatur und der Luftfeuchte sind nicht vorhersagbare Werte. Die Frage ist, ob die verwendeten Sensoren mit anschließender Analog-Digital-Wandlung immer noch eine genügende Menge zufälliger Information enthalten.

	Minimale Entropie	Shannon-Entropie
BATTERYVOLTAGE	3,84	4,88
EXTERNVOLTAGE	4,11	4,77
RPXVOLTAGE	4,93	6,56
MICVOLTAGE	4,28	4,28

Tabelle 4.4: Gemessene Entropie verschiedener Sensoren in Bit[96]

Um eine Information zu erhalten, wie zufällig die Inhalte der Register sind, die die niederwertigen Daten der Analog-Digital-Wandlung der Sensoren speichern, wurde über eine Reihe von 65 000 Messungen[17] die Entropie nach Shannon berechnet. Gleichzeitig wurde dabei ebenfalls die minimale Entropie festgestellt, die bei den Messungen auftrat. Diese Register stellen 12 Bit zur Verfügung und Versuche mit der NIST-Testumgebung [89] ergaben, dass dieses eine gute Basis für Zufallszahlen bildet. Die Resultate schwanken zwischen vier bis sechs Bit Entropie pro 12-Bit-Register. Per Akkumulation kann die erhaltene Gesamtentropie weiter erhöht werden. Wegen der Hardwarebeschränkungen wird für die Erstellung der Saat auf den Sensorknoten die Entweder-Oder-Funktion verwendet und verschiedene gemessene Werte einfach verknüpft, um eine näherungsweise ideale Entropie zu erhalten.

Funktionsweise Der Zufallszahlengenerator besteht aus zwei Teilen, einem Entropiesammler und einem PRNG. Ersterer erstellt ein 512 Bit großes Reservoir an zufälligen Bits (engl. *randomness pool*), aus denen die verschiedenen Saaten gezogen werden. Stehen noch keine 512 Bit zur Verfügung, werden diese pseudozufällig aufgefüllt – ohne volle 512 Bit Entropie zu erhalten. Die Ausgabewertgröße von SHA-1 beträgt 160 Bit, laut [6] enthalten Abbildungen von Zufallszahlen mit SHA-1 mindestens 128 Bit Entropie (und höchstens 160 Bit). Das Zufallszahlen-Reservoir sollte also eine Mindest-Entropie von 160 Bit aufweisen, damit es nicht der limitierende Faktor ist.

Systemtest Es wurden 10 000 Zufallszahlen generiert und auf diese Weise 1,6 Millionen Zufallsbit erzeugt. Die statistische Auswertung erfolgte nach der vom NIST standardisierten „Statistical Test

[16] Der Saatwert ist die Eingabefunktion eines PRNG, der dessen maximale Güte der Zufälligkeit bestimmt.
[17] Im FIPS-Standard 140 sind (für den Monobittest) nur 20 000 Messungen vorgeschrieben.

Suite“ (STS) [89]. Keine der im STS zusammengefassten Testfunktionen zeigte, dass die Zufallszahlen nicht zufällig seien. Zufälligkeit können diese Tests nicht nachweisen, lediglich das Ausbleiben von Vorhersagbarkeit (im Sinne der obigen Wortbedeutung). Dem Vergleich mit anderen benutzten kryptographischen Zufallszahlengeneratoren hält die Testimplementierung stand. Dieser benötigt dazu lediglich die Sensorregisterwerte, eine Entweder-Oder-Funktion und den SHA-1-Algorithmus und ist somit sehr ressourcensparend. Die Ergebnisse reichen aus, um die Machbarkeit eines Zufallszahlengenerators für eine Containereinheit als Gegeben anzunehmen.

4.5.4 Feldversuch

Zum Ziel eine Machbarkeitsstudie erstellen zu können, wurde ein Prototyp einer Containereinheit erstellt[18]. Dieser sendet über Radiowellen ein aktuellen Statusreport an eine gegenstelle, die per HTTP die Daten an einen Server übermittelt. Dieser Speichert die Sensorwerte in einer Datenbank ab und gibt sie über einen Webserver aus.

Abbildung 4.4: Feldversuch Containereinheit

Die Anwendung basiert auf den ScatterWeb-Sensorknoten und benötigt 36706 Bytes des 60 Kilobyte umfassenden Flashspeichers. Von den 2 Kilobytes RAM werden 1613 benötigt, wobei dies bereits die abgelegten Messungen beinhaltet.

Auf der Testeinheit wurden keine kryptographischen Operationen durchgeführt, da der Feldversuch die funktionelle Machbarkeit der Sensorik und die Anforderungen an die Robustheit zum Ziel haben.

Die Tests unter Laborbedingungen liefen positiv ab. Alle Sensoren haben gemäß den Anforderungen funktioniert. Die Containereinheit ist so entworfen, dass sie neben einer permanenten Überwachung

[18] Das Insititut Telematik der TU Hamburg-Harburg erstellte diesen Prototypen im Rahmen eine Projektseminars.

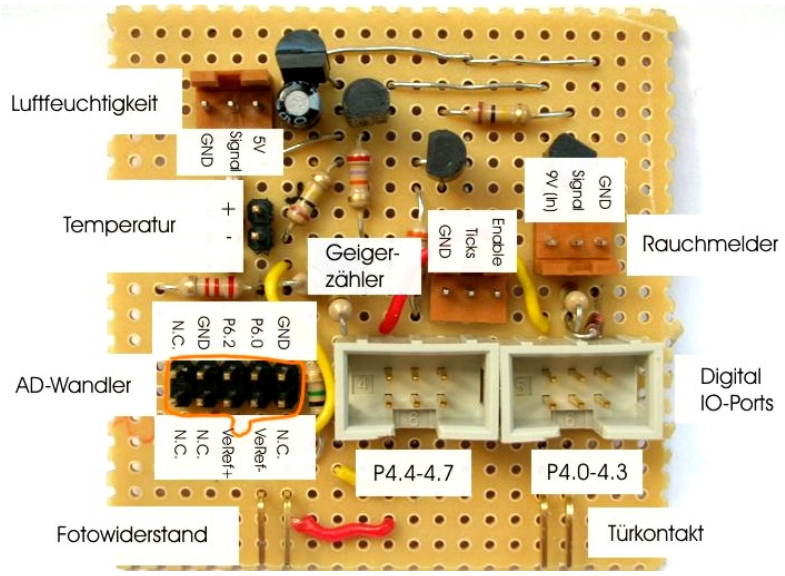


Abbildung 4.5: Sensoransteuerung

auch mit Handlesegeräten funktioniert, mit denen Mitarbeiter alle Container abgehen können. Diese Variante führt zu geringeren Kosten in der Übergangsphase. Test in realen Umgebungen stehen bevor.

4.6 Schlussbetrachtung MASC-ST

Die Absicherung der Datenübertragung zwischen Containereinheit und MASC-Gegenstelle ist ein wichtiger Baustein in der benötigten Sicherheitsarchitektur. Die drahtlose Übertragung zwischen Container und Basisstation über die Luftschnittstelle ist leicht abzuhören, wenn der Angreifer in Reichweite kommt. Auf Autobahnen oder in Bahnhöfen ist dieses für potenzielle Angreifer leicht möglich. Auf Containerstellplätzen könnten Mitarbeiter des Stellplatzbetreibers oder Einbrecher die Luftschnittstelle erreichen. Befindet sich der Container auf hoher See, ist es für Angreifer schwierig, unerkannt und persönlich in Containerreichweite zu kommen, allerdings ist es denkbar, dass benachbarte Container das Abhören oder Angreifen der Luftschnittstelle übernehmen. Auf dem weiteren Weg verläuft die Datenübertragung zwischen Basisstation und Gateway. Auch hier kommen nur Angreifer in Frage, die sich Zugang zum lokalen Umfeld verschaffen können. Die Kommunikation zwischen Gateway und Gegenstelle hingegen läuft durchs Internet. Die Angriffe auf dieser Schnittstelle sind von überall lancierbar. Zwar sind sie schwerer durchzuführen, weil die Route der Pakete ermittelt werden muss, dafür aber anonym möglich.

Um illegales Abhören oder Injizieren von Falschmeldungen zu verhindern, baut das MASC-ST einen kryptographischen Tunnel zwischen Containereinheit und Gegenstelle auf. Dadurch ist die Vertraulichkeit der Informationen gewährleistet. Beide Teilnehmer authentifizieren sich gegenseitig, indem sie den Besitz eines geteilten Geheimnisses beweisen, ohne dieses Geheimnis anderen zugänglich zu

machen. Die Wirksamkeit dieses Verfahrens basiert daher auf der Annahme, dass keine andere Partei in Besitz dieses Geheimnisses gelangt. Die Sicherstellung dessen ist Aufgabe von Maßnahmen in der Containereinheit (siehe Abschnitt 6.1) und bei der Gegenstelle. Um letztere abzusichern, kann allerdings auf Standardmethoden zurückgegriffen werden, die bei vielen gewöhnlichen Internetdiensten bereits implementiert sind.

Die MASC-Anwendung verwandelt die Containereinheiten und die Basisstation zu einem kleinen Sensornetzwerk, bei dem die Container zu Sensorknoten werden. Allerdings handelt es sich dabei keinesfalls um ein typisches Sensornetzwerk, bei dem gleichberechtigte Knoten untereinander eine weiterleitende Kommunikation aufbauen und sich gegenseitig vernetzen. Die Besonderheit liegt in der Tatsache, dass der Kommunikationsendpunkt außerhalb des Netzwerks liegt und die Basisstation nur ein Übermittler ist. Da jeder Knoten nur direkte Kanäle aufbaut, gelten Anforderungen an das Protokoll, die eher einem Client-Server-Protokoll aus den achtziger Jahren nahekommen als einem heutigen Sensornetzwerk. Gleichermaßen gelten aber die Sensornetzwerkanforderungen für die Knoten bezüglich Energieeffizienz, Datenübertragung und Kommunikationssicherheit. Durch den Verzicht auf Meshing ist kein kooperatives Verhalten erforderlich und keine Routingverfahren müssen angewendet werden.

Die Testimplementierungen haben gezeigt, dass das auf symmetrischer Verschlüsselung basierende MASC-ST mit geeigneter Hardware realisierbar ist. Allerdings kam die verwendete Hardware bei den Tests so sehr an ihre Grenzen, dass die Annahme nicht zu bewerkstelligender Umsetzung von asymmetrischer Kryptographie (auch mit anderer Hardware) wohl zutrifft. Die Anforderung nach einer mehrjährigen Lebensdauer erfordert aber auch mit dem MASC-ST eine besondere Hardwareauswahl und extrem stromsparendes Verhalten der Software auf der Containereinheit.

5 MASC-ID – Delegierter Zugriff auf logistische Daten

Durch die Verwendung des MASC-ST werden Integrität und Vertraulichkeit des Datentransports zwischen MASC-Einheit im Container und der Gegenstelle im Internet sichergestellt. Die Gegenstelle speichert die Ereignisse in der dafür vorgesehenen Datenbank ab. Die Aufgabe des MASC-Informationsdiensts (MASC-ID) ist die Weitergabe der Daten an die berechtigten Endpartner, wobei die Informationssicherheit gewahrt bleiben soll. Das Datenbanksystem auf dem Server bei der RSO ist dabei das Bindeglied zwischen Gegenstelle und MASC-ID. Es folgen zwei Definitionen für den nachfolgenden Gebrauch.

Definition 5.1 (Datenbanksystem)**.** *Das Datenbanksystem ist der Dienst, der alle Datenbanken enthält und bereitstellt. Er regelt in letzter Instanz, welche Parteien auf welche Daten in den Datenbanken zugreifen dürfen. Für das MASC-Szenario unterscheidet er aber nur zwischen dem Zugriff von Gegenstelle und MASC-ID. Der Zugriff der logistischen Parteien läuft über den MASC-ID, der bezüglich des Datenbanksystems die zugreifende Partei ist.*

Definition 5.2 (Datenbank)**.** *Das Datenbanksystem enthält viele Datenbanken. Eine Datenbank ist dabei eine logisch getrennte Einheit, die keine Verbindungen über die Grenzen der Datenbank hinaus hat. Bestandteile der Datenbank sind Tabellen mit verschiedenen Rechten. Die Tabellen enthalten Datensätze, die die Reihen der Tabellen bilden. Die Spalten repräsentieren die Datentypen, die jeder Datensatz enthält. Innerhalb einer Datenbank kann es Querverweise zwischen den einzelnen Tabellen geben.*

Für jeden Transport legt der MASC-ID eine neue Datenbank im MASC-Datenbanksystem an. Diese transportabhängige Datenbank enthält mehrere Tabellen. Auf die Tabelle Containerereignisse erhält die Gegenstelle Schreibrechte und der MASC-ID Leserechte. Die Spalten stehen für alle im Container befindlichen Sensoren (zuzüglich Ereignis-ID, Lokalisierungsinformation und aktuellem Frachtführer). Jedes empfangene Ereignis wird von der Gegenstelle entschlüsselt und in die Datenbank übertragen. Eine zweite Tabelle enthält Metainformationen für den Transport. In dieser sind die für die Kommunikation notwendigen geheimen Schlüssel gespeichert. Nur die MASC-Gegenstelle erhält Dateizugriff. Welche Tabellen der MASC-ID benötigt, soll dieses Kapitel erläutern.

5.1 Problemstellung

Der MASC-ID soll zwei Probleme lösen. Die erste Aufgabe ist die Zugriffsregelung von Shipper, Versicherung, Spedition, Zollbehörden und der Logistiker auf die Sensorinformationen. Darüber hinaus soll die bestehende neue Infrastruktur gleich dazu genutzt werden, die Vertragsabläufe zu unterstützen, um Manifestinformationen automatisiert bereitzustellen.

5.1.1 Zugriff auf Containerereignisse

Die Containerereignisse werden von der Gegenstelle in Empfang genommen und in einer Datenbanktabelle gespeichert, so dass der MASC-ID lesenden Zugriff hat. Der MASC-ID bietet als zentralen Dienst das Abrufen dieser Sensordaten an. Die Informationen über potenzielle Transportschäden oder Transportfehler, die aus den Sensordaten gezogen werden können, sind als vertraulich zu betrachten und sollten nicht für jeden verfügbar sein. Es ist daher von Bedeutung, dass nur beim Transport involvierte Parteien Datenzugriff erhalten. Noch feiner lässt sich sogar einschränken, dass keine Notwendigkeit besteht, dass ein Frachtführer, der seine Ware bereits übergeben hat, sehen kann, wie andere Frachtführer mit der Ware umgehen. In erster Instanz ist es aber der Shipper, der überhaupt festlegen können soll, welche Informationen von den Logistikern eingesehen werden können. Da der Spediteur für den Shipper die Transportkette organisiert, ist es der Spediteur, der – stellvertretend für den Shipper – die Richtlinien für den Sensordatenzugriff der einzelnen Parteien bestimmen soll.

Es ist also ein Zugriffskontrollsystem für den MASC-ID notwendig. Gemäß Abschnitt 3.1 ist der MASC-ID PDP, PEP und PIP in einem. Der PAP soll aber dezentral realisiert werden. Daher erhält der organisierende Spediteur nach der Transportregistrierung die Aufgabe, die Richtlinienerstellung zu übernehmen. Er muss also Shipper und Versicherer die benötigten Rechte zuweisen. Sind mehr als ein Shipper in einem Transport involviert und dementsprechend auch mehrere Versicherer, muss der Zugriff so definiert werden, dass die Parteien die Zugriffsrechte nur für den ihren Transportbeteiligungszeitbereich erhalten. Des Weiteren sollte der Spediteur verschiedenen Shippern verschiedene Sensorsichtbarkeiten als Dienst anbieten. Shipper *A* hat nur die Daten des Feuchtigkeitssensors gebucht und Shipper *B* interessiert nur die Türöffnung. Auf diese Weise kann der Spediteur den zugreifenden Parteien genaue Richtlinien zuweisen.

Sinngemäß betrifft die Richtlinienvergabe auch die Frachtführer und Unterspeditionen. Bei Auftragsvergabe, den Containertransport oder deren Organisation für eine Teilstrecke zu übernehmen, weist der Spediteur seinen Vertragspartnern Richtlinien zu, die ihnen erlauben, während ihres Teiltransportes bestimmte Sensordaten einzusehen. Ferner sollten die Richtlinien-zugewiesenen Logistiker in der Lage sein, die erhaltenen Rechte weitergeben zu dürfen. Wenn ein Unterspediteur weitere Frachtführer oder Unterspediteure mit Teilen seines Teiltransportes betraut, dann muss dieser die Möglichkeit haben, Richtlinien für die neuen Transportkettenteilnehmer auszustellen. Bei Fortführung dieses Verfahrens werden Stufe für Stufe weitere Parteien mit der Richtlinienerstellung betraut, wobei die zu erstellenden Richtlinien immer nur Teilmengen der eigenen Befugnisse erlauben.

Es ist also notwendig, im Rahmen des MASC-ID den PAP auf mehre Schultern zu verteilen und ihn somit zu dezentralisieren. Dieses ermöglicht den Parteien in der Transportkette die eigenen Zugriffsrechte an andere Parteien weiterzugeben, um so die Transportkette gänzlich erfassen und die Entscheidungen über die (logischen) Richtlinien an der Stelle treffen zu können, an der sie letztendlich auch durch (reale) Vertragsabschlüsse festgelegt werden.

5.1.2 Automatische Transportdokumentierung

Im Transportwesen ist eine Reihe von standardisierten Dokumenten anzufertigen. Besonders bei Grenzübertritten benötigen Zollbehörden Export- und Importpapiere. Jeder Frachtführer benötigt daher eine BoL vom Shipper und muss eigene Dokumente für den Transport erstellen. Bei der USA-Einfuhr muss dieses Dokument sogar die gesamte Transportkette enthalten.

Durch die verteilte Richtlinienerstellung können alle Teilnehmer der Transportkette auf sie zugeschnittene Zugriffsrechte auf die Sensordaten erhalten. Der PIP erhält dadurch einen Großteil des Wissens über die Transportkette: Die Richtlinien beschreiben den Zuständigkeitsbereich der einzelnen Transportbeteiligten und ergeben zusammen ein geschlossenes Bild der Transportkette. Nur wenige Zusatzinformationen wären nötig, um ein automatisiertes Erstellen der benötigten Transportdokumente vom MASC-ID zu realisieren.

Die zusätzlich benötigten logistischen Daten können ebenfalls in der Transportdatenbank hinterlegt werden. Der MASC-ID verwaltet diese Tabellen und die dezentrale Richtlinienerstellung (PAP) kann neben den Zugriffsrechten für die Sensordaten ebenfalls die für die Transportdokumentation erstellen. Auf diese Weise entsteht ein weiterer Mehrwert durch eine kleine Erweiterung des MASC-ID.

5.2 Delegierung

Die Besonderheit des MASC-ID ist die dezentral organisierte Richtlinienerstellung. Eine zentrale Zugriffsverwaltung wäre nur mit viel Kommunikation zu führen. Bei jedem neuen Vertrag müssten beide Vertragspartner umgehend mit dem System kommunizieren und ein Informationsaustausch zwischen Unterauftragnehmern und der zentralen Stelle wäre nötig. Der Forderung nach der leichten Integrierbarkeit in aktuelle Abläufe könnte mit einem zentralisierten System nicht nachgekommen werden. Deswegen wird im MASC-ID der PAP auf die Transportbeteiligten verteilt, der den Vertragsabschlüssen folgt.

Definition 5.3 (Delegieren)**.** *Bei der Delegierung übergibt eine Partei eigene Verantwortlichkeit für einen Teil ihrer Aufgaben an eine andere Partei.*

Die Definition lässt offen, ob die Verantwortlichkeiten von der delegierenden Partei behalten werden oder abgegeben werden. Dieses ist von der Delegierung unabhängig und kann beides beinhalten. Für diese Arbeit behält der Delegierende seine Verantwortlichkeiten bis diese widerrufen werden oder nicht mehr gültig sind.

Im Zusammenhang mit Zugriffskontrolle wird bei der Delegierung eine Autorisierung weitergegeben, indem der Delegierende eine Richtlinie anlegt, die dem Begünstigten das entsprechende Recht zuweist. Dadurch ist es möglich, die Richtlinienerstellung durch Delegierung dezentral zu organisieren. Anwendungsbezogen erhält der Spediteur initial die Aufgabe, PAP zu sein. In weiteren Schritten kann er somit Richtlinien erstellen und die Richtlinienerstellung für Teilbereiche der Transportkette an Unterspediteure delegieren. Unter Betrachtung des Vertragsbaumes (siehe 1.1.3) und der Tatsache, dass die Parteien nur ihre Vertragspartner kennen, kann dennoch vom organisierenden Spediteur jedem Knoten des Baumes durch mehrfache (transitive) Delegierung eine Erlaubnis zugeweisen werden.

In dieser Arbeit werden zwei verschiedene Zugriffssysteme betrachtet, die Delegierung ermöglichen. Zum einen existiert unter dem Namen *SPKI/SDSI* eine einfache PKI, die Delegierung von Rechten erlaubt. Zum anderen gibt es *AMANDA*, welches eine zusätzliche Kontrollinstanz einführt, um so Richtliniendelegierung überwachen zu können. AMANDA ist damit etwas serverbasierter und nicht ganz so dezentral wie SPKI/SDSI, erlaubt es aber Delegierungsbeschränkungen zu erstellen. Inzwischen ist AMANDA in *XACML*[1] *delegation* [64] überführt worden. Der folgende Abschnitt erläutert diese Systeme.

5.3 Technologien zur Delegierung

In diesem Abschnitt werden die beiden Delegierungstechnologien SPKI/SDSI und AMANDA erklärt. Zum besseren Verständnis werden fiktive Beispiele in Anhang C behandelt, die immer wieder referenziert werden. SPKI/SDSI verwendet im Vergleich zu anderen PKIs gleiche Begriffe, verbindet damit aber andere Prozesse. AMANDA ist wiederum sehr vergleichbar zu SPKI/SDSI, verwendet aber seinerseits (bewusst) eine andere Terminologie. Die beigefügten Beispiele sollen helfen, Komplikationen in der Verständlichkeit nicht aufkommen zu lassen. Im nachfolgenden Abschnitt werden diese beiden Technologien in die MASC-Welt überführt. Dies geschieht aus dem Grund, dass anhand des MASC-Beispiels nur schlecht die Gesamtkonzeption der jeweiligen Systeme deutlich gemacht werden kann, da die MASC-Anwendung die Möglichkeiten dieser Systeme bei weitem nicht ausnutzt. Allerdings ist das grobe Verständnis der gesamten Anwendung für das Verstehen der Konzepte erforderlich.

Das **Delegieren** wird von beiden Projekten ebenfalls unterschiedlich definiert. Bei SPKI/SDSI erhält ein Teilnehmer ein Zertifikat, in dem Zugriffsrechte für den Teilnehmer definiert sind. Wenn ihm das Recht zum Delegieren gegeben ist, darf er ihm zustehenden Zugriffsrechte an andere weitergeben. Die Delegierung bezieht sich bei SPKI/SDSI also nur auf Zugriffsrechte, die selber besessen werden und die durch Delegierung auch nicht verloren gehen, sondern dupliziert und weitergegeben werden. Bei AMANDA ist es darüber hinaus möglich Zugriffsrechte an andere zu delegieren, ohne dass sie selbst besessen werden. Es wird also nicht die Zugriffsrecht delegiert, sondern das Recht die Richtlinie dafür anzulegen. Wie die folgenden Abschnitte zeigen, sind die Unterschiede der Delegierung zwar

[1] XACML steht für eXtensible Access Control Markup Language, siehe B.3

konzeptioneller Art (Rechtedelegierung versus Richtlinienerstellungsdelegierung), von der Wirkung unterscheiden sie sich aber kaum.

5.3.1 SPKI/SDSI

Bei SPKI/SDSI[2] handelt es sich um eine Zusammenführung zweier Konzepte. Carl Ellison, Bill Frantz, Brian Thomas und Tatu Ylönen entwickelten die „Simple Public Key Infrastructur (SPKI)“ und parallel dazu Butler Lampson und Ronald Rivest „Simple Distributed Security Infrastructure (SDSI)“. SPKI/SDSI wurde 1999 in zwei RFCs überführt, die heute den Quasistandard für SPKI/SDSI bilden [29, 30]. Diese Arbeit verwendet folgend SPKI und SPKI/SDSI als Synonyme, da die experimentellen Ursprünge der beiden Einzelprojekte keine Rolle mehr spielen. Einzig sei noch erwähnt, dass das Hauptaugenmerk von SDSI auf der lokalen Namensstruktur lag, während Ellisons SPKI die Autorisierungsverwaltung in das gemeinsame Projekt einbrachte.

SPKI kennt ausschließlich (öffentliche) Schlüssel als Subjekte. Es ist dem System egal, ob dieser Schlüssel einem Teilnehmer, einem Prozess oder irgendeiner virtuellen Entität zugeordnet ist. Für diese Arbeit sind aber nur parteienbezogene Schlüssel relevant, weswegen im Folgenden ein öffentlicher Schlüssel immer einem Teilnehmer zugeordnet ist. Es gilt die Annahme, dass nur dieser Teilnehmer den privaten Schlüssel kennt, der zu dem öffentlichen Schlüssel gehört. Nachfolgend wird der Besitz des öffentlichen Schlüssels nur dem Teilnehmer zugerechnet, der auch Inhaber des privaten Schlüssels ist. Nur der Besitzer kann letzteren zum Signieren und Authentisieren verwenden. Jeder, dem der öffentliche Schlüssel bekannt gemacht wird, kann ihn ausschließlich zur Signaturverifizierung und Besitzerauthentifizierung verwenden, ist aber nicht dessen Besitzer.

Zertifikatstypen

SPKI betrachtet sich selbst als PKI. Allerdings ist es umstritten und hängt von der Definition von PKI ab, ob SPKI wirklich in diese Kategorie fällt. Bei klassischen PKIs weist sich ein Teilnehmer einem vertrauenswürdigen Dritten gegenüber aus, der ihm ein unterzeichnetes Zertifikat zurückgibt, welches bestätigt, dass diesem Teilnehmer der enthaltene öffentliche Schlüssel gehört. Der Dritte überprüft demnach Identität (meist über einen Personalausweis oder Postident-Verfahren) und Schlüsselbesitz und zertifiziert die Zuordnung. Die Identitäten, wie sie von klassischen PKIs verwendet werden, sind also Identifizierungen, die auf (behördlicher) Überprüfbarkeit (zum Beispiel Personalausweis, Firmenausweis) oder bei kleineren Gruppen auch auf Vertrauen basieren. Bei SPKI ist das einzige Identifizierungsmerkmal der öffentliche Schlüssel. Ein Teilnehmer weist nach, dass er diesen besitzt, indem er seine Anfragen mit seinem privaten Schlüssel unterzeichnet. Lassen diese sich mit dem öffentlichen Schlüssel verifizieren, können diese nur vom Besitzer des öffentlichen Schlüssels stammen.

[2] SPKI wird wie das englische „spookey“ und SDSI wie „sadsi“ ausgesprochen - eine sehr gute deutsche Einführung ist in [88] zu finden.

Bei SPKI handelt es sich bei einem Zertifikat nicht um einen Nachweis der erfolgreichen Identitätsprüfung. SPKI verwendet stattdessen Attributzertifikate und knüpft auf diese Weise Zugriffsrechte (beziehungsweise Autorisierungen) oder selbst gewählte Namen an (den Besitzer eines) öffentliche(n) Schlüssel(s). SPKI unterscheidet dementsprechend zwei Zertifikatstypen.

Definition 5.4 (Autorisierung)**.** *Wegen der Trennung von Ressource und Richtlinienerstellung ist es notwendig, dass dezentrale Systeme deutlich machen, für welches Objekt ein Zugriffsrecht gilt. Eine Autorisierung ist das Paar bestehend aus einem Objekt (Ressource) und dem dafür geltenden Zugriffsrecht.*

- Bei **Autorisierungszertifikaten** handelt es sich um eine Bindung eines öffentlichen Schlüssels oder eines Namens an eine Autorisierung. Ein Autorisierungszertifikat besteht aus fünf Informationen: *Aussteller* des Zertifikats, *Delegationsbit*, welches angibt, ob die enthaltenen Zugriffsrechte von dem Zertifikatsbegünstigten an andere Teilnehmer weitergegeben werden dürfen, *Subjekt* (kann ein öffentlicher Schlüssel oder ein Name sein), über den der Begünstigte identifiziert wird, *Autorisierung* also die Bindung eines Zugriffsrechts an eine Ressource, und *Gültigkeitsdauer* des Zertifikats. Zusätzlich existiert ein Kommentarfeld, das aber nicht für die Verarbeitung herangezogen wird, sondern lediglich zur Erhöhung der menschlichen Lesbarkeit optional verwendbar ist.
- Bei **Namenszertifikaten** handelt es sich um die Bindung eines öffentlichen Schlüssels an einen Namen. Mit diesem Zertifikat kann ein lokaler Namensraum aufgebaut werden. Ein Namenszertifikat besteht aus vier Informationen: *Aussteller*, *Name* der für das Subjekt gelten soll, *Subjekt*, also der öffentliche Schlüssel und *Gültigkeitsdauer* des Zertifikats. Auch hier gibt es ein Kommentarfeld, welches ebenfalls nicht bei der maschinellen Verarbeitung herangezogen wird. Mit diesem Zertifikat kann ein Benutzer seinen Namen bzgl. eines Namensraums einem seiner öffentlichen Schlüssel zuweisen.

AUSSTELLER	`sha1 |l4ISqKZdgmMIRCMWUibwuBkAucA=|`
SUBJEKT	`sha1 |JxVrP5ujvSdpo6cokRFbMURRDdo=|`
AUTORISIERUNG	*Zugriffsrecht(Objekt)*
DELEGIERUNG	`true`
GÜLTIGKEIT	(2007-01-10_23:59:59 , 2008-01-10_23:59:59)

Tabelle 5.1: Beispiel eines Autorisierungszertifikates in SPKI [2]

Tabelle 5.1 zeigt ein Beispiel für ein Autorisierungszertifikat. Der Aussteller wird über den Hash seines öffentlichen Schlüssels in diesem Zertifikat identifiziert. Das Subjekt zeigt den Hash des öffentlichen Schlüssels des Begünstigten. Mit einem Autorisierungszertifikat bindet der Aussteller ein eigenes Zugriffsrecht an einen anderen Teilnehmer und delegiert ihm dieses Recht. Bei der Zugriffsüberprüfung wird das Zertifikat auf Gültigkeit überprüft und wird zur Richtlinie. Aus Speicherplatzgründen werden in Zertifikaten lediglich die Hashwerte der Schlüssel gespeichert. Wenn im Folgenden öffentliche

Schlüssel in Zertifikaten vorkommen, handelt es sich immer nur um dessen Hashwert, aus praktischen Gründen wird diese Genauigkeit (o.B.d.A.) nicht weiter berücksichtigt.

AUSSTELLER	`sha1 |l4ISqKZdgmMIRCMWUibwuBkAucA=|`
SUBJEKT	`sha1 |JxVrP5ujvSdpo6cokRFbMURRDdo=|`
NAME	*Lokaler Name*
GÜLTIGKEIT	(2007-01-10_23:59:59 , 2008-01-10_23:59:59)

Tabelle 5.2: Beispiel eines Namenszertifikates in SPKI

Namenszertifikate sind sinngemäß strukturiert (siehe Tabelle 5.2). Ihre Aufgabe ist es einem Teilnehmer einen Namen zuzuordnen. Anstelle der Autorisierung enthalten sie demnach den gewählten Namen und das Delegierungsbit existiert nicht. Namen existieren ausschließlich bezüglich eines Ausstellers.

Namensraum

Jeder Teilnehmer kann sich einen Namensraum aufbauen, in dem er willkürliche Namen vergibt. Die Zuordnung eines Namens geschieht über ein Namenszertifikat. Auf diese Weise wird dem Subjekt (Besitzer des öffentlichen Schlüssels) des Zertifikats ein Name zugewiesen. Existiert ein Namensraum, kann ein Autorisierungszertifikat anstelle des Schlüssels auch Namen des Namensraumes tragen (siehe Tabelle 5.3).

AUSSTELLER	`sha1 |l4ISqKZdgmMIRCMWUibwuBkAucA=|`
SUBJEKT	**Lokaler Name**
AUTORISIERUNG	*Zugriffsrecht(Objekt)*
DELEGIERUNG	`true`
GÜLTIGKEIT	(2007-01-10_23:59:59 , 2008-01-10_23:59:59)

Tabelle 5.3: Autorisierungszertifikat mit Namen

Gruppen entstehen in dem ein Aussteller mehreren Begünstigten den gleichen Namen gibt, sprich in dem er mehrere Namenszertifikate ausstellt, die denselben Namen verschiedenen Subjekten (Teilnehmern) zuordnen. Alle diese Teilnehmer können nachfolgend mit dem Namenszertifikat nachweisen, dass ihnen der Name zugeordnet ist. Auf diese Weise können über Autorisierungszertifikate mit Namen ganzen Gruppen Zugriffsrechte zugeordnet werden.

Namenszertifikate dienen lediglich der Vereinfachung der Funktionsweise von SPKI und erleichtern das Bilden von Gruppen und Zuordnungen. Für Autorisierungen ist es unwichtig, ob sie auf Namen zugewiesen werden, die dann über Namenszertifikate auf Schlüssel abgebildet werden, oder direkt auf die Schlüssel abgebildet werden. Da für diese Arbeit lediglich die Richtlinienerstellung delegiert werden soll, werden Namenszertifikate nicht weiter berücksichtigt, da sie keine weitere Funktionalität bringen.

Richtlinienerstellung mit SPKI

Mit Hilfe von Autorisierungszertifikaten können bei SPKI Autorisierungen delegiert werden. Damit nicht beliebige Teilnehmer das Recht delegieren zu dürfen für sich in Anspruch nehmen können, muss es initiale Richtlinien geben, denen das System per definitionem vertraut. Es sind dann schließlich nur Delegierungen gültig, die von diesen Initialen abgeleitet sind. SPKI nennt die Liste dieser Initialeinträge SPKI-ACL, wobei das ACL für *Access Control List* steht. Die SPKI-ACL ist eine Richtlinientabelle, die Teilnehmern Autorisierungen zuweist.

SUBJEKT	`sha1 |JxVrP5ujvSdpo6cokRFbMURRDdo=|`
AUTORISIERUNG	*Zugriffsrecht(Objekt)*
DELEGIERUNG	`true`
GÜLTIGKEIT	(2007-01-10_23:59:59 , 2008-01-10_23:59:59)

Tabelle 5.4: Beispiel eines ACL-Eintrags in SPKI

Die Einträge in der SPKI-ACL (vgl. Tabelle 5.4) ähneln den Autorisierungszertifikaten, nur dass sie keinen Aussteller haben und nicht unterzeichnet und weitergegeben werden. Wegen der nicht enthaltenen Signatur, dürfen diese auch nicht als Zertifikat bezeichnet werden. Alle ACL-Einträge sind per definitionem gültig. Jeder beliebige Teilnehmer kann auf diese Weise durch initiale Einträge Autorisierungen zugewiesen bekommen. Wenn das Delegierungsbit gesetzt ist, dürfen die Begünstigten diese Zugriffsrechte anderen Teilnehmern weitergeben. Dies geschieht durch das Erstellen von Autorisierungszertifikaten.

Ein durch einen ACL-Eintrag begünstigter Teilnehmer darf ein Autorisierungszertifikat erstellen, wenn er bezüglich einer Autorisierung die Berechtigung zur **Delegierung** erhalten hat. Das Absenderfeld des Zertifikatausstellers muss dabei dem ACL-Eintrag entsprechen und die Signatur des Zertifikats muss mit dem privaten Schlüssel des Ausstellers erfolgt sein. So folgt das Zertifikat einem ACL-Eintrag. Auf gleiche Weise können Folgezertifikate erstellt werden, die dann alle eine Zertifikatskette bilden, die auf einen ACL-Eintrag zurückzuführen ist.

Ein erläuterndes Beispiel ist in Anhang C.1.1 angeführt, welches Delegierungsabläufe in SPKI funktional erläutert.

Zertifikatskettenüberprüfung

Wenn ein bestimmtes Zugriffsrecht mehrfach delegiert wurde, reicht beim Zugriff nicht die Vorlage des erhaltenen Zertifikats, um die Autorisierung zu erhalten. Es muss gleichzeitig nachgewiesen werden, dass der Zertifikatsaussteller delegierungsberechtigt ist. Dies gilt rekursiv für dessen Vorgänger. Daher muss immer eine konsistente Kette, bestehend aus allen Zertifikaten von ACL-Eintrag bis zum Zugreifenden, vorgelegt werden (siehe Beispiel im Anhang C.1.1).

Die Aufgabe des PDP läuft folgend ab: Er reduziert die Zertifikatskette zu einem **Ergebniszertifikat**, indem er die 5-Tupel-Reduktion anwendet. Dieser Algorithmus dient der Klärung, ob die Delegierungen korrekt durchgeführt worden sind. Schlussendlich muss sich ein ACL-Eintrag finden lassen, der sich auf gleiche Weise reduzieren lässt.

5-Tupel-Reduktion Damit der Überprüfer (PDP) bei einer Anfrage eine Erlaubnisentscheidung treffen kann, muss er zuerst auf die erhaltene Zertifikatssequenz die 5-Tupel-Reduktion anwenden, um die Kompatibilität der ausgestellten Zertifikate in der Kette zu überprüfen. Dabei werden schrittweise die ersten beiden Zertifikate zu einem resultierenden Zertifikat reduziert. Anschließend wird dieses resultierende Zertifikat mit dem nächsten (dritten) Zertifikat reduziert, und so weiter. Am Ende muss es ein ACL-Eintrag geben, mit dem sich das reduzierte Zertifikat zu einem Ergebniszertifikat zusammenfassen lässt. Es enthält bei Korrektheit den Aussteller „self“ und als Subjekt den Zugreifenden.

Ein Basiselement der Tupel-Reduktion ist die Schnittmengen-Relation $\cap$. Angewendet auf Autorisierungen und Gültigkeitsintervalle bildet es die resultierenden Eigenschaften bei der Reduktion. Zur korrekten Anwendung der **Schnittmenge** müssen die Autorisierungen und Gültigkeitsintervalle als Mengen aller enthaltenen kleinstmöglichen Elemente betrachtet werden. Bezüglich der Gültigkeitsintervalle ist dazu eine Diskretisierung des Zeitbereiches notwendig – was bereits durch die computergestützte Verarbeitung inhärent erfolgt.

Definition 5.5 (5-Tupel-Reduktion)**.** *Die zwei aufeinanderfolgenden zu reduzierenden Zertifikatsketteneleme seien Z_A und Z_B und bestehen aus fünf Elementen und einer Unterschrift. Die fünf Elemente werden zu einem Fünf-Tupel zusammengefasst und entsprechend T_A und T_B bezeichnet. ACL-Einträge sind bereits Tupel aus vier Elementen, sie werden um das Schlüsselwort „self“ an vorderer Stelle ergänzt und werden auf diese Weise auch zu 5-Tupeln.*

$$T_A = [A_A, S_A, E_A, D_A, G_A]$$
$$T_B = [A_B, S_B, E_B, D_B, G_B]$$

- *A sei das Tupelelement, welches für den Aussteller steht.*
- *S sei das Subjekt (der Begünstigte) des zu reduzierenden Zertifikats/ACL-Eintrags.*
- *E sei die Autorisierung (Erlaubnis), die das Zertifikat dem Subjekt zuordnet.*
- *D sei das Delegationsbit (*1 *bedeutet Delegierung erlaubt,* 0 *bedeutet Delegierung nicht erlaubt)*
- *G sei das Gültigkeitsintervall für die die Erlaubnisweitergabe gilt.*

Zwei Tupel lassen sich zu einem resultierenden Ergebnistupel T_R reduzieren, wenn folgende Bedingungen erfüllt sind.

- $S_A = A_B$: *Das Tupel welches dem nachfolgenden Zertifikat Z_B entspricht hat als Aussteller den Begünstigten des Zertifikats Z_A.*

- $D_A = 1$: *Der Begünstigte des vorhergehenden Zertifikats ist delegierungsberechtigt.*

Das reduzierte Tupel T_R wird wie folgt berechnet:

$$T_R = [A_A, S_B, \{E_A \cap E_B\}, D_B, \{G_A \cap G_B\}]$$

Das Tupel C_R wird durch Umwandlung in Zertifikatsstruktur und Anhängen einer digitalen Unterschrift zum Ergebniszertifikat Z_R.

Beim Zugriffsversuch übergibt der Zugreifende dem Wächter (PDP und PEP) die signierte Zertifikatskette und den Zugriffswunsch. Der PDP authentifiziert den Zugreifenden durch Signaturverifizierung und erstellt bei Erfolg durch Mehrfachanwenden der 5-Tupel-Reduktion ein Ergebniszertifikat. Wenn sich dieses berechnen lässt und schließlich als Aussteller „self“, als Subjekt den Zugreifenden, als Autorisierung eine Berechtigung enthält, die den Zugriffswunsch enthält und die zum Ausführungszeitpunkt gültig ist, dann gewährt der PEP den Zugriff. Die dieser Entscheidung zu Grunde liegende Zugriffsrichtlinie entspricht dem Ergebniszertifikat.

Die Reduktion von Namenszertifikaten verläuft sinngemäß mit der 4-Tupel-Reduktion ab.

Richtlinienverwaltung

Bei SPKI ist die Zugriffskontrolle zweigeteilt. Der Wächter eines Dienstes beherbergt PDP und PEP. Die Richtlinienerstellung wird auf die Teilnehmer verteilt und entspricht einem verteilten PAP. Der PIP existiert genau genommen nicht. Zur Richtlinienüberprüfung (5-Tupel-Reduktion) werden Einträge der zentralen SPKI-ACL und die im Rahmen des PAP generierten Zertifikatsketten ausgewertet. Die Richtlinien (Ergebniszertifikate) werden lediglich temporär berechnet und danach nicht weiter berücksichtigt. Es gibt also keinen Richtlinienspeicher sondern nur den Speicher des Vertrauensanker (SPKI-ACL), der als Basis für die Richtlinienberechnung dient. Dieser wird daher als PIP bezeichnet.

Die beschriebenen Vorgänge setzen voraus, dass alle Zertifikate vom Aussteller signiert sind und die Signaturen vom Wächter ebenfalls überprüft werden. Nur so kann die Echtheit und Fälschungssicherheit der Zertifikate gewährleistet werden.

Eine Optimierung von SPKI ist die Rückgabe der signierten Ergebniszertifikate vom Wächter an den Zugreifenden. Auf diese Weise muss der PDP bei späteren Zugriffen nicht mehr eingreifen und der PEP kann bereits bei Übergabe des Ergebniszertifikats entscheiden, ob der Zugriff erlaubt ist.

Widerruf

Die Delegierung von SPKI ist eine Erlaubnisweitergabe, die eigenen Zugriffsrechte anwenden zu dürfen. Ein Widerruf der eigenen Befugnisse muss sich also so auswirken, dass alle Delegierungen gleichermaßen unwirksam werden. Es gibt verschiedene Varianten, diesen Widerruf umzusetzen.

Das Widerrufen von Zertifikaten ist in SPKI nicht einheitlich geregelt. Problematisch ist die SPKI-inhärante unüberwachte Weitergabe von Autorisierungen und die damit verbundene Unmöglichkeit, die Delegierungen bei den Delegierten zu widerrufen. Es werden aber einige Varianten im Rahmen der SPKI-RFCs [29, 30] vorgeschlagen, die SPKI in verschiedenen Bereichen einschränken: Eine Variante sind die so genannten „Onlinetests“, bei denen ein Verifizierer per Internet eingereichte Zertifikate beim Herausgeber dieses Zertifikats direkt überprüfen kann. Dieses Modell ist sowohl für Autorisierungs- als auch für Namenszertifikate anwendbar. Dies zentralisiert jedoch die Anwendung und jeder Delegierende muss immer, wenn eine Delegierung stattfindet, einen Webdienst in Betrieb haben. Alternativ könnte eine zentrale Widerrufliste pro PEP eingerichtet werden. Die so bezeichneten CRLs („Certificate Revocation List“) widersprechen aber genau genommen dem dezentralen Ansatz vieler SPKI-Anwendungen. Weitere Widerrufstrategien sind denkbar, die Widerrufstrategie muss dabei dem Einzelfall angepasst werden.

SPKI-Zertifikate sind nicht eindeutig zu identifizieren, da sie keine Seriennummer enthalten. Wenn ein Aussteller die gleiche Autorisierung zweimal einem Begünstigten delegiert, ist es nicht möglich, nur eines der beiden zu widerrufen. Auf diese Weise können nur Zertifikatsmuster widerrufen werden. Enthält ein Widerruf Aussteller, Subjekt und Autorisierung, werden von allen Zertifikaten, die von dem Aussteller an das Subjekt gerichtet sind, die widerrufenen Rechte entzogen. Enthält ein Widerruf lediglich Aussteller und Subjekt, werden alle Zertifikate widerrufen, die von dem Austeller an das Subjekt gehen – auch zukünftige. Ebenfalls ist eine sinngemäße Beeinträchtigung des Gültigkeitsintervalls denkbar. Enthält ein Widerruf Autorisierung und Zeitintervall, wird nicht die Autorisierung widerrufen, sondern nur deren Gültigkeitsdauer zusätzlich eingeschränkt.

Wenn zur Optimierung Ergebniszertifikate verwendet werden, ist der Widerruf nur von Zugreifenden möglich. Ein Widerruf des Kettenelements kann nicht mehr erfolgen, da dieses im Ergebniszertifikat nicht mehr referenziert wird.

Schlussbemerkung

SPKI ist ein leichtgewichtiges und dezentrales Mittel, um die Weitergabe von Autorisierungen zu ermöglichen. Es hat jedoch Beschränkungen beim Widerruf von getroffener Erlaubnisentscheidungen und bei der Möglichkeit, Delegierungen zu kontrollieren. Jeder, der bei SPKI ein Delegierungsrecht besitzt, kann es unbeschränkt nutzen. Es mag allerdings wünschenswert sein, dass ein Teilnehmer einer anderen Partei eine Autorisierung erteilen möchte ohne das inherante Zugriffsrecht selbst zu besitzen. Dies ist mit SPKI genauso wenig möglich, wie das Beschränken von Delegierungen auf vorgebbare Teilnehmerkreise.

5.3.2 AMANDA

AMANDA steht für „Authorisation MANagement for Distributed Applications“ und ist ein Schema, um Zugriffsmanagement zu regeln. AMANDA wurde durch ein militärisches Projekt inspieriert, bei dem das Zusammenarbeiten von KFOR[3]-Truppen besser geregelt werden sollte und entstand in Kooperation mit *Microsoft Research*. Die nationalen Einheiten unterliegen dem eigenen Militärsystem, müssen aber vor Ort mit den anderen zusammenarbeiten. Dazu ist es notwendig, Aufgaben über Einheitsgrenzen zu delegieren, ohne dass das komplette Kommando über die Truppe abgegeben wird. Aus der „Constrained Delegation“-Anwendung entstand das C++-basierte Softwareprojekt „Delegant“. Schließlich wurde ein Patch für die XACML Version 3.0 herausgebracht, mit derer sich das Konzept „XACML delegation“ nennt. AMANDA basiert auf den drei Papieren [76, 4, 77].

Während bei SPKI Zertifikate nicht nur zur Rechteweitergabe, sondern auch zur Teilnehmerauthentifizierung verwendet werden, lässt AMANDA bewusst offen, wie der AMANDA-verwendende Dienst Teilnehmer identifiziert, authentifiziert und überhaupt verwaltet. Dabei wird nicht definiert, wie Subjekte und Aussteller von Zertifikaten aussehen; dieser Aufgabe wird einer unberücksichtigten Schicht überlassen. AMANDA beschränkt sich vielmehr auf die Richtlinienverwaltung und die dafür benötigte Logik als auf deren konkrete Umsetzung. AMANDA unterscheidet zwischen der Autorisierung und dem Recht die Autorisierung jemanden zuzuordnen. Es wird also nicht eine Autorisierung weitergegeben und delegiert, sondern die Berechtigung, eine Richtlinie für einen Anderen anlegen zu dürfen, der die Autorisierung erteilen darf. AMANDA verallgemeinert Autorisierungen und Richtlinienerstellungen zu *Privilegien*.

Privilegien

Bei AMANDA gibt es zwei verschiedene Privilegien. Die Autorisierung ist ein Privileg, das einem Teilnehmer erteilt werden kann. Dieses Privileg entspricht der SPKI-Autorisierung ohne Delegierungsrecht. Beim zweiten Privileg handelt es sich um die Befugnis, Privilegien erstellen zu dürfen (vgl. [76]):

- **Zugriffsprivileg** – Das Zugriffsprivileg ist gleichbedeutend mit der Autorisierung.
- **Managementprivileg** – Ein Managementprivileg erlaubt die Erstellung neuer Privilegien. Entweder kann ein weiteres Managementprivileg erstellt werden oder ein Zugriffsprivileg. Ein Zugriffsprivileg kann ausschließlich von einem Managementprivilegierten erteilt werden.

Ein Privileg bezieht sich immer auf eine spezielle Autorisierung (vergleichbar mit den Delegierungen bei SPKI). Das Managementprivileg zu einer Autorisierung erlaubt dem Privilegierten, als PAP für diese Autorisierung zu wirken. Nur wer ein Managementprivileg hat, kann einer Partei ein Zugriffsprivileg zuteilen. Er kann das Recht, PAP für diese Autorisierung zu sein, aber auch weiter delegieren.

Ein Beispiel zum besseren Verständnis der Privilegien in AMANDA ist in Anhang C.2.1 beigefügt.

[3] KFOR steht für Kosovo-Force und damit für die Militäreinheiten, die an der Kosovo-Befriedung teilnehmen.

Definition 5.6 (Teilnehmer)**.** *Ein Teilnehmer ist die Entität, die Privilegien erhalten und ausüben kann. Über die Menge aller Teilnehmer $\mathfrak{T}$ sei ein Alphabet definiert; jeder Teilnehmer ist ein Element dieses Alphabets.*

Für diese Arbeit werden AMANDA-Teilnehmer mit kleinen kursiven Buchstaben geschrieben (a, b, c). Das Alphabet ist dabei die Menge aller Teilnehmer.

Definition 5.7 (Delegierungskette)**.** *Eine Delegierungskette* D *sei eine geordnete Menge (Liste) von Teilnehmern und gilt nur für eine definierte Autorisierung.*

$$\mathsf{D} = \{\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3 \ldots \mathsf{d}_n\}\,,\ \forall_i \mathsf{d}_i \in \mathfrak{T}$$

Die Liste der Teilnehmer einer Delegierungskette korrespondiert mit den Ausstellern von Privilegiendelegierungen, die schlussendlich dazu führt, dass ein Teilnehmer eine Autorisierung erhält. Ausschließlich der Teilnehmer, der durch das letzte Element einer Delegierungskette beschrieben wird (d_n*), erhält die Autorisierung.*

Eine Delegierungskette, die beschreibt, dass a ein Managementprivileg an b und dieser ein Zugriffsprivileg an c ausgestellt hat, wird in der Form (abc) mit $a, b, c \in \mathfrak{T}$ geschrieben.

Als *unvollständige* Delegierungskette wird eine nicht bei einem zugriffsprivilegierten Teilnehmer endende Kette bezeichnet.

Zertifikatstypen

Das Zuteilen von Privilegien geschieht bei AMANDA über eine ausgelagerte Richtlinienerstellung. Die Zuweisung kann ebenfalls über Zertifikate durchgeführt werden. Wegen des besseren Verständnisses und der besseren Vergleichbarkeit mit SPKI wird dieser Ansatz im Folgenden gewählt. Das Zertifikat muss dabei nicht notwendigerweise unterschrieben sein, sondern ist lediglich ein Richtliniencontainer. Es ist ebenfalls nicht definiert wie die Zertifikate den PEP erreichen. Bei SPKI wurden sie zu einer Zertifikatskette durch Teilnehmerweitergabe zusammengefasst und vom Zugreifenden eingereicht. AMANDA lässt offen, ob die Zertifikate direkt (vom Aussteller) oder indirekt (vom Privilegierten) eingereicht werden.

Es werden zwei Zertifikatstypen unterschieden: Das dem SPKI-Autorisierungszertifikat entsprechende Zertifikat zur Ausübung und Delegierung von Privilegien wird folgend Privilegierungszertifikat genannt. Mit diesem werden anderen Teilnehmern Privilegien zugewiesen. Das andere ist ein Widerrufszertifikat, um Privilegierungszertifikate zu widerrufen.

Tabelle 5.5 stellt ein **Privilegierungszertifikat** (PZert) dar. Das Feld *Aussteller* enthält die Identifizierung des Privilegierten. AMANDA lässt offen, mit welchem Identifizierungsmerkmal sich ein Teilnehmer identifiziert. Es könnte wie bei SPKI der Hashwert seines öffentlichen Schlüssels sein. Das *Subjekt*

AUSSTELLER	Aussteller-Identifizierung
SUBJEKT	*Constraint*
AUTORISIERUNG	Autorisierung
GÜLTIGKEIT	(2007-01-10_23:59:59 , 2008-01-10_23:59:59)
SERIENNUMMER	*eindeutige ID*

Tabelle 5.5: Beispiel: Privilegierungszertifikat in AMANDA

ist nicht gleich dem Empfänger des Zertifikats, sondern eine qualifizierte Aussage, für welche Teilnehmer dieses Zertifikat gilt und wie die nachfolgende Delegierungskette auszusehen hat. Constraints (siehe unten) werden deshalb als Subjekte verwendet. Die *Autorisierung* entspricht dem gleichnamigen Feld des SPKI-Zertifikats. Gleiches gilt für die *Gültigkeit* mit der Ausnahme, dass es sich nicht auf den Zugriffszeitpunkt, sondern auf den der Delegierung bezieht. Es ist demnach unerheblich, ob die Delegierenden beim Zugriff delegierungsberechtigt sind. Lediglich die Berechtigungen zum Delegierungszeitpunkt sind entscheiden. Dies ist ein Unterschied zu SPKI. Schließlich enthält jedes PZert eine *Seriennummer*, über die es eindeutig zu identifizieren ist.

AUSSTELLER	Aussteller-Identifizierung
SERIENNUMMER	*ID eines Priviegierungszertifikats*
GÜLTIGKEIT	(2007-01-12_23:59:59 , 2008-01-10_23:59:59)

Tabelle 5.6: Beispiel: Widerrufszertifikat in AMANDA

Das **Widerrufszertifikat** (WZert) (siehe Tabelle 5.6) enthält lediglich die *Aussteller*-Identifizierung, die *Seriennummer* des zu widerrufenden Zertifikats und das zu widerrufende *Gültigkeitsintervall*.

Gruppenmitgliedschaften

Um die Privilegienvergabe nicht nur an einzelne Teilnehmer zu delegieren, sondern es ganzen Gruppen zu ermöglichen, ein Privileg ausüben zu können, ist es notwendig, dass Benutzer in Gruppen zusammengeschlossen werden können. Mit Hilfe von Gruppen ist es möglich, Bedingungen zu formulieren, die die Weitergabe von Privilegien nur einer bestimmten Teilnehmergruppe erlauben.

Es gibt dabei eine gewisse Parallele zu der Delegierung an einen Namen bei SPKI. Einem Namen können genau wie einer Gruppe mehrere Teilnehmer zugeordnet werden. Allerdings ist es durch *Constraints* möglich, nicht nur allen Teilnehmern einer Gruppe ein Recht zuzuweisen, sondern die erlaubten Kettenelemente einer Delegierungskette von Beginn an vorzuschreiben (vgl. [4]).

Definition 5.8 (Gruppen)**.** *Eine Gruppe* G *sei eine (ungeordnete) Menge von Teilnehmern. Über die Menge aller Gruppen* $\mathfrak{G}$ *sei ein Alphabet definiert; jede Gruppe ist eine Menge aus Teilnehmern.*

$$\mathsf{G} = \{\mathsf{g}_1, \mathsf{g}_2, \mathsf{g}_3, \ldots\}, \mathsf{G} \in \mathfrak{G}, \forall_i \mathsf{g}_i \in \mathfrak{T}$$

Im Folgenden werden Gruppen mit einem großen Buchstaben (A, B, C) geschrieben und deren Mitglieder mit einem kleinen $(a \in A, b \in B, c \in C)$. Gegebenenfalls werden verschiedene Gruppenmitglieder mit einem Index unterschieden $(c_1, c_2, c_3 \in C)$.

Constraints

Die wesentliche Charakteristik von AMANDA ist die Beschränkung der Delegierung auf einen vorgeschriebenen Rahmen. So ist es möglich zu bestimmen, dass eine Autorisierung nur von Mitgliedern einer bestimmten Gruppe ausgeübt werden kann und es nur Mitgliedern einer anderen Gruppe erlaubt ist, die Richtlinie dafür zu erstellen. Ein Constraint beschreibt den Rahmen der erlaubten Delegierungsketten. Durch Constraints können Delegierungen auf Gruppen beschränkt werden.

Definition 5.9 (Constraint)**.** *Ein Constraint ist eine geordnete Menge (Liste) von Gruppen. Ein Constraint schreibt die erlaubten Delegierungsketten vor: Jedes für eine Gruppe stehende Element eines Constraints muss listenpositionsgetreu auf ein Teilnehmer-korrespondierendes Element der Delegierungskette treffen, wobei der Teilnehmer ein Gruppenmitglied sein muss, damit die Delegierungskette gültig ist.*

$\mathfrak{C}$ *sei die Menge aller möglichen Constraints.*

$\mathfrak{D}$ *sei die Menge aller möglichen Delegierungsketten.*

Eine Delegierungskette $\mathsf{D} = \{\mathsf{d_1 d_2 d_3} \ldots\}$ *bezüglich einer Autorisierung* α *ist nur zugelassen, wenn es ein Richtlinie (Zertifikat) gibt, die* α *an das Constraint* $\mathsf{C} = \{\mathsf{C_1 C_2 C_3} \ldots\}$ *bindet, so dass gilt:*

$$\forall_i : \ \mathsf{d_i} \in \mathsf{C_i} \ \ \textit{mit}\ \mathsf{C} \in \mathfrak{C},\ \mathsf{D} \in \mathfrak{D}, \mathsf{C_i} \in \mathfrak{G},\ \mathsf{d_i} \in \mathfrak{T}$$

Ein Constraint schreibt alle möglichen Delegierungsketten vor, die bezüglich einer Autorisierung erlaubt sind. Dargestellt wird der Constraint durch eine Liste der erlaubten Gruppen beziehungsweise Teilnehmer. Das erste Delegierungskettenelement muss Mitglied des ersten Constraintelements sein. Dieser muss also das Managementprivileg für die Autorisierung besitzen (außer, es gibt keinen Nachfolger in der Delegierungskette) – er darf also Privilegien delegieren. Die Position des von ihm Begünstigten in der Delegierungskette schreibt vor, ob er ein Managementprivileg oder ein Zugriffsprivileg delegieren darf. Da die Delegierungskette eine elementweise Teilmenge des Constraints sein muss, kann der Constraint auf diese Weise vorschreiben, welche Privilegien welchen Teilnehmern ausgestellt werden dürfen. Durch die Verwendung der Gruppen können die Positionen möglicher Teilnehmer genau vorgeschrieben werden. Enthält ein Constraint eine Gruppe als letztes Element, dürfen nur Mitglieder dieser Gruppe durch Delegierungsketten die Autorisierung erhalten. In Beispiel C.3 wird eine einfache Delegierungskette beschrieben, die einem Constraint folgt.

Zur noch genaueren Bestimmung der Constraints, können sie auch aus regulären Ausdrücken bestehen. Dass gerade reguläre Ausdrücke verwendet werden, hat zwei Gründe: Zum einen gibt es für die

algorithmische Auswertung regulärer Ausdrücke eine Vielzahl von Interpretern, die im Anwendungsfall eingesetzt werden können. Zum anderen waren bei sämtlichen erdachten Anwendungen, für die AMANDA entwickelt wurde, alle Constraints durch reguläre Ausdrücke abbildbar.

Definition 5.10 (Erweiterte Constraints)**.** *Die geordnete Menge (Liste) der Gruppen kann durch reguläre Ausdrücke erweitert werden. Die Auswertung des regulären Ausdrucks führt zu gültigen Constraints. Ein erweiterter Constraint kann also als Menge von (einfachen) Constraints gesehen werden, wobei eine Delegierungskette nur einem Constraint der Menge genügen muss. Die Mengenelemente sind dabei alle durch Auswertungen des regulären Ausdrucks zu erreichenden Constraints.*

Zwei Beispiele für reguläre Ausdrücke seien erwähnt: Der Kleene-Stern hinter einem Gruppensymbol ($A^*, A \in \mathfrak{G}$) besagt, dass beliebig viele Gruppeninkarnationen der Gruppe hintereinander auftreten dürfen. Für die Delegierungskette bedeutet das, dass Gruppenmitglieder mehrfach hintereinander vorkommen. Ebenfalls das keinmalige Auftreten des Gruppenelementes ist erlaubt. $A^{\{0,1\}}, A \in \mathfrak{G}$ besagt, dass das Gruppensymbol im resultierenden Constraint genau einmal oder keinmal vorkommen darf. Beispiel C.4 zeigt die Verwendung des Kleene-Sterns.

Bei Verwendung dieser *erweiterten Constraints*, entspricht das letzte Constraintelement nicht zwangsläufig dem letzten Element der Delegierungskette. Dementsprechend erhält nicht unbedingt ein Repräsentant des letzten Constraintelements die Autorisierung.

Definition 5.11 (Bedingter Constraint)**.** *Wird bei einem Constraint einem Gruppensymbol eine Bedingung in einer Klammer nachgestellt, handelt es sich um einen bedingten Constraint. Die Bedingung muss im Moment der Auswertung des Constraints durch den PDP gültig sein.*

Während erweiterte Constraints die Anzahl der abbildbaren Delegierungsketten erhöhen, schränken *bedingte Constraints* diese ein, indem zum Zeitpunkt der Privilegienausübung eine weitere Bedingung vorschreiben, die erfüllt sein muss. Das bedingte Constraintelement $A(Bedingung)BC,\ A,B,C \in \mathfrak{G}$ erlaubt einem Gruppenmitglied von A die Zuweisung einer Autorisierung an ein Gruppenmitglied von B nur, wenn zum Zeitpunkt der Privilegienausübung die Bedingung erfüllt ist. Es ist natürlich auch möglich, die Bedingung auf die Ausübung der Autorisierung zu beziehen. In diesem Fall müsste das letzte Constraintelement mit dieser Bedingung versehen werden. Eine Bedingung am Ende einer Kette ($ABC(Bedingung),\ A,B,C \in \mathfrak{G}$) bezieht sich also auf den Ressourcenzugriff, während eine Bedingung an einer anderen Stelle ($AB(Bedingung)C,\ A,B,C \in \mathfrak{G}$) zum Delegierungszeitpunkt erfüllt sein muss.

Die Constraints geben lediglich den äußeren Rahmen des Delegierungsbaums vor, der konkrete Anwendungsfall (Delegierungskette) muss nur in diesem Rahmen bleiben. Es ist im Übrigen auch möglich, bei jedem Delegierungsschritt die Constraints weiter einzuschränken. Sowohl bei der Autorisierung als auch bei den Gruppen sind feinere Einschränkungen erlaubt. Es muss nur gelten, dass die neue Autorisierung eine Teilmenge (gemäß Definition 5.12) der alten ist und die Menge der mit dem neuen Constraint erzeugbaren Delegierungsketten eine Teilmenge der mit dem alten Constraint erzeugbaren.

Im Anhang C.2.3 ist ein ausführliches Beispielszenario beschrieben, wie Delegierungsbeschränkungen effektiv mit AMANDA durchgeführt werden können.

Delegierungsdurchführung

AMANDA gibt dem Nutzer die Möglichkeit, Privilegien an ganze Gruppen zu verteilen. Diese Eigenschaft in der Praxis umzusetzen, ist jedoch nicht in jedem Anwendungsfall möglich. Da AMANDA offen lässt, wie die Kommunikation zwischen den Teilnehmer vonstatten geht, liegt dies also an der konkreten Umsetzung der Gesamtanwendung.

Bei SPKI dienen Zertifikate nicht nur der Weitergabe der Rechte, sondern auch der Benachrichtigung des Begünstigten, dass der Aussteller ihm dieses Recht zugeordnet hat. Wenn AMANDA auf dem gleichen Weg benutzt werden soll, wäre ein Multicastsendeverfahren notwendig, um alle Begünstigten von einer Gruppendelegierung zu unterrichten.

Anders sieht es aus, wenn AMANDA nur einem zentralen PIP die neuen Richtlinien mitteilt. Hier können sehr wohl Gruppendelegierungen vergeben werden. Die Mitglieder der berechtigten Gruppen müssten nachfolgend über einen anderen Weg von ihren Privilegien in Kenntnis gesetzt werden. Das Dienstzugriffsprogramm könnte dies zum Beispiel übernehmen und bei der Systemanmeldung alle Berechtigungen abfragen und dem Teilnehmer anzeigen.

Aus diesem Grund müssen bei AMANDA zwei Anwendungsfälle unterschieden werden, die AMANDA verschiedene Entfaltungsmöglichkeiten geben. Es geht dabei um die Art und Weise der Erstellung von Richtlinien. Bei der *Zentralen Delegierung* muss bei jeder neuen Privilegierung ein zentraler PDP instruiert werden, eine neue Richtlinie anzulegen. Bei der *Verteilten Delegierung* wird das SPKI-Modell verwendet, in dem Zertifikatsketten gebildet werden, die allerdings spätestens unmittelbar vor dem Zugriff der PDP übergeben werden müssen.

Zentrale Delegierung Die zentrale Delegierung ist das klassische AMANDA-Modell. Kernstück dieses Verfahrens ist ein zentraler PDP, der exklusiven Zugriff auf den PIP hält. Der Aussteller eines Zertifikats sendet sein Zertifikat nicht an den Begünstigten der Delegierung, sondern an diesen PDP. Dieser übernimmt die unmittelbare Überprüfung des Zertifikats. Im Besonderen muss der PDP sicherstellen, dass der Zertifikatsaussteller berechtigt ist, die Autorisierung an das Subjekt zu delegieren. Ist das Zertifikat korrekt, wird dieses in die Richtliniendatenbank übernommen.

Verteilte Delegierung Bei der Verteilten Delegierung ist es erlaubt, dass Zertifikatsketten bei dem PDP eingereicht werden. Somit ist nicht jeder Delegierungsschritt mit dem Server rückgekoppelt, sondern es können ohne sein Wissen SPKI-äquivalente Privilegierungsketten gebildet werden. Diese können zu jedem beliebigen Zeitpunkt bei dem Verifizierer eingereicht werden – auch unmittelbar vor dem Zugriffswunsch. Im Gegensatz zu SPKI, bei dem die Zertifikatsketten bei jedem Zugriff zu übergeben sind, muss bei AMANDA lediglich dafür gesorgt werden, dass sie spätestens vor dem ersten Zugriff zur

Verfügung stehen. Die Zertifikate werden zu gültigen Richtlinien, wenn der PDP nach deren Kontrolle diese dem PIP zur Speicherung übergibt. Beim nachfolgenden Zugriff wendet sich ein Teilnehmer an den PEP und dieser ermittelt (nach Rücksprache mit dem PDP) ob der Zugriff erlaubt ist. PEP und PDP sind (im Gegensatz zu SPKI) konzeptionell getrennt.

Der Delegierungsvorgang unterscheidet sich bei den beiden Modellen. Während bei der Zentralen Delegierung jedes neue Privileg dem PDP direkt mitgeteilt wird, werden bei der Verteilten Delegierung Zertifikatsketten gebildet und diese in einem eingereicht. Allerdings können auch Zertifikatsketten eingereicht werden die unvollständigen Delegierungsketten repräsentieren oder nur einzelne Zertifikate. Es ist (anders als bei SPKI) nicht vorgeschrieben, dass eine Zertifikatskette immer die gesamte Delegierungskette abzubilden hat – eigentlich ist die Verteilte Delegierung nur die Zentrale Delegierung mit der Möglichkeit einer späteren (und gesammelten) Zertifikatsüberprüfung.

Die Verteilte Delegierung schränkt AMANDA in zwei Weisen ein. Zum einen können keine Gruppen von einer Delegierung profitieren[4], jedes Gruppenelement müsste einzeln ein PZert zugesendet bekommen. Zum anderen kann der Delegierungszeitpunkt nicht mehr festgestellt werden. Der PDP kann nicht zwischen Erstellungs- und Einreichungszeitpunkt unterscheiden: er kann die Gültigkeitsüberprüfung also erst zu dem Zeitpunkt durchführen, bei dem das Zertifikat eingereicht wird. Die Constraints und Gültigkeiten beziehen sich dadurch nicht mehr auf den Privilegierungszeitpunkt, sondern auf den verlagerten Einreichungszeitpunkt. Auf diese Weise kann es sein, dass ein Teilnehmer nicht mehr berechtigt ist, die Richtlinie zu erstellen, obwohl er es zum Delegierungszeitpunkt war und er bei Zentraler Delegierung berechtigt gewesen wäre.

Richtlinienüberprüfung

Unabhängig von der Wahl des Delegierungsmodells läuft die Überprüfung der Zertifikate ab. In einem Fall wird zwar jedes Zertifikat einzeln eingereicht und im anderen ganze Zertifikatsketten, aber die Ketten werden elementweise in Erstellungsreihenfolge verarbeitet, so dass schließlich die gleiche Überprüfungsprozedur für jedes Zertifikat verwendet wird. Nachfolgender Überprüfungsalgorithmus basiert auf [4].

Definition 5.12 ($\leq$-Relation)**.** *Die $\leq$-Relation bezieht sich auf die Restriktivität der Einschränkung. Eine Autorisierung oder ein Constraint welches $<$ als ein anderes ist, ist eine Verfeinerung der Beschränkung. Beim Constraint entspricht das $\leq$ einer Teilmengen -Relation bezüglich der Anzahl der vom Constraint ermöglichten Delegierungsketten. Für eine Autorisierung bedeutet $\leq$, dass höchstens die gleichen effektiven Berechtigungen enthalten sind.*

Definition 5.13 (Zertifikat)**.** *Das Zertifikat (PZErt) ist ein Richtliniencontainer, der gemäß vorheriger Beschreibungen ein Tupel aus vier Elementen ist: Aussteller p, Subjekt (Constraint) S, Autorisierung α und Einreichungszeitpunkt t.*

[4] Außer es gibt die Möglichkeit auf ein Multicastsendeverfahren zurückzugreifen. Bei Internetanwendungen ist dies nicht möglich.

Überprüfungsalgorithmus Der Zustand der Richtlinien-Datenbank zum Zeitpunkt t sei $\mathfrak{R}_t$. Das Einfügen eines Zertifikates $z = [p, S, \alpha, t]$ geschehe durch die Funktion $declare(z)$, wobei $S \in \mathfrak{C}$ und $p \in \mathfrak{T}$.

$$\mathfrak{R}_t \overset{declare(z)}{\rightarrow} \mathfrak{R}_{t+1},$$

ist erlaubt, wenn ein Zertifikat $z' = [p', S', \alpha', t'] \in \mathfrak{R}_t$ existiert und folgende Bedingungen gelten:

$$\alpha \leq \alpha'$$
$$\forall_{\omega \in S} \exists_{A' \in \mathfrak{G}} \exists_{\omega' \in \mathfrak{C}} \ \omega \leq \omega' \wedge A'\omega' \in S' \wedge p \in A'.$$

Das Zertifikat z ist dann akzeptiert und $\mathfrak{R}_{t+1} = R_t \cup \{z\}$, andernfalls gilt $\mathfrak{R}_{t+1} = R_t$. Die Akzeptanz von z ist gleichbedeutend mit der Ableitbarkeit z von z'. Die Ableitbarkeit ist keine eindeutige Relation, es kann also verschiedene Ableitungen geben.

Interpretation Ein Zertifikat (z) wird nur akzeptiert, wenn es bereits ein Zertifikat (z') beim PIP gibt, welches dem Aussteller (p) erlaubt, eine Autorisierung (α) zu delegieren. Die Autorisierung α muss dabei höchstens die Rechte oder Regeln umfassen, die α' umfasst hatte. Die Autorisierung darf aber beliebig eingeschränkt werden ($\alpha \leq \alpha'$). Desweiteren muss sich ein beliebiges ω finden lassen, welches Bestandteil des Constraints (S von z) ist und welches ω' entspricht oder eine Verfeinerung dessen ist und sich durch Vorstellen eines beliebigen Constraintelement A' zu Subjekt S' (von z') ergänzen lässt. Es muss schließlich nur noch überprüft worden sein, ob der Aussteller dem von ihm entnommenen Constraintelement A' angehört. Die gesamte Auswertung muss nur zum Zeitpunkt t gelten, um als Richtlinie in die Datenbank geschrieben zu werden.

Der Schlüssel der Überprüfung ist das Constraintelement A', welches vom Aussteller bezüglich des Vorgängerzertifikats entnommen werden muss. Im Beispielszenario in Anhang C.2.3 ist die Entnahme des selbst berechtigenden ersten Constraintelements bei der Weiterdelegierung der Autorisierung gut zu sehen. Ein Aussteller ist mit anderen Worten nur zur Delegierung berechtigt, wenn er ein Element des ersten Constraintelementes des Vorgängerzertifikates ist. Diese Abhängigkeit wird mit $p \in A'$ beschrieben. Das Subjekt des nachfolgenden Zertifikats ist dann vom vorhergehenden Subjekt ableitbar.

Widerruf

Bei AMANDA können Widerrufe dank der eindeutigen ID gezielt durchgeführt werden [77]. Der Aussteller kann jedes seiner Zertifikate genau widerrufen, indem er die ID des zu widerrufenden Zertifikats referenziert. Genau genommen handelt es sich vielmehr um eine Korrektur des Zertifikats, denn das Zertifikat kann nicht nur gänzlich widerrufen werden, sondern auch eingeschränkt werden. Die entscheidende Variable beim Widerruf ist die Zeit, ab wann dieser gelte. Wird ein Widerruf eingereicht,

wird also nicht das Bestätigungszertifikat mit einer ID gelöscht, sondern lediglich das Gültigkeitsintervall aktualisiert. Erhält das Bestätigungszertifikat ein Gültigkeitsintervall $1 < t < 1000$[5] und der Widerruf entzieht das Recht für $t > 500$, wird das resultierende Intervall auf $1 < t \leq 500$ abgeändert. In Anhang C.2.4 ist dies an einem Beispiel gezeigt.

Zurückversetzter Widerruf Eine spezielle Variante zum Widerruf bereits bestehender Zertifikate besteht darin, die Startgültigkeit des Widerrufs in die Vergangenheit zurückzuversetzen. Dies würde dann ebenfalls schon getätigte nachgefolgte Delegierungen wiederrufen. Wenn dieser zurückversetzte Widerruf möglich sein soll, dann müssen die PZerts in der Datenbank ebenfalls einen Zeitstempel erhalten, bei dem sie eingereicht wurden, damit bereits erstellte Delegierungen datiert werden können und zurückversetzte Widerrufe genau anwendbar sind.

Bei AMANDA ist ein Widerruf des Widerrufs zurzeit nicht implementiert, allerdings ist dies durchaus denkbar. Resultierend wären allerdings nicht zusammenhängende Zeitintervalle. Angenommen ein PZert beinhalte als zeitliche Beschränkung $5 < t < 25$ und dieses PZert werde widerrufen von einem WZert mit $t > 15$. Der Widerruf des Widerrufs gelte für das Intervall $t > 20$. Das resultierend gültige Zeitintervall bestünde dann aus zwei Zeitintervallen ($5 < t < 15 \cap 20 < t < 25$), was die Verarbeitung verkomplizierte. Wenn diese Funktionalität jedoch wünschenswert ist, dann braucht lediglich ein Widerruf formuliert und ein neues Zertifikat mit dem zweiten Zeitintervall eingereicht zu werden. Dadurch wird Gleiches erreicht. Der Widerruf des Widerrufs ist also nicht notwendig.

Die Wurzel des Vertrauens

Wie bei SPKI muss die Zertifikatskette auf einer Wurzel des Vertrauens basieren. Bei SPKI wird dazu eine ACL definiert, auf die sich jede Delegierung zurückführen lassen muss (5-Tupel-Reduktion). Bei AMANDA werden spezielle Einträge in der Richtlinientabelle verwendet, auf den sich die Ketten zurückführen lassen (der Eintrag ist dann „geerdet“). Diese können über ein leeres Ausstellerfeld dargestellt werden. Die Rückführbarkeit aller Delegierungsketten auf vertraute Einträge kann durch die Ableitungsrelation der Constraints in den Richtlinien nachgeprüft werden. Nur wenn Constraints nachfolgender Delegierungen (Zertifikate) sich von ihren Vorgängern ableiten lassen, und sich so eine Kette bis zu Eintrag ohne Aussteller zurückführen lässt, ist eine Richtlinie geerdet.

Wenn auf die Überprüfung der Gültigkeit beim Zeitpunkt des Einreichens von Zertifikaten verzichtet und diese Überprüfung immer nur auf den Zeitpunkt des Datenzugriffs verlagert wird, kann mit „schlafenden Ketten“ eine weitere Variante der Zugriffskontrolle eingeführt werden. Dabei handelt es sich um nicht-geerdete Ketten, die zu einem Zeitpunkt x durch Anbinden an eine Wurzel des Vertrauens später aktiviert werden. Es können also ganze Ketten gebildet werden, die nicht gültig sind, solange der Initialeintrag fehlt.

[5] Zur Vereinfachung wird eine logische Zeit gewählt, die Element der natürlichen Zahlen ist.

Allerdings können bei Verzicht auf die rechtzeitige Überprüfung ungültige Einträge die Datenbank überfluten, da jeder PZerts einreichen kann[6], die angenommen werden müssen. Abhängig vom Umfeld und der Vertrauensrelation zu den Mitbenutzern kann dies zu DoS-Angriffen führen. Die gleichen Angriffe könnten berechtigte Benutzer zwar in jedem Fall durchführen, doch haben diese unter Umständen einen anderen Vertrauensstatus und deren Interesse daran ist wahrscheinlich gering. Außerdem ist durch die Constraints eine Vielzahl der möglichen Einreichungen beschränkt, da bei einer rechtzeitigen Überprüfung nur PZerts angenommen werden, die dem Überprüfungsalgorithmus standhalten. Das Ermöglichen von schlafenden Ketten sollte also überlegt erfolgen.

Zusammenfassung

Während bei SPKI Zugriff und Vorlage der Zertifikatskette in einem Schritt passieren, werden diese beiden Konzepte bei AMANDA getrennt. AMANDA beschreibt lediglich den Delegierungsprozess der Autorisierungen um möglichst technologieunabhängig zu sein. Während sich bei SPKI nur Zertifikatsketten per 5-Tupel-Reduktion zu Richtlinien abbilden lassen, entspricht bei AMANDA jedes Zertifikat einer eigenen. Mit Verwendung der Verteilten Delegierung und öffentlichen Schlüsseln als Aussteller und Identifizierungsmerkmal könnte AMANDA so verwendet werden wie SPKI. Es hätte sogar die Möglichkeit Rahmenbedingungen für Delegierungsketten vorzugeben und dabei den Delegierenden nicht den Zugriff auf die Autorisierungen zu geben. Bei Verwendung in der Zentralen Delegierung ist AMANDA jedoch noch mächtiger und kann ganzen Gruppen Autorisierungen delegieren. In XACML wird die Zentrale Delegierung verwendet, wobei die Benutzersteuerung von XACML übernommen wird und die delegierte Richtlinienerstellung als reine administrative Tätigkeit der zentralen XACML-Richtlinien zu sehen ist. In diesem Umfeld müssen Teilnehmer nicht mit Zertifikaten anderen ihre Autorisierungen mitteilen, sondern ausschließlich dem System die neuen Richtlinien über XML-Nachrichten mitteilen.

Im folgenden Abschnitt wird beleuchtet, wie sich diese beiden Konzepte im MASC-ID umsetzen lassen. Das leichtgewichtige SPKI kann sehr schlank umgesetzt werden und jeder Teilnehmer einer Transportkette benötigt lediglich ein Schlüsselpaar, um daran teilzunehmen. Allerdings obliegt die Kontrolle der Rechteweitergabe den Parteien selbst. Mit AMANDA ist es hingegen möglich eine systeminhärente Kontrolle des Delegierungsbaumes vorzunehmen, was allerdings mit einer Teilnehmeranmeldung und Gruppenzuordnung jeder beteiligten Transportpartei einhergeht.

5.4 MASC-ID Zugriffssteuerung

Der MASC-ID basiert auf einem Datenbanksystem, das für jeden Transport[7] eine eigene Datenbank bereitstellt, die aus mehreren Tabellen besteht. Bild 5.1 zeigt schematisch, wie die Datenbank aussieht.

[6] Zumindest wenn keine umgebende Benutzerverwaltung die Anzahl der Benutzer beschränkt, sondern wie bei SPKI jeden teilnehmen lässt, der einen öffentlichen Schlüssel hat.

[7] Ein Transport beschreibt das einmalige Durchlaufen einer Transportkette von und zu jeweils einem Containerdepot.

Die Schnittstellen zur Gegenstelle und zum MASC-ID sind dabei in zwei gegenüberliegenden Ecken dargestellt. Der größte Teil der Datenbank ist unabhängig von der Wahl des Delegierungsverfahrens.

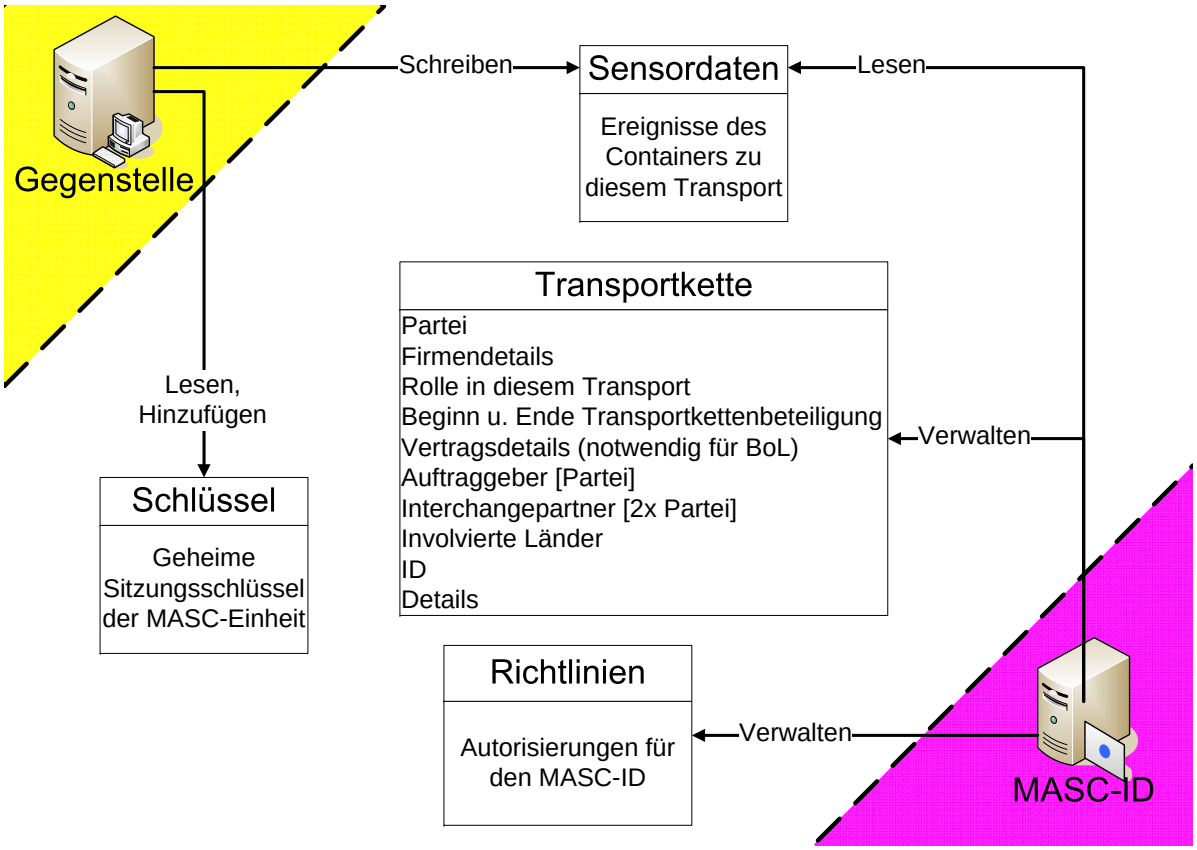


Abbildung 5.1: Transport-Datenbank

In der *Schlüssel*-Tabelle der Transportdatenbank sind zum einen die geheimen symmetrischen Schlüssel gespeichert, die als verteilte Geheimnisse für den MASC-ST dienen. Zum anderen enthält sie alle Langsitzungsschlüssel und den aktuellen Kurzsitzungsschlüssel (abhängig von der Länge der Kurzsitzungen). Der Inhalt der Tabelle kann aus Effizienzgründen auf mehrere Tabellen aufgeteilt werden.

In der Tabelle *Sensordaten* werden die Containerereignisse abgespeichert. Sie ist der Spiegel des Ereignisspeichers in der MASC-Einheit und die einzige (pro Transport), auf die Gegenstelle und MASC-ID gemeinschaftlich zugreifen. Die Gegenstelle erhält exklusiven Schreib-/Lesezugriff, der MASC-ID lediglich Lesezugriffsrechte. Dadurch gibt es auch keine Probleme bei gleichzeitigem Zugriff.

Die *Transportkette*-Tabelle enthält die notwendigen Daten zu allen beteiligten Parteien und zum Transport, um die automatische Dokumentenerstellung zu bedienen. Der Inhalt wird von den Transporteuren und dem Shipper (gegebenenfalls über den Spediteur) selber erstellt und in der Datenbank (über den MASC-ID) abgelegt (siehe 5.4.5).

Als PIP für die Zugriffskontrolle dient die Tabelle *Richtlinien* aus Bild 5.1. Bei SPKI würde sie die SPKI-ACL und die Widerrufliste bereitstellen und wäre somit die Vertrauensbasis für den verteilten

PAP. Bei Verwendung von AMANDA entspräche diese Tabelle der Zertifikatstabelle, bestehend aus PZerts und WZerts, also den Richtlinien für den Datenzugriff.

Die Zugriffskontrolle muss zum einen den lesenden Zugriff auf die *Sensordaten* überwachen und zum anderen dafür Sorge tragen, dass die Teilnehmer ihre Transportdaten in der *Transportkette* hinterlegen können. Die Zugriffsrichtlinien werden dezentral von den Teilnehmern selber unter Verwendung von SPKI oder AMANDA erstellt. Der PAP ist somit verteilt auf alle Teilnehmer. Bei Verwendung von SPKI stellt das System eine Verquickung von PDP und PEP bereit, denn beim Zugriffsersuch wird auch die Zertifikatskette eingereicht. Das Erstellen der Richtlinien (Auswerten der Zertifikatskette) und das Gewähren des Zugriffs werden in einer Transaktion durchgeführt. Bei Verwendung von AMANDA werden die Zertifikate einem PDP übergeben, der die Richtlinien in die Datenbank einfügt. Beim späteren Zugriff meldet sich der Benutzer lediglich beim PEP an. Die Zertifikate in der *Richtlinien*-Tabelle bilden den PIP und der MASC-ID übernimmt die alleinige Kontrolle über diese Datenbank.

5.4.1 Generelle Funktionsweise

Zunächst wird ein neuer Transport vom Spediteur bei einem MASC-ID seiner Wahl angemeldet. Dieser bestimmt einen (MASC-)Container, der dem Transport zugewiesen und zu Beginn des Transports von einem Frachtführer an einem Depot abgeholt wird[8]. Im Datenbanksystem wird bei der Registrierung eine neue Datenbank für den Transport angelegt. Der Spediteur sorgt im weiteren Verlauf dafür, dass er oder durch Delegierung seine Unterauftragnehmer die *Transportkette*-Tabelle mit Daten füllen.

Es ist die Aufgabe des Systems, rechtzeitig vor Transportbeginn alle benötigten Daten zu sammeln und die Transportdokumentierung zu manifestieren (Exportpapiere, 24-Stunden-Manifest, Frachtpapiere et cetera). Entweder können die Zollbehörden berechtigt, werden auf die dynamischen Daten lesend zuzugreifen, oder das aktuelle starre Bild der Transportkette (Datenbank) wird erstellt und den betroffenen Zollbehörden per Dokument (Status Quo) mitgeteilt.

Parallel zur Dokumentenerstellung sorgt der Spediteur initial dafür, dass die Leserechte der Sensordaten entsprechend verteilt werden. Zunächst kann er auswählen, welche Sensordaten für den Transport überhaupt zur Delegierung freigegeben werden sollen. Er kann sich dabei aus einer Liste die gewünschten Sensoren bestellen. Die initialen Richtlinien werden zu Gunsten des Spediteurs angelegt und machen ihn zum PAP für diesen Transport.

Das Beispiel in Anhang C.3.1 zeigt die ersten möglichen Delegierungsschritte für einen Transport eines Consolidation-Containers.

[8] Es wird angenommen, dass ein Transport o.B.d.A. aus nur einem Container besteht. Werden mehrere Container transportiert, sind mehrere Transporte anzumelden. Wie dies den Systembenutzern dargestellt wird ist Angelegenheit der Anwendungssoftware und nicht des MASC-ID.

5.4.2 Identitäten im MASC-ID

Alle Parteien, die auf den MASC-ID lesend oder schreibend zugreifen, seien die Teilnehmer. Basis für die Identifizierung eines Teilnehmers bildet ein selbst (beziehungsweise mit Hilfe eines Programms) erstelltes asymmetrisches Schlüsselpaar. Der öffentliche Schlüssel ist das Identifizierungsmerkmal. Mit Hilfe einer mit dem privaten Schlüssel erstellten Signatur ist eine Authentisierung (Nachweis des Schlüsselbesitzes) eines Teilnehmers möglich.

Der Spediteur geht eine Kundenbeziehung mit dem RSO ein, indem er einen MASC-Transport bestellt. Dazu ist eine Registrierung notwendig, die der RSO dem Spediteur bekannt macht. Es ist davon auszugehen, dass die RSO im Rahmen dieser Registrierung eine nachweisbare Zuordnung von öffentlichem Schlüssel zur Partei und gegebenenfalls die Eingabe der Bankverbindung erfordert, da der Spediteur ein haftender Vertragspartner ist. Wie die Registrierung auszusehen hat, ist allerdings Sache der RSO und kann von ihr frei gewählt werden. Schließlich dient eine interne Zuordnung von Identifizierungsmerkmal zur Kundendatei der Wiedererkennbarkeit eines Spediteurs. Für die RSO sind neben den Zollbehörden die Spediteure die einzigen Kunden in dieser Anwendung.

Der Spediteur wurde von einem Shipper beauftragt und stellt die Transportkette zusammen, indem er mit weiteren Parteien Vertragsbeziehungen eingeht. Es liegt in seinem Verantwortungsbereich, auf welcher Basis die Überprüfung dieser Parteien erfolgt. Da viele Verträge per Fax oder E-Mail abgewickelt werden, wäre eine Übersendung des Schlüssels und der Delegierungen per E-Mail denkbar. Eine Benutzersoftware für den MASC-ID würde Dateien erzeugen, die per E-Mail an den anderen gesendet würden. Ein Abgleich von Hashwerten der Dateien könnte parallel telefonisch, per Fax oder per direkter Übergabe überprüft werden[9]. Es obliegt also den Parteien selbst, welche Informationen der Vertragspartner offenbaren muss und wie sie die erhaltenen Daten überprüfen möchten.

Gleiches gilt für alle weiteren Beziehungen im Vertragsbaum. Jeder Vertrag entspricht einer potenziellen Delegierung. Jeder Teilnehmer ist selber dafür verantwortlich, welche Anforderung er an die Nachprüfbarkeit der Identität des Vertragspartners stellt. Für das System existieren nur öffentliche Schlüssel und jeder Teilnehmer selbst kann eine private Zuordnung von öffentlichen Schlüsseln zu Zusatzinformationen der korrespondierenden Partien führen.

Bei Verwendung von AMANDA müsste sich jeder Teilnehmer bei einem übergeordneten Verbund der RSOs registrieren[10]. Die RSOs können bestimmen, welche Anforderungen an die Registrierung gestellt werden, welche Daten welche Parteien auf welche Weise zu übermitteln haben. Sollten Zollbehörden dieses System akzeptieren, aber darauf bestehen, dass alle Teilnehmer auf eine geforderte Weise identifizierbar sein sollen, wäre dieses mit AMANDA umsetzbar. Eine Kopplung dieser Angaben mit den Beschreibungen der Parteien für die Transportkettenauswertung würde weitere Synergien erzeugen.

[9] Vergleichbar mit der Überprüfung des HBCI-Schlüssel beim Online-Banking.

[10] Das Einrichten eines Systems, bei dem alle RSOs zusammenarbeiten, sollte kein Problem darstellen, da bereits alle unter dem Dach der IMO zusammengeschlossen sind. Auf den Verbund könnte verzichtet werden, allerdings müsste sich eine Partei bei jeder RSO neu anmelden, wenn sie den einen Transport mit einer anderen RSO durchführt.

Die RSO kann aber auch auf diese Anforderung verzichten und die Systemanmeldung kann ungeprüft erfolgen.

Bei Verwendung einer Registrierung mit Eingabe von parteibezogenen Daten könnte zusätzlich erwogen werden, dass Vertragspartner ebenfalls die Möglichkeit bekommen, diese Registrierungsdaten (oder einige devon) einzusehen. Auf diese Weise könnten sie sicher gehen, mit welcher Partei sie einen Vertrag eingehen.

5.4.3 MASC-ID mit SPKI/SDSI

Der organisierende Spediteur erhält ein SPKI-Zertifikat, dessen Autorisierungen den Zugriff auf die Datenbank erlauben. Dabei sind alle potenziellen Befugnisse enthalten, welche die Transportkette und der Shipper haben sollen. Die für die Terrorbekämpfer bestimmten Sensordaten sind dementsprechend nicht enthalten. Es ist folglich des Spediteurs Entscheidung, ob er die *Transportkette*-Tabelle selber ausfüllt oder dies an die Unterauftragnehmer delegiert. Diese stehen vor der gleichen Entscheidung. Entweder erstellen sie die Transportketteneinträge ihres Teilbaums selber oder sie delegieren die Aufgaben an ihre Unterauftragnehmer. Dies geschieht mit SPKI-Delegationen der eigenen Zugriffsbefugnisse. Es entsteht eine beim Spediteur beginnende Zertifikatskette, die bei dem letztendlichen Zugreifenden endet.

Die Schachtelung der SPKI-Autorisierungspfade kann dabei komplexer werden, wenn SPKI-Zertifikate nicht nur an die Unternehmer, sondern auch an Mitarbeiter in den einzelnen Firmen weitergegeben werden. Dies ist durchaus sinnvoll, wenn von verschiedenen Arbeitsplätzen auf die Daten zugegriffen werden soll und die Unternehmen keinen zentralen Zugriffsproxy verwenden wollen.

Die vereinfachte Abbildung 5.2 zeigt an Hand eines Beispiels, wie sich die Informationsverteilung des MASC-ID in eine Baumstruktur überführen lässt. Komplexer (und nicht mehr zweidimensional knotenfrei darstellbar) wird die Struktur, wenn mehrere Transporte betrachtet werden und verschiedene MASC-IDs beteiligt sind. Dann hat der Baum nicht mehr nur eine Wurzel, sondern jeder Informationsdienst-Server bildet einen Ausgangsknotenpunkt. Darüber hinaus könnte eine Spedition als Unterspedition und in einem anderen Transport als deren „Überspedition“ auftreten und es entsteht ein komplexer Graph. Je unübersichtlicher die graphische Abbildung dieser Struktur ist, desto sinnvoller ist eine dezentrale Verwaltung der Zugriffsrichtlinien, besonders wenn die Skalierbarkeit des Gesamtsystems gefordert wird.

Delegierung

Bei der Verwendung von SPKI entspricht die Datenbanktabelle *Richtlinien* zwei Tabellen (siehe Abbildung 5.3). Die *SPKI-ACL* dient dabei als Wurzel des Vertrauens. In ihr werden die initialen Delegierungen verankert. Gemäß dem Beispiel von Abbildung 5.2 entsprächen die ACL-Einträge den drei SPKI-Zertifikaten vom MASC-ID zu den Spediteuren respektive zur Zollbehörde.

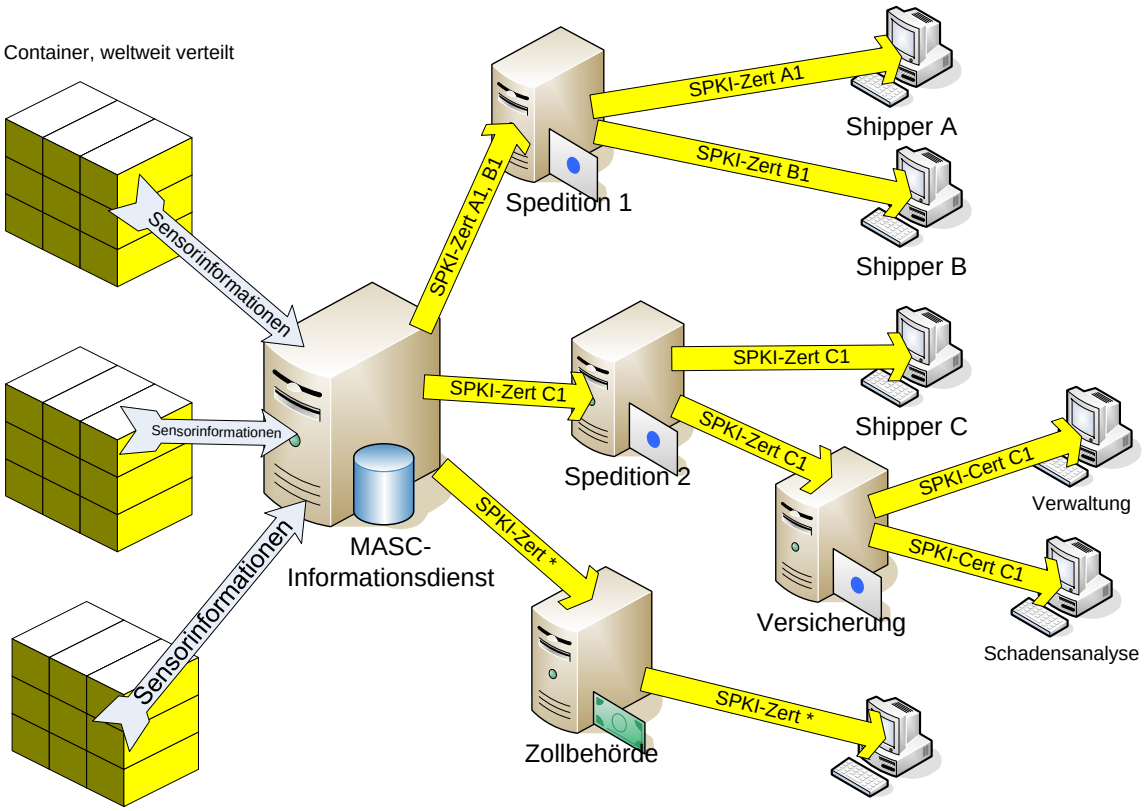


Abbildung 5.2: Rechteweitergabe mit SPKI für einen Transport (vereinfacht und beispielhaft)

Die Gültigkeit aller Autorisierungen wäre auf den Transport und die anschließende Verweildauer in der Datenbank begrenzt. Da dies aber durch die Lebensdauer der Transportdatenbank bestimmt wird, benötigen die ACL-Einträge und die SPKI-Zertifikate im MASC-Szenario keinen Gültigkeitszeitraum. Sie müssen sich lediglich eindeutig auf einen Transport (Transport-ID) beziehen.

Der Spediteur kann im Folgenden die ihn berechtigenden Autorisierungen gemäß SPKI an andere delegieren und so selber Richtlinien (Zertifikatsketten) für den Transport erstellen. Damit eine Zertifikatskette wirksam ist, muss also der Aussteller (Spediteur, Zollbehörde) eines ersten Zertifikats der Kette in der ACL als Subjekt geführt sein. (Zur Erinnerung: ACL-Einträge enthalten niemals einen Aussteller, da sie per definitionem gelten.) Zur Überprüfung der Gültigkeit ist ein erfolgreiches Abschließen der 5-Tupel-Reduktion[11] und der Signaturüberprüfungen erforderlich.

AUSSTELLER	Hash: Öffentlicher Schlüssel von Spedition A
SUBJEKT	Hash: Öffentlicher Schlüssel von Frachtführer C
AUTORISIERUNG	(Transport-ID:) *11204*, (Autorisierungen:) α, β
DELEGIERUNG	`true`

Tabelle 5.7: Beispiel einer ersten SPKI-Delegierung

[11] Es wird weiterhin von 5-Tupel-Reduktion gesprochen, obwohl das Weglassen des Gültigkeitsfeldes erfolgt ist (4-Tupel-Reduktion ist bereits anderweitig definiert).

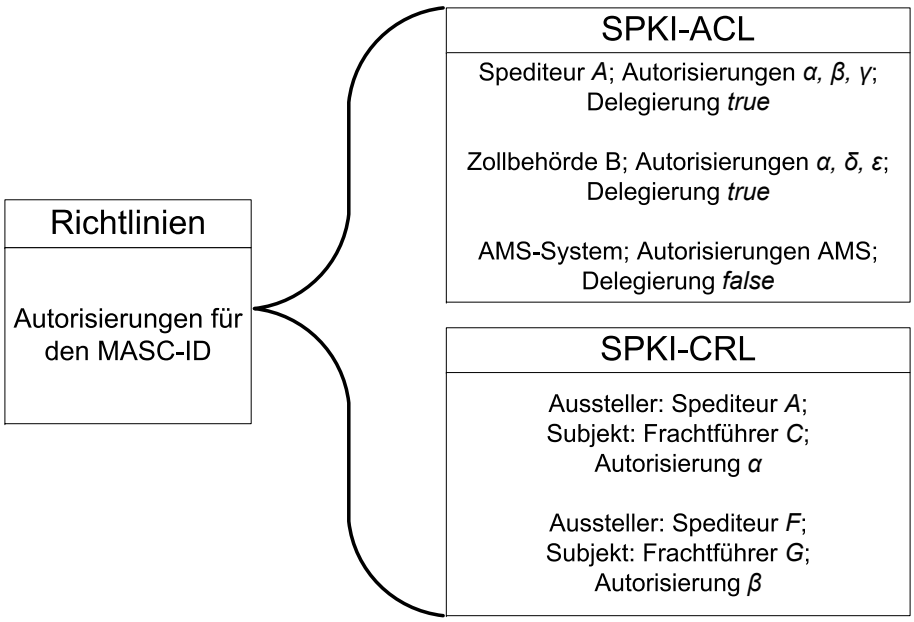


Abbildung 5.3: Zugriffsrechte-DB bei SPKI

Tabelle 5.7 zeigt ein Beispiel, wie ein Zertifikat aussieht, welches der Spediteur folgend einem Frachtführer ausstellen kann. Gemäß der ACL könnten für den Spediteur die Zugriffsrechte *a, b* und *c* gelten. Das Zertifikat 5.7 überträgt die Autorisierungen α und β auf den Frachtführer *C*. Die Transport-ID *11204* sei dabei exemplarisch die des betreffenden Transportes.

Die Autorisierungen entsprechen einer Zuordnung der Datenbanktabellen, Datenbankzeilen und Datenbankspalten mit den Zugriffsrechten, die der Begünstigte für diese Ressource erhält. Eine textuelle Darstellung einer Autorisierung könnte folgendermaßen aussehen:

(Sensordaten:(Türöffnung,Temperatur))lesen; (Transportkette{id=5}:∗)verändern

Diese Autorisierung erlaubt den Lesezugriff auf die Spalten Türöffnung und Temperatur der Datenbanktabelle *Sensordaten* und das Verändern einer ganzen Zeile in der Tabelle *Transportkette*. Die Identifizierung der Zeile geschieht über das Element $id = 5$ und bezeichnet die Zeile, die in der Spalte *id* den Wert fünf annimmt. Dieses Beispiel einer Autorisierung steht für die Autorisierungen, wie sie in einem Datenbanksystem verwendet werden können. Abhängig von der Wahl der Datenbank sollten diese angepasst werden. Ob sich diese mit den SPKI-eigenen S-Expressions (siehe [29, 30]) oder direkt in Datenbanksyntax darstellen lassen sollten, ist eine Optimierungsfrage, die von der Wahl des Datenbanksystems abhängt.

SPKI unterscheidet sich von AMANDA hauptsächlich darin, dass bei der Delegierung die Aussteller selber auf die korrekte Rechteverteilung achten müssen. Der Vorteil ist, dass durch die offene und nicht reglementierte Vorgehensweise die Elternknoten entscheiden können, ob sie die Aufgabe ihrer

Kindknoten übernehmen oder ob und welche Rechte sie delegieren. Der Nachteil ist, dass dabei Fehler oder Unachtsamkeiten auftreten können. So gibt es keine Möglichkeiten, die Delegierungen von einer Kontrollinstanz überwachen zu lassen. Die einzige Einschränkung besteht darin, dass jeder Beteiligte höchstens seine eigenen Erlaubnisse delegieren kann – wie er mit diesen umgeht, liegt aber außerhalb der Kontrolle.

Widerruf

Der Zertifikatswiderruf der Autorisierungszertifikate kann bei SPKI sehr unterschiedlich umgesetzt werden. Drei Varianten werden folgend diskutiert (vgl. [2]).

1. Die SPKI-RFCs [29, 30] schlagen für den Zertifikatswiderruf eine Widerrufliste vor, die von jedem Zertifikatsaussteller (also den zugreifenden Parteien) bereitgestellt wird. Die Liste müsste zur Laufzeit bei der Autorisierungsentscheidung vom Wächter online überprüft werden.

   Dieser Ansatz ist für den MASC-ID nicht geeignet, da der Aufwand, dass alle Logistiker und eventuell Shipper einen Server bereitstellen müssten, viel zu groß ist und bei jeder Zugriffsentscheidung eine zusätzliche Datenkommunikation aufgebaut werden müsste.

2. Alle Zertifikate werden durch den MASC-ID mit einer *ID* identifiziert. Eine lokale Liste aller IDs der widerrufenen Zertifikate wird zentral verwaltet. Die Teilnehmer müssten den MASC-ID von jedem Zertifikatswiderruf in Kenntnis setzen, der anschließend die IDs in die Widerrufliste einträgt.

   Der Nachteil dieser Lösung ist die Zuweisung von IDs an alle Zertifikate im System. Es müsste dazu jeder Teilnehmer eine eindeutige ID zuordnen können. Es wäre denkbar, dass jeder Teilnehmer seinen Hash des öffentlichen Schlüssels verwendet und mit einem laufenden Zähler ergänzt. Allerdings kann dann auch gleich der öffentliche Schlüssel referenziert und auf die *ID* verzichtet werden, wie folgender Ansatz zeigt.

3. Die widerrufenen Zertifikate werden nicht durch IDs, sondern durch bestimmte Zertifikatattribute identifiziert. Die Teilnehmer müssen aber auch hier den MASC-ID über jeden Widerruf informieren. Zwar stellt das eine Zentralisierung bezüglich des Zertifikatswiderrufs dar. Da aber Widerrufe beim MASC-ID als nur selten auftretend angenommen werden, entsteht durch diese Lösung kein Effizienzverlust.

   Da jedes Zertifikat nur bezüglich eines Transportes gültig ist, muss es auch nur bezüglich des entsprechenden Transportes widerrufen werden. Daher müssen die Tabelleneinträge in der Widerrufliste nicht alle Informationen über das Zertifikat enthalten, sondern nur Autorisierung, Zertifikatsaussteller und Zertifikatsbegünstigten. Es werden auch nicht genaue Zertifikate widerrufen, sondern alle Delegierungen bezüglich der genannten Autorisierungen, die vom Aussteller an das Subjekt erfolgten.

Das Beispiel aus Abbildung 5.3 widerruft die Autorisierung β bezüglich des speziellen Transportes für alle Zertifikate, die von Spedition *A* ausgestellt wurden und Frachtführer *C* als Begünstigten enthalten. Ausgehend davon, dass Autorisierungszertifikat 5.7 ausgestellt wurde, wäre dieses Autorisierungszertifikat für zukünftige Zugriffe so eingeschränkt, dass die Autorisierung β nicht mehr gälte. Wenn kein Widerruf des Widerrufs implementiert wird, dann ist es nicht mehr möglich, dem Subjekt später ein Zertifikat auszustellen, was ihm die Autorisierung zurückgibt. Da der Eintrag in der Widerrufliste steht und einzelne Zertifikate nicht identifiziert werden können, gilt der Widerruf des Zugriffsrechts für alle jemals ausgestellten und auszustellenden Zertifikate, die den gleichen Aussteller und das gleiche Subjekt haben (bezüglich des einen Transportes).

Die Einschränkung der Verwendung dieses Widerrufsschemas ist, dass Ergebniszertifikate nicht mehr zur Optimierung verwendet werden können. Der Verifizierer muss bei jedem Zugriff die gesamte Transportkette auswerten, damit er sicherstellen kann, dass keines der Zertifikate widerrufen wurde. Sollen aus Effizienzgründen die Ergebniszertifikate weiterhin verwendet werden, muss sich der Verifizierer alle Zertifikate zu jedem Ergebniszertifikat merken und bei Widerruf neue Ergebniszertifikate erstellen. Beim nächsten Zugriff würde dieses dann aktualisiert. Allerdings steigt dabei der Aufwand auf dem Server und die lokale Datenbank der Ergebniszertifikate wäre eine starke Annäherung an AMANDA. Daher sollte im MASC-Fall entweder gleich AMANDA benutzt, auf Widerruf gänzlich verzichtet oder die Verwendung der Ergebniszertifikate verworfen werden.

5.4.4 MASC-ID mit AMANDA

Die *Zentrale Delegierung* mit dem MASC-ID umzusetzen, erforderte eine eigene Benutzersteuerung, wie sie unter anderem bei Web-Services und XACML vorliegt. Der gesamte Dienst müsste also auf einer Internetanwendung mit eigener Benutzersteuerung basieren. Jeder eingeladene neue Teilnehmer müsste sich registrieren und bei jedem Zugriff beim System anmelden; das System würde zentral verwaltet. Die Richtlinien würden zwar AMANDA-konform von den Teilnehmern selber erstellt, die schlanke Struktur ginge aber verloren. Der große Vorteil der Verwendung der *Zentralen Delegierung* ist im Vergleich zum verteilten AMANDA-Pendant die mögliche Nutzung von direkten Gruppenrichtlinien – der Nachteil die große Zentralität der Anwendung. In Abschnitt 5.5.2 wird auf eine Testimplementierung mit XACML eingegangen.

Für den MASC-ID wird folgend AMANDA in der *Verteilten Delegierung* betrachtet. Identifizierung und Authentifizierung der Parteien erfolgt gemäß Abschnitt 5.4.2 mit öffentlichen Schlüsseln. Zur Richtlinienübermittlung werden signierte Zertifikate und für die Zugriffsgesuche eine signierte Anfrage verwendet, die sich in ihrer Syntax an die Autorisierungen anlegt. Die eingebettete MASC-ID-Zugriffssoftware übernimmt dabei die benötigten Abstraktionen einer GUI.

Wegen der in Abschnitt 5.3.2 genannten Angreifbarkeit sollen die Zertifikate bei der Richtlinienerstellung überprüft werden. Somit ist eine Teilnehmerregistrierung (und Gruppenzuordnung) Pflicht, bevor die Richtlinien in den PIP aufgenommen werden. Es wird dabei offen gelassen, welche Kriterien

bei dieser Registrierung angefragt werden. Lediglich die Zuordnung der möglichen Rollen (Spedition, Transporteur, Shipper, Versicherer) einer Partei muss dabei erfolgen. Die Erstellung von schlafenden Ketten soll wegen der besprochenen Angreifbarkeit (siehe 5.3.2) nicht möglich sein.

Zertifikate

AUSSTELLER	Öffentlicher Schlüssel Aussteller (Hash)
SUBJEKT	*Constraint*
AUTORISIERUNG	Datenbankautorisierungen
SERIENNUMMER	*eindeutige ID*

Tabelle 5.8: AMANDA-Privilegierungszertifikat der MASC-Anwendung

AUSSTELLER	Öffentlicher Schlüssel Aussteller (Hash)
SERIENNUMMER	*ID eines Privilegierungszertifikats*
GÜLTIGKEIT	(binär) rückwirkend/zukünftig

Tabelle 5.9: Beispiel: Widerrufszertifikat in AMANDA

Die Tabellen 5.8 und 5.9 stellen dar, wie die AMANDA-Zertifikate in der SPKI-ähnlichen Variante aussehen. Der Aussteller wird gleichermaßen über seinen (Hash vom) öffentlichen Schlüssel identifiziert und durch eine angehängte Signatur authentifiziert. Schließlich erhält jedes PZert eine ID, damit es von einem WZert referenziert werden kann.

Die Wahl der ID ist dabei nicht trivial, weil sie vom Zertifikatsaussteller zu wählen ist. Da sich IDs nicht wiederholen dürfen, muss eine Struktur gewählt sein, die eindeutig ist. Durch die Möglichkeit, Zertifikatsketten erstellen zu können, bevor ein Teilnehmer registriert ist und eine eindeutige Teilnehmer-ID erhalten hat – die Registrierung muss ja erst vor dem Einreichen der Zertifikatskette erfolgen, nicht vor deren Erstellung –, muss ein Teilnehmer ohne ID in der Lage sein, eine systemweit eindeutige Nummer zu vergeben. Folgender Ansatz ist für die Seriennummer gewählt:

Seriennummer: Teilnehmer-ID|Zertifikatszähler|Delegierungszähler1|Delegierungszähler2

Es gibt bei diesem Ansatz drei Zähler: Den Zertifikatszähler und die beiden Delegierungszähler. Ersterer wird nur von dem Teilnehmer hochgezählt, der in Besitz der Teilnehmer-ID der Seriennummer ist, letztere von Delegierten. Bei Erstellung eines Folgezertifikats einer Zertifikatskette hat der Aussteller die Wahl auf zwei Weisen eine gültige und eindeutige Seriennummer zu erzeugen. Entweder kann er eine Seriennummer unter der Wahl seiner Teilnehmer-ID, dem nächsten freien *Zertifikatszähler* und den Delegierungszählern 0|0 generieren oder die Teilnehmer-ID und den Zertifikatszähler des vorhergehenden Zertifikats wiederverwenden und den *Delegierungszähler1* um 1 inkrementieren. Bei letzterer Variante können beliebig viele[12] Zertifikate erstellt werden, die alle durch die Veränderung

[12] Obergrenze durch Größe des *Delegierungszähler2* definiert.

des *Delegierungszähler2* unterschieden werden können. Damit ist es möglich, Zertifikate zu erstellen, ohne bereits eine Teilnehmer-ID zugeteilt bekommen zu haben. Bei zeitaufwändigeren Registrierungsprozeduren könnten Teilnehmer unregistriert bereits Privilegierungszertifikate erstellen. Das gültige Einreichen der entstehenden Zertifikatskette ist aber erst nach der Registrierung möglich.

Da in dieser Anwendung auf Zeitstempel verzichtet wird, benötigt der Widerruf ebenfalls kein Zeitintervall, für das der Widerruf gilt. Dabei wird das WZert ebenfalls in den PIP eingetragen. Es gibt beim Widerruf zwei Optionen. Entweder wird die zu widerrufende Richtlinie gelöscht, was bereits getätigte Delegierungen ebenfalls ungültig macht, oder sie bleiben bestehen und sind damit noch gültig. Im zweiten Fall werden keine neuen Delegierungen mehr erlaubt, alte Delegierungen bleiben aber gültig. Dies kann problemlos umgesetzt werden, indem der PDP bei der Richtlinienüberprüfung (der Zeitpunkt, zu dem er Zertifikate des PAP entgegennimmt) überprüft, ob ein Kettenelement bereits widerrufen wurde, sprich ob es ein WZert mit der ID eines Kettenelementes gibt. Beim Zugriff und der Auswertung der Richtlinien werden WZerts nicht berücksichtigt, da sie sich auf bereits enthaltene Richtlinien nicht auswirken.

Zu dieser Widerrufregelung sei gesagt, dass sich die Bezeichnungen *rückwirkend* und *zukünftig* wegen der Verteilten Delegierung nicht wirklich auf den Erstellungszeitpunkt der Zertifikate beziehen, sondern auf ihren Einreichungszeitpunkt.

Zugriffsteuerung

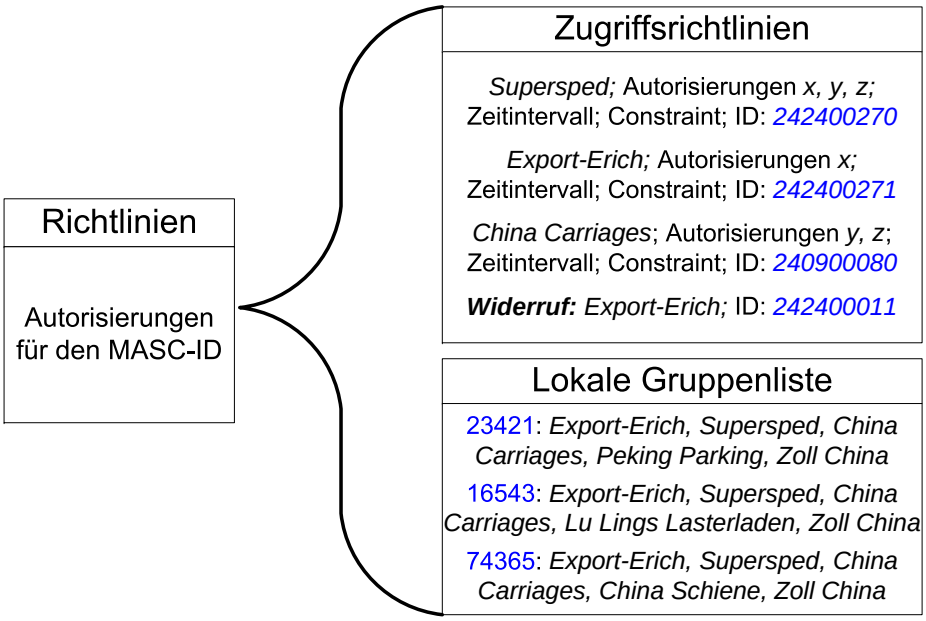


Abbildung 5.4: Zugriffsrechte-DB bei AMANDA

Die *Richtlinien*-Tabelle der Datenbank würde im AMANDA-Fall wie in Abbildung 5.4 abgeändert. Die Richtlinien sind in der Tabelle *Zugriffsrichtlinien* gespeichert. Sie enthalten inhaltlich die AMANDA-

Zertifikate, nachdem diese überprüft worden sind. Ebenfalls gespeichert sind die Widerrufe, wobei sie auf den Aussteller und die ID beschränkt wurden. Ist ein Widerruf enthalten, wird jede Delegierung, die auf einer das widerrufene Zertifikat enthaltenden Kette basiert, nicht akzeptiert.

Nach der Teilnehmerregistrierung sind alle Teilnehmer **globalen Gruppen** zugeordnet. Diese beschreiben, welche Rollen die Teilnehmer in einem Transport einnehmen dürfen. Bezüglich des Transports legt der Spediteur oder ein Unterspediteur fest, welche Rolle die Parteien einnehmen, die in die Transportkette aufgenommen werden. Dies geschieht mit dem entsprechenden Eintrag in der *Transportkette*-Tabelle. Damit ist jedem Teilnehmer eine Rolle zugeordnet. Tritt er mehrfach in verschiedenen Rollen auf, muss es auch mehrere Transportkettenelemente in der Datenbank geben, die sich auf seinen Teilnehmer beziehen.

Wie in Abschnitt 5.3.2 erwähnt, können AMANDA-Constraints auch Bedingungen definieren, indem sie die Bedingung in einer Klammer anfügen: $A(\mathit{Bedingung})$. Dieses Konzept wird beim MASC-ID mit AMANDA verwendet, um Transportkettenbeteiligungen zuzuweisen. Auf diese Weise können Zugriffsrechte auf die Sensordaten mit der Bedingung *Teilnehmer ist am Teiltransport beteiligt* eingeschränkt werden.

Um den Teilnehmern neben ihrer Rolle auch ihre genaue Position in der Transportkette nachweisen zu können, wird der Transport in kleinste Blöcke aufgeteilt. Jeder dieser Blöcke entspricht einem Teiltransport, bei dem genau ein Transporteur für den Container verantwortlich ist, und wird durch Interchanges begrenzt. Für jeden Teiltransport wird eine **lokale Gruppe** angelegt. Der Gruppe werden als Mitglieder alle an diesem Teiltransport beteiligten Parteien zugewiesen: die Shipper, deren Ware im Container ist, die verantwortlichen Spediteure, die Versicherer der Ware und gegebenenfalls des Transports, die Frachtführer und eventuell die Zollbehörden (bei Grenzübertritt). Identifiziert wird jede Gruppe über eine Transportmittel-ID, die jedem MASC-ausgerüsteten Depot, Terminal oder Verkehrsmittel eindeutig zugeordnet ist. In Anhang C.3.2 sind lokale Gruppen genauer und mit Hilfe eines Beispiels erklärt.

Durch Verwendung der Constraints mit Bedingungen und den Transportmitteln, die bei der Übertragung der Daten ihre ID mit angeben, können alle Ereignisse einer lokalen Gruppe zugeordnet werden. Auf diese Weise kann für jede Datenübermittlung eine Gruppe der Teilnehmer identifiziert werden, die zugreifen darf. Durch eine Bedingung kann diese Abfrage bei den Zugriffsrechten erzwungen werden. Wenn aufeinanderfolgende Transportmittel keine MASC-Infrastruktur anbieten und auch keine identifizierbaren Handlesegeräte verwendet werden, dann können die Zuordnungen von aufgetretenen Sensordaten zu lokalen Gruppen nur über die Zeitangabe und die Angabe der geplanten Interchanges erfolgen, wie sie in der Datenbank enthalten sind. Soll auf die Abfrage der Bedingungen verzichtet werden und jeder für den ganzen Transport berechtigt sein, die geforderten Sensordaten zu lesen, kann die Bedingung im Constraint weggelassen werden.

5.4.5 Transportdatenzugriff

Für die automatisierte Erstellung der Zolldokumente, des Manifests und der Ladungspapiere (BoL) ist es erforderlich, dass die beteiligten Logistiker schreibenden Zugriff auf die betreffenden Datenbankspalten bekommen. Da der MASC-ID bereits den lesenden Zugriff auf die Sensordaten für alle Parteien wie beschrieben regelt, lässt sich der Zugriff auf die Transportdaten über die gleichen Mechanismen in das System integrieren.

Datenbankstruktur

Transportkette	
Partei	*Shanghai Shipping*
Rolle	*Frachtführer*
Firmendetails	...
Transportbeginn	*20080122*
Transportende	*20080124*
Transportmittel	TM-ID
Vertragsdetails	...
Auftraggeber	*China Carriages*
Interchangepartner Beginn	*China Schienenverkehr*
Interchangepartner Ende	*Container-Claus*
Involvierte Länder	*China*
ID	8

Tabelle 5.10: Beispiel: Tabelle Transportkette

Gemäß Bild 5.1 sind die Transportdaten in der Tabelle **Transportkette** zusammengefasst. Ein Beispiel ist in Tabelle 5.10 dargestellt. Jede *Partei* ist ein Element der Transportkette. Parteien, die mehrfach vorkommen oder verschiedene Rollen einnehmen, müssen auch mehrfach in der Tabelle eingetragen werden. Jeder Eintrag besteht aus dem Namen der Partei und ihrer *Rolle* bezüglich dieses Transportes, also ob es sich um einen Spediteur, Frachtführer, Shipper oder Versicherer handelt. Für die BoL und das Manifest ist darüber hinaus relevant, um welche Firma es sich handelt, wo der Firmensitz ist et cetera. Bei der Verwendung von AMANDA können diese *Firmendetails* bei der Registrierung einmalig in das System eingepflegt werden und RSO-übergreifend zentral administriert werden. Diese Daten würden dann automatisiert in die Datenbank eingefügt. Bei SPKI wären die Firmen selber für die korrekte Dateneingabe verantwortlich, wie es auch zurzeit bei den BoLs der Fall ist. Weiterhin ist der *Bereich des Mitwirkens* der Partei relevant. Bei Transportführern ist dies die geplante Übernahme und spätere Übergabe des Containers. Bei dem Shipper und seinem Versicherer ist es das Intervall zwischen Containerbe- und -entladung. Bei Spediteuren entspricht dieses dem Bereich der Verantwortlichkeit. Die Transportmittel-ID ist eine Identifizierung des Gateways, mit dem der Container in diesem Teiltransport kommuniziert. Er identifiziert den Containerstellplatzort. Depots, Yards und Terminals haben genauso eine ID wie jedes *Transportmittel*, das mit einen MASC-Gateway bestückt ist. Die *Vertragsdetails* werden für die BoL benötigt. Jede Partei muss dort ihre Verantwortlichkeit der Transportkette

dokumentieren. Der Shipper teilt in diesem Rahmen Informationen über die Fracht mit; Frachtführer versichern sich an Transportbedingungen zu halten; Versicherungen teilen ihre Bedingungen mit und Speditionen erläutern ihre Haftbarkeit ... Als nächster Eintrag enthält die Datenbank einen Verweis zu der Partei, die der *Auftraggeber* ist. Gemäß dem Vertragsbaum ist es der jeweilige Elternknoten. Der organisierende Spediteur wurde vom Shipper beauftragt. Shipper haben keinen Eintrag in diesem Feld. Die beiden Felder für *Interchangepartner* sind nur bei Frachtführern, Containeryards und Containerterminals gefüllt. Containerdepots haben nur einen Eintrag. Versicherer, Shipper und Speditionen haben keine Interchangepartner. Die involvierten Länder müssen ebenfalls nur bei den direkt am Transport beteiligten Parteien eingepflegt werden. Die *ID* sorgt für eine bessere Referenzierbarkeit und Eindeutigkeit aller Transportkettenelemente. Weitere *Details* müssen die verschiedenen Partner gemäß ihrer Rolle im Transport angeben. Diese Vielzahl der Daten wird ebenfalls in der Datenbank gespeichert.

Die Einträge in den Spalten *Firmendetails* und *Vertragsdetails* werden für die automatisierte Verarbeitung der Ladungspapiere benötigt. Dabei benötigen die Zollbehörden spezielle Daten. Das amerikanische AMS-System erfordert die Eingabe wiederum anderer Daten. Mit Anhang E ist eine FAQ angefügt, die die Daten auflistet, die das AMS-System, der deutsche Zoll bei der Wareneinfuhr und die BoL benötigen. Entsprechend diesen Anforderungen müssten die beiden Tabellenspalten mit benötigten Daten gefüllt werden.

Teiltransporte

Diese Arbeit führt folgende Definition ein: Es gibt zwei Arten von Teiltransporten: *Logische Teiltransporte* und *echte Teiltransporte*. Ein echter Teiltransport bezieht sich auf einen Teiltransport, der sich auf den wirklichen Kontakt mit dem Container bezieht. Ein logischer Teiltransport entspricht mehreren echten Teiltransporten und bezieht sich auf den Verantwortungsbereich eines (Unter-)Spediteurs. In dem klassischen Beispiel, bei dem ein Spediteur den gesamten Transport organisiert und Subspediteure Teilbereiche übernehmen, entspricht der erste logische Teiltransport der gesamten Transportkette. Die weiteren logischen Abschnitte sind durch die Bereiche definiert, die von den Subspeditionen organisiert werden.

Jedes Element der Tabelle Transportkette, bei dem die Rolle *Spediteur* ist, ist ein logischer Teiltransport. Jedes Element, bei dem die Rolle *Frachtführer, Containerdepot, Containeryard* oder *Containerterminal* (also ein Transporteur) ist, entspricht einem echten Teiltransport.

Mit Bezug auf das Beispiel aus Abbildung C.3, bei dem der organisierende Spediteur ebenfalls ein Frachtführer für eine Teilstrecke ist, wären zwei Einträge in der *Transportkette*-Tabelle notwendig: Einer für den Spediteur und den logischen Teiltransport, der andere für den Frachtführer und den echten Teiltransport. Es gibt bei diesem Beispiel zwei logische Teiltransporte: Der erste umfasst die gesamte Transportkette und ist dem Datenbankeintrag *Supersped* mit Rolle *Spedition* zugeordnet, der zweite entspricht dem Transport der Unterspedition *China Carriages*, deren Rolle ebenfalls *Spedition*

ist. Alle anderen Einträge in *Teiltransporte* sind echte Teiltransporte. Die zugeordneten Parteien haben somit alle als Rolleneintrag entweder *Frachtführer*, *Containerdepot* oder *Containerterminal*.

Wenn alle beteiligten Parteien bei dem System angemeldet sind und jeder seine eigenen Daten im System verwaltet, existiert eine gewisse Redundanz zwischen der Transportdatentabelle und den lokalen AMANDA-Gruppen. Aus beiden ließe sich der Vertragsbaum zurückrechnen. Aus zwei Gründen sollten diese Konzepte aber nicht vermischt werden. Zum einen soll die Transportdatenverarbeitung auch mit SPKI zusammenarbeiten und daher unabhängig von den Zugriffsrechten sein. Zum anderen kann es sein, dass sich nicht alle Parteien an dem System beteiligen wollen und daher der Spediteur die Einträge seiner engagierten Frachtführer übernimmt. In diesem Fall weichen die AMANDA-Zugriffsrechte von der reinen Parteiensicht ab.

5.5 Testimplementierungen

Es wurden zwei Testimplementierungen durchgeführt, um die Machbarkeit des MASC-ID zu überprüfen. Dabei wurde zum einen die sehr dezentrale SPKI-Variante [2] und zum anderen ein System basierend auf *XACML delegation* [1] implementiert. Da es sich dabei um ein System basierend auf der Zentralen Delegierung handelt und XACML verwendet wird, ist diese zweite Testentwicklung eine Internetanwendung, die bis auf die Richtlinienerstellung alles zentral steuert.

5.5.1 SPKI-Implementierung

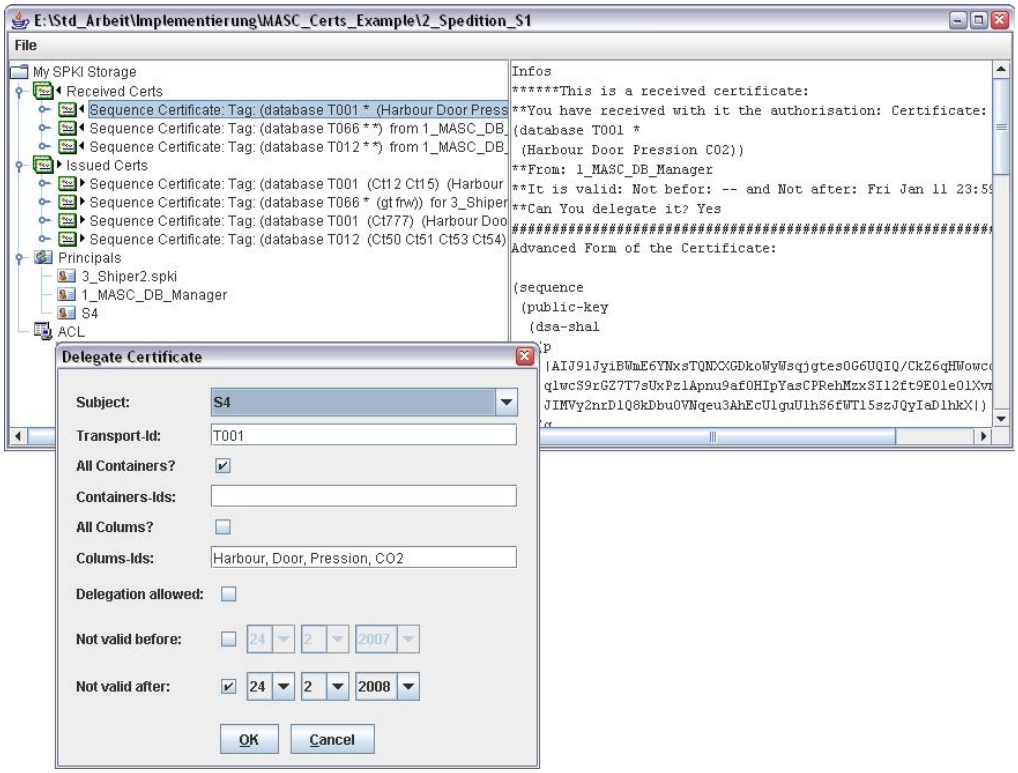


Abbildung 5.5: SPKI-Implementierung – Delegierungssicht

Im Rahmen einer SPKI-Implementierung wurde zwar noch keine einfache Oberfläche programmiert, die schlussendlich entscheidend ist, ob die Benutzer diese Anwendung benutzen oder nicht, aber die Machbarkeit der Anwendung dokumentiert [2]. Das Zugriffsmanagement auf SPKI-Basis funktioniert wunschgemäß. Abbildung 5.5 zeigt die GUI der Zertifikatsdelegierung, die die genauen Zertifikate und deren Rechte zeigt. Die Auswertung funktioniert bereits problemlos, und in einer anderen Sicht lassen sich alle Sensorwerte anzeigen, die der Benutzer ansehen darf. Die Sensorwerte wurden dabei frei gewählt.

Die Testimplementierung hat gezeigt, dass es notwendig ist, dass eine benutzerfreundliche GUI die Zertifikatsstrukturen vor ihren Anwendern versteckt. Die Darstellung der Hashwerte und der Zertifikatsstrukturen als „S-Expressions“ führt zu einer recht unleserlichen und unverständlichen Darstellung der Zertifikate. Ebenfalls wurden die angesprochenen Schwierigkeiten beim Widerruf festgestellt und evaluiert.

5.5.2 AMANDA-Implementierung

Eine weitere Testimplementierung basiert auf (der Vorabversion von) XACML 3.0 mit der Möglichkeit der Delegierung [1]. Es handelt sich dabei um die weiterentwickelte AMANDA-Version. Um in das XACML-Schema zu passen, werden keine Zertifikate als Richtliniencontainer verwendet, sondern XACML-Richtlinien im XML-Format. Die Testimplementierung geht o.B.d.A. davon aus, dass alle Parteien vorher den RSOs bekannt sind und ein Eintrag in einer globalen Datenbank vorliegt.

Der Spediteur meldet einen neuen Transport an. Der MASC-ID erstellt daraufhin eine neue Datenbank für diesen Transport (vgl. Bild 5.1). Bei der Transportanmeldung werden zwei Einträge in der *Transportkette* angelegt: Spediteur und Shipper. Die Daten des Spediteurs werden gemäß der globalen RSO-Datenbank ausgefüllt, die alle Parteien kennt. Der Eintrag des Shippers ist zunächst leer. Die Richtlinien erlauben dem Spediteur den Zugriff auf diese beiden Spalten, um die Daten fertig auszufüllen. Er darf dabei das Einfüllen des Shipper-Datensatzes an ihn delegieren.

Des Weiteren ist es dem Spediteur erlaubt, Spalten hinzuzufügen. Jede vom Spediteur hinzufügte Spalte erhält automatisch den Spediteur als *Auftraggeber*. Ist eine Spalte angelegt, darf deren Ausfüllen an die entsprechende Partei delegiert werden. Nur Spediteure dürfen neue Spalten anlegen. Für jede Delegierung muss die entsprechende Partei einen Richtlinien-Eintrag hinzufügen.

Enthält ein angelegter Eintrag die Rolle eines Transporteurs, wird automatisch ein lokaler Listeneintrag generiert. In dem Fall handelt es sich schließlich um einen echten Teiltransport und jeder echte Teiltransport entspricht einer lokalen Gruppe. Gemäß den bislang im System existierenden Spalten werden automatisch alle Parteien hinzugefügt, deren Transportkettenbeteiligungsintervall diesen Teiltransport einschließt.

Ist die Transportkette korrekt in der Datenbank eingetragen, sind automatisch alle lokalen Gruppeneinträge ausgefüllt. Jeder Benutzer darf nur Daten aus der *Sensordaten*-Tabelle sehen, wenn er in den

Teiltransport involviert ist, bei dem die Ereignisse auftraten. Diese Constraints sind fest im MASC-ID verankert. Wenn der Spediteur die Rechte an die anderen Parteien delegieren will, müssen diese die Constraints erfüllen. Wenn der Shipper die Temperaturdateneinsicht beim Spediteur bucht, dann delegiert der Spediteur lediglich die Einsicht der Temperaturereignisse an den Shipper. Wenn der Spediteur seinem Subspediteur alle Sensordaten delegieren will, muss er aufpassen, dass er das nur für den zeitlichen Bereich ausführt, in dem die Container im Organisationsbereich des Subspediteurs liegen. Eine weiterführende Delegierung würde das System (gemäß den Constraints) nicht erlauben.

Für die Umsetzung der Anwendung wurde die XACML-Implementierung in Version 2.0 von Sun [91] mit einem Patch der AMANDA-Erfinder [92] auf eine Testversion von *XACML delegation* gebracht. Trotz einiger Softwarefehler in der Testversion konnte die Mqachbarkeitsstudie entstehen. Das Anlegen eines neuen Transports, das Delegieren von Zugriffsberechtigungen und das Verwalten von Transporten sind in der Oberfläche integriert. Die Tabelle der lokalen Gruppen ist bei dieser Implementierung nicht enthalten, sondern wird implizit aus der Transportkettentabelle zum Zugriffszeitpunkt hergeleitet. Die Generierung und das Auslesen der Transportdaten-Tabelle wurden quasi auf den Zugriffzeitpunkt verlagert und demnach durch „Lazy Evaluation[13]“ nur lokal durchgeführt.

Die Implementierung zeigt, dass es durchaus eine Alternative ist, die Anwendung als Internetdienst zu zentralisieren. Zwar entsteht dadurch ein höherer (und vor allem unnötiger) zentraler Aufwand und die Möglichkeit der direkten Weitergabe unter den Vertragspartnern entfiele, aber als neue Anwendung im Containerwesen wäre auch diese Variante durchführbar.

5.6 Schlussbetrachtung MASC-IS

Genau genommen werden in diesem Abschnitt drei verschiedene MASC-IS Versionen vorgestellt. Der SPKI-basierte MASC-IS ist äußerst schlank und dezentral. Es gibt aber auch wenig systembedingte Kontrolle der Zugriffsregeln. Für die Ausbreitung der Autorisierungen sind ausschließlich die Zugreifenden verantwortlich. AMANDA in der Verteilten Delegierung ist ebenfalls dezentral organisiert, allerdings werden die Richtlinien zentral gespeichert. Folgend ist (sofern kein Widerruf erfolgt) der Zugriff über den PEP ohne erneute Vorlage der Zertifikate möglich. Das System hat volle Kontrolle, nur die Teilnehmerauswahl wird initial vom Spediteur durchgeführt und kann ebenfalls delegiert werden. Die Umsetzung einer zentralisierten Internetanwendung auf XACML-Basis ist die dritte Variante. Mit ihr sind Gruppendelegierungen möglich und der Zeitpunkt der Delegierung kann exakt erfasst werden. Dies lässt Widerrufe genauer funktionieren.

Die Hauptanforderung an die Richtlinienerstellung ist, dass sie dem Vertragsbaum folgt, um so eine gute Abbildung der Realität erzeugen zu können. Diese basiert jedoch auf Eins-zu-eins-Relationen. Daher sind Gruppendelegierungen nicht notwendig. Die Constraints beschränken auch in der Verteilten

[13] Fachbegriff für das Verlagern einer Berechnung auf einen späteren Zeitpunkt. Die Berechnung wird erst dann durchgeführt, wenn sie benötigt wird.

Delegierung den Zugriff auf Gruppen, die unmittelbare Delegierung ist dabei (genau wie beim Vertragsbaum) an nur einen Teilnehmer gerichtet. Daher sind die Gruppendelegierungen nicht erforderlich. Das Konzept des Widerrufes wird nur dann gebraucht, wenn ein falscher Teilnehmer eingeladen wurde oder es nach erfolgtem Vertragsabschluss Änderungen gibt. Entweder sind alle Autorisierungen (auch die in der Vergangenheit bereits ausgeübten) hinfällig und müssten versagt werden, oder die Delegierung eines Teiltransportes an Unterauftragnehmer soll bestehen bleiben, seine aktive Rolle beim Transport aber aufgekündigt werden. In diesem Fall ist der Widerruf nur für zukünftige Delegierungen notwendig, während seine selbst erhaltenen Sensordatenzugriffserlaubnisse gänzlich gelöscht werden sollen. Mit den zwei bereitgestellten Widerrufvarianten der Verteilten Delegierung lassen sich beide Varianten abbilden – es müsste nur sichergestellt werden, dass die noch gültigen Zertifikatsketten vor dem Widerruf beim PDP überprüft werden und somit die Richtlinien bereits erstellt sind. Es besteht daher keine Notwendigkeit, die Anwendung zu zentralisieren. Die Verteilte Delegierung hat effektiv die gleiche Mächtigkeit. Die XACML-Variante wird deswegen nicht weiter berücksichtigt.

Bei der Verwendung von SPKI gibt es nur den rückwirkenden Widerruf. Alle erstellten Delegierungen werden ebenfalls zurückgerufen. Im Fall des nur zukünftigen Widerrufs müsste der Widerrufende alle gewünschten Delegierungen des widerrufenen Teilnehmers erneut ausführen. Dies kann zwar im Einzelfall aufwändig sein, erwartungsgemäß tritt dies aber selten auf[14]. Daher sollte diesem Szenario nicht zu viel Aufmerksamkeit geschenkt werden.

Viel interessanter ist die Beleuchtung der zwei wesentlichen Unterschiede von AMANDA und SPKI in den betrachteten zwei Varianten: Wie soll das System mehr Kontrolle über die Delegierungen erlangen? Ist die Teilnehmerregistrierung gewünscht?

5.6.1 Teilnehmerregistrierung

Bei AMANDA ist eine Teilnehmerregistrierung erforderlich. Die RSOs können dabei bestimmen, nach welchen Kriterien diese abläuft. Bei SPKI kann jeder Teilnehmer außer dem organisierenden Spediteur ohne vorherige Registrierung bei der RSO teilnehmen. Die Überprüfungen der Teilnehmer und Auswahl der dabei angewendeten Kriterien werden an die Vertragspartner delegiert.

Es kann aber für die beteiligten Parteien durchaus von Interesse sein, wenn hohe Kriterien beim Registrierungsprozess angewendet werden. Versicherer möchten im Schadensfall nicht für Fehler anderer aufkommen müssen. Logistiker möchten für Transportausfälle anderer nicht gerade zu stehen haben. Ebenfalls kann für staatliche Behörden eine vorherige Überprüfung und Akkreditierung von Firmen erwünscht sein. Ob es sich um Probleme bei der Wareneinfuhr oder zur Abwehr oder Aufklärung terroristischer Aktivitäten handelt: Das Wissen um nachgewiesene Parteienzuordnung hilft den Ermittlern.

Gegen eine erforderliche Registrierung der Teilnehmer und somit gegen das Verwenden von AMANDA spricht der benötigte Durchführungsaufwand. Dieser ist eine Einstiegshürde in das System und

[14] gemäß der in 5.3 getroffenen Annahmen

das wirkt erfahrungsgemäß meist abschreckend. Es sollte der initiale Aufwand für jeden Teilnehmer minimiert werden, da genau dieser häufig die hinter den Parteien stehenden Menschen davon abhält, an einer Anwendung wie MASC teilzunehmen. Je größer der Aufwand ist, der bei einer Teilnehmerregistrierung von den Parteien gefordert ist, desto geringer wird deren Akzeptanz.

Für die Verwendung von AMANDA spricht, dass Behörden oder RSOs Prozeduren vorgeben können, die Teilnehmer erfüllen müssen, um an einem MASC-Transport teilnehmen zu dürfen. Jedoch entfernt der für jedwede Registrierungen notwendige Aufwand die Anwendung weiter vom durch SPKI besser abbildbaren Status Quo. Für die Terrorbekämpfer ist eine Registrierungspflicht Mindestvoraussetzung, für die Logistiker ist sie lästiger Zusatzaufwand.

5.6.2 Delegierungskontrolle

Die bessere Kontrolle der Delegierungen, wie mit AMANDA möglich, ist aus zwei Gründen vorteilhaft: Zum einen kann dem Shipper gewährleistet werden, dass seine Daten nur kontrolliert weitergegeben werden, wobei gezielt die Sensordaten für den Spediteur unsichtbar gemacht werden können. Zum anderen können Delegierungen verhindert werden, die aus Unachtsamkeit oder Fehlverhalten zu viele Berechtigungen weitergeben. Eine bessere Geheimhaltung der Ereignisse ist erreicht, wenn das System die Weitergabe der Autorisierungen kontrollieren kann. Der durch AMANDA eingeführte erhöhte Aufwand dezentralisiert das System. Die unkontrollierte direkte Abbildung des Status Quo mit SPKI ist die elegantere Lösung.

5.6.3 Migration

Es sei an dieser Stelle noch kurz der Migrationsaufwand betrachtet, der notwendig wird, wenn der MASC-ID von einem SPKI-basierten Ansatz auf AMANDA in der Verteilten Delegierung umgestellt werden soll. Die Datenbankstrukturen der beiden Systeme unterscheiden sich lediglich in der Struktur der beiden Tabellen, die unter *Richtlinien* in Abschnitt 5.4 zusammengefasst werden. Es müssten also die SPKI-ACL und die Widerrufliste in die AMANDA-Richtlinien und die lokale Gruppenliste geändert werden. Es wäre auch denkbar, dass erst bei der Transportanmeldung entschieden wird, ob AMANDA oder SPKI verwendet werden soll – also ein MASC-ID im Parallelbetrieb möglich ist. In dem Fall würden die Tabellen zur Datenbankgenerierung entsprechend angelegt. Für den einzelnen Transport wäre allerdings nur entweder AMANDA oder SPKI möglich. Dieses ist auch sinnvoll, da die Vorteile von AMANDA verschwinden, wenn sie durch SPKI umgangen werden können.

Bei SPKI sind PDP und PEP ein System, bei AMANDA nicht notwendigerweise. Aus Gründen der Kompatibilität wäre dies aber auch mit AMANDA umsetzbar. Bei SPKI reicht ein Teilnehmer bei jedem Zugriff seine Zertifikatskette ein, bei AMANDA nur beim ersten. Ein kompatibler Wächter wäre wie folgt aufgebaut: Ist eine Zertifikatskette angehängt, übergibt der PEP diese (mit der Identifizierung des Teilnehmers) dem PDP und der generiert die Zugriffsrichtlinie. Im AMANDA-Fall fügt er diese in die Datenbank (PIP) ein, bei SPKI verweilt sie im lokalen Speicher. In beiden Fällen extrahiert er die

zum aktuellen Zeitpunkt geltenden Zugriffsregeln für den gerade zugreifenden Teilnehmer und sendet diese an den PEP zurück. Sendet der Teilnehmer keine Zertifikatskette an den PEP (AMANDA-Fall beim wiederholten Zugriff), übergibt er nur die Teilnehmeridentifizierung an den PDP. Dieser wertet die Richtlinie auf Basis des PIP aus und extrahiert die aktuellen Zugriffsrechte. Auch diese werden dem PEP zurückgeliefert. Der PEP entscheidet den Zugriff auf Grund der Regeln, die der PDP ihm zurückgegeben hat.

Ein Parallelbetrieb sollte auf diese Weise möglich sein, was eine Migration von SPKI auf AMANDA vereinfachen sollte. Wenn AMANDA so wie hier beschrieben umgesetzt würde, wäre auch eine Migration in die andere Richtung leicht möglich. Bedingung ist, dass die Zertifikatsketten nicht gesondert an den PDP übermittelt, sondern (wie bei SPKI) beim ersten Zugriff dem PEP vorgelegt werden, der diese weitergibt.

5.6.4 Fazit

Zwei Probleme soll die MASC-Anwendung lösen: Die logistischen Abläufe sollen besser organisisert und die Transportqualität erhöht werden. Zur Entdeckung terroristischer Aktivitäten sollen die Transportkette besser durchleuchtet und Vorgänge im Container entdeckt werden. Letzteres kann nur mit AMANDA sinnvoll umgesetzt werden. Für die Verbesserung der Transportvorgänge ist AMANDA hingegen nicht erforderlich. Es obliegt dem Spediteur, eine Versicherung direkt auszuhandeln. Seine eigenen Kriterien, die er an die Auswahl der Transportkette anlegt, kann er mit dem Versicherer (und bezüglich Geheimhaltung mit dem Shipper) selber abstimmen und bei Verträgen mit Unterauftragnehmern gleichermaßen weitergeben. Dieser Ansatz ließe sich auch mit SPKI realisieren.

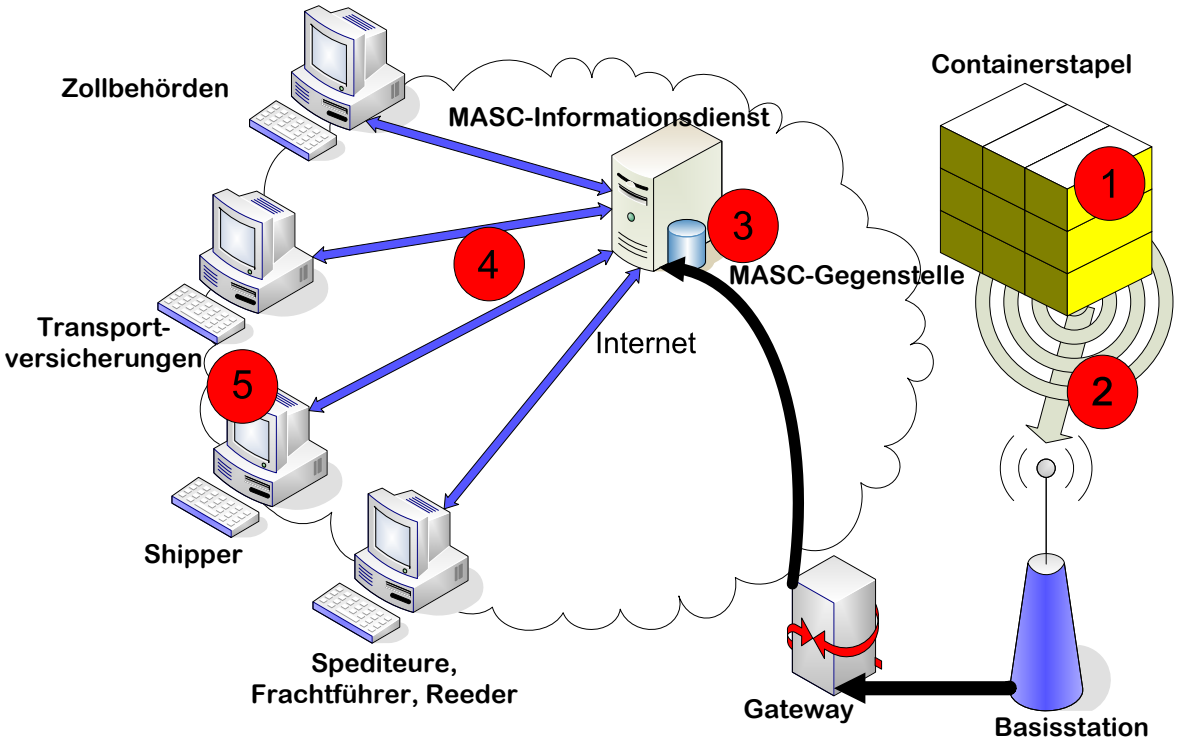
Das Argument der Einstiegshürde ist nicht zu vernachlässigen. Gerade das MASC-System ist darauf angewiesen, dass viele Teilnehmer daran mitwirken, um die Infrastrukturkosten zu reduzieren. Auf Grund dessen ist die Realisierbarkeit mit der Frage verknüpft, ob es überhaupt von Behörden akzeptiert würde. Dies würde das System interessanter machen und hätte zur Folge, dass schneller eine benötigte Teilnehmerzahl erreicht wird. Daher wäre eine Verquickung mit AMANDA unumgänglich, um den Bedürfnissen der Behörden besser gerecht zu werden.

Wenn Behörden nicht oder nur lokal einstiegen, wäre ein Hybridsystem eine interessante Variante. In Abhängigkeit von den Anforderungen des Spediteurs oder der Behörden bestimmter Länder könnte dieser zwischen einer Transportüberwachung mit automatischer Dokumentenerstellung (bei Verwendung von AMANDA) und einer einfachen Transportüberwachung (mit SPKI) wählen. So könnte die Wahl davon beeinflusst werden, ob der Transport ein Land passiert, das ein System auf AMANDA-Basis unterstützt. Dies ist möglich, da die beiden Schemata sehr ähnlich sind und somit der MASC-ID zu beiden kompatibel sein kann.

Auf die Verwendung von SPKI sollte nur zurückgegriffen werden, wenn die Anerkennung der automatisierten Dokumente nicht realisiert werden kann. Ansonsten sollte AMANDA gewählt werden.

6 Zusätzliche Sicherungsmaßnahmen

Eine Sicherheitsarchitektur ist immer nur so stark wie ihre Schwachstellen. Daher muss eine adäquate Absicherung ALLER Komponenten betrieben werden. Abschnitt 3.4 behandelte bereits diesen Sachverhalt, die dortige Abbildung sei hier noch einmal wiederholt.



Gegenmaßnahmen gegen Angriffe am Angriffspunkt 2 wurden in Kapitel 4 mit Einführung des MASC-ST besprochen. Die integere Verteilung der Zugriffsrechte (Punkt 4) wurde in Kapitel 5 mit durch den MASC-ID gelöst. Bleiben die Hardware-Sicherung der MASC-Einheit, die Absicherung der Datenübertragung des MASC-ID und die Sicherung der Klientensysteme (Punkt 5) und der Server (Punkt 3).

6.1 Zugriffsschutz der MASC-Einheit

Für das Sicherheitskonzept ist die Absicherung der MASC-Einheit von Bedeutung. Selbst wenn die gesendeten und weiterverteilten Daten gut geschützt sind, die MASC-Einheit hingegen nicht, wäre

diese das schwächste Glied der Kette. Die Sicherungsmechanismen wären somit leicht umgänglich, indem die MASC-Einheit aufgeschraubt und die Daten direkt aus dem Speicher ausgelesen werden.

Durch den Einsatz von kryptographischen Verfahren ist es möglich, die generierten Ereignisse verschlüsselt im Speicher abzulegen. Wird die Einheit geöffnet und der Speicher ausgelesen, können diese Daten nur bei bekanntem Schlüsselmaterial rekonstruiert werden. Dieses ist allerdings ebenfalls im Speicher abgelegt. Wenn der Angreifer sowohl das Schlüsselmaterial als auch die gesicherten Daten ausliest, kann er diese dechiffrieren. Daher ist es notwendig, dass die Schlüsselinformationen der Einheit vor Fremdzugriff geschützt sind und bei Erkennung eines unberechtigten Zugriffversuchs gelöscht werden.

Eine Analyse von Hardwarezugriffsmaßnahmen hat ergeben, dass die effizienteste Methode der Absicherung der MASC-Einheit die Verwendung von Antifuses mit Hardware-Ummantelung ist [102].

Bei Antifuses handelt es sich um invertierte Sicherungen. Sie brennen beim Anlegen einer Spannung durch und schaffen damit eine permanente Verbindung (anders als bei Sicherungen, die beim Durchbrennen den Kontakt abreißen lassen). Wenn beispielsweise das Gehäuse geöffnet wird, entsteht eine Spannung im Antifuse. Diese schaltet damit ihre Verbindung frei, die wiederum dafür sorgt, dass der Schlüsselspeicher unwiederbringlich gelöscht wird.

Abbildung 6.1: Epoxidharzummantelung einer Netzwerkkarte

Die zweite Sicherung ist das Eingießen in Epoxidharz. Diese Ummantelung schützt die Hardware vor Fremdzugriff. Der Angreifer muss die Harzschicht abkratzen. Das ist allerdings sehr schwierig, weil die Hardware beim Abkratzen leicht beschädigt werden kann. Im Rahmen von praktischen Tests wurde eine Netzwerkkarte (als Beispiel für beliebige Hardware) in transparentem Epoxidharz eingegossen. Abbildung 6.1 zeigt das Resultat.

Die Kombination aus Epoxidharz und Antifuses besteht darin, dass die Siliziumschicht der Antifuses direkt unter der Epoxidharzschicht liegt. Wird beim Aufkratzen der Epoxidschicht das Silizium getroffen, wird die Antifuse aktiviert und der Schlüsselspeicher zerstört.

Allerdings ist es möglich, Epoxidharz mit rauchender Salpetersäure aufzulösen. Auf diese Weise könnte die Harzschicht entfernt werden, ohne die Antifuses auszulösen. Um dieses auszuschließen, sollte die Epoxidummantelung mit einem eingewebten Gitternetz kombiniert werden. Beim Versuch, das Harz aufzulösen, berühren sich die in zwei Ebenen verlegten horizontalen und vertikalen Gitternetzparallelen. Dieser Kontakt kann leicht erkannt werden und die Antifuses ebenfalls auslösen.

Eine weitere Schwachstelle der MASC-Einheit sind die Sensorverbindungen. Diese Kabel können durchtrennt und Sensoren angeschlossen werden, die Fehlinformationen verbreiten. Dies kann nur sehr schwer ausgeschlossen werden. Daher sollten die wichtigen Sensoren in der MASC-Einheit integriert sein. Die Verbindung zu Türsensor und Sendeantenne sollte zudem baulich so gesichert werden, dass ein Angreifer nicht ohne weiteres an die Kabel herankommt.

Sicherheitsprozessoren, wie sie in Smartcards verwendet werden, sind ebenfalls für den Einsatz in Containereinheiten interessant. Sie sind speziell gegen invasive Angriffe geschützt und auf geringen Stromverbrauch optimiert. In der Smartcardforschung entwickelte Techniken und Technologien sollten bei der Prozessorwahl der Containereinheit Verwendung finden. Im Besonderen wäre es denkbar, direkt Smartcards einzusetzen, um den Schlüssel im System abzusichern. Die fingernagelgroßen SIM-Karten-ähnlichen Chips könnten initial von der RSO in die Containereinheit eingeführt werden, um sie anschließend mit Epoxydharz einzugießen (siehe oben). Die Smartcard würde auch die Schlüsselableitung ausführen, indem ihr die Nonces übergeben werden und sie die Sitzungsschlüssel berechnen kann. Auf diese Weise verließe der sichere Schlüssel niemals die Smartcard. Bei Zugriffserkennung könnte die Containereinheit die Smartcard zerstören und somit das Schlüsselmaterial auf der Einheit vernichten. Auf diese Maßnahme könnte verzichtet werden, wenn der Smartcardsicherheit so sehr vertraut wird, dass ein Schlüsselauslesen (während der gesamten Lebensdauer der MASC-Einheit) ausgeschlossen werden kann. Aus heutiger Sicht ist diese Annahme aber kaum zu erfüllen, da der Wettlauf zwischen Herstellern und Schwachstellenausnutzern von häufigen „Führungswechseln“ geprägt ist.

6.2 Datenübertragung des MASC-ID

Die Datenkommunikation zwischen den Klienten und dem MASC-ID sollte nicht im Klartext erfolgen. Weder die Anfragen (die auf eine Zuordnung zwischen Anfragenden und Transportbeteiligung zurückschließen lassen) noch deren Antworten sollten für jeden lesbar sein. Diese sind bei Sensordaten-Abfrage gleichbedeutend mit den Ereignissen, die hauptsächlich durch die MASC-Anwendung zu schützen sind. Zur Wahrung der Vertraulichkeit ist also eine Datenverschlüsselung notwendig. Zur Integritätswahrung ist es erforderlich, dass sich beide Parteien gegenseitig authentifizieren.

Für diese Anforderung gibt es Verfahren, die vielerorts verwendet werden. Im Besonderen eignet sich TLS[1] für diese Aufgabe. Wie der Name schon sagt, setzt die Transportverschlüsselung auf der im Internetprotokoll hauptsächlich genutzten Transportschicht TCP[2] auf. Es befindet sich demnach zwischen der Transportschicht und den Anwendungsprotokollen wie HTTP[3], SMTP[4] oder dem MASC-ID.

Bei TLS wird zunächst das SSL-Handshake-Protokoll durchgeführt, das eine zertifikatsbasierte Zwei-Wege-Authentifizierung sicherstellt. Nach der beidseitigen Authentifizierung einigen sich beide Parteien auf ein symmetrisches Verschlüsselungsverfahren und einen Schlüssel. Die Kommunikation von Klienten und MASC-ID wird anschließend in einen symmetrisch verschlüsselten TCP-Datenstrom eingebunden.

TLS basiert allerdings auf X.509-Zertifikaten (gemäß [40]) und die Serverauthentifizierung ist nur optional. Zur Absicherung der Datenkommunikation des MASC-ID müsste die beidseitige Authentifizierung vorgeschrieben werden und die X.509-Zertifikate durch SPKI- oder sinngemäß AMANDA-Zertifikate ersetzt werden. Wegen der durch TLS eingeführten Teilnehmerauthentifizierung muss beim Einreichen von Zertifikatsketten keine Signierung mehr erfolgen. Bei der Überprüfung muss der PDP allerdings auf die TLS-Teilnehmeridentifizierung zugreifen können, um nachzuvollziehen, ob die Zertifikatsketten ihn zum Zugriff befähigen.

Vertraulichkeit und Integrität werden mit TLS hergestellt. Die Verfügbarkeit des MASC-ID ist allerdings der wunde Punkt. Diese kann nicht sichergestellt werden, da es sich um einen internetbasierten Dienst handelt: DoS-Angriffe sind durchführbar. Deswegen ist es wichtig, dass es möglichst viele RSOs gibt, die diesen Dienst auf eine breite Serverbasis stellen. Zwar ist immer nur eine RSO für einen Transport zuständig, aber wenn der Dienst auf viele Schultern verteilt ist, kann ein Gesamtsystemausfall für einen Angreifer nur schwer realisiert werden.

6.3 Klientensysteme

Der effektivste Angriff ist vermutlich gegen kleine Shipper durchzuführen, wenn diese nur einen Computer mit einer direkten Einwahlverbindung haben und diese zur MASC-Überwachung genutzt wird. Die Angreifer müssten lediglich eine Schwachstelle im System finden, die ihnen Zugriff auf den Rechner gewährt. Mit diesem Zugriff können sie alle Ereignisse direkt aus dem System auslesen. Bei Computersystemen hinter Firewalls ist dieser Angriff schwerer, aber ebenfalls denkbar. Gleichermaßen sind die Internetserver vor Schwachstellen nicht gefeit und müssen adäquat abgesichert werden.

Diese Sicherungsmaßnahmen sind allerdings keine besonderen, sondern die gleichen, die bei aktuellen Computern und Internetdiensten reiflich getestet werden. Den besten Schutz bietet ein System,

[1] TLS steht für *Transport Layer Security* und ist der Nachfolger von SSL (*Secure Socket Layer*). [25, 78, 79]
[2] TCP ist die Abkürzung für das *Transmission Control Protocol*. [69, 44]
[3] HTTP ist die Bezeichnung für das im WWW verwendete *Hypertext Transfer Protocol*. [32]
[4] SMTP ist das *Simple Mail Transfer Protocol* und wird für das Senden von E-Mails verwendet. [47]

welches stets die neuesten Aktualisierungen und Fehlerbeseitigungen enthält und durch Firewalls zusätzlich abgesichert ist. Die Praxis lehrt aber, dass das benötigte Bewusstsein häufig nur bei größeren Firmen vorhanden ist und auch dort nicht flächendeckend. Kleinere Betriebe und Privatpersonen sind im Querschnitt zu schlecht gesichert.

Diese Arbeit kann keine Lösung für dieses Problem finden. Da normale Computersysteme für Kunden als Zugriffsschnittstelle für das MASC-System genutzt werden sollen, wird es auch überall dort existieren, wo Standard-Computersysteme eingesetzt werden. Die Server der RSOs sind genauso betroffen wie die Klientensysteme. Da sich RSOs Sicherheitsorganisationen nennen, bleibt zu hoffen, dass sie sich auch über IT-Sicherheit genügend Gedanken machen. Bei den Kunden wird aber vermutlich dieser Schwachpunkt bleiben.

Allerdings sollte Erwähnung finden, dass die Firmen, die auf eine Sicherung ihrer Daten bestehen, es selbst in der Hand haben, ihre Systeme abzusichern. Problematischer ist es, wenn Unterspediteure oder Frachtführer zum Schwachpunkt in der Kette werden. Da alles vertraglich regelbar ist, könnte der Shipper bei extrem vertraulichen Transporten Shipper darauf bestehen, dass die Spedition keinerlei Sensordaten-Leserechte weitergibt. Da der Spediteur der initiale PAP ist, also die Richtlinien erstellt und genau kontrollieren kann, welche Richtlinien er zur delegierten Erstellung weitergibt, ist dies unabhängig von der Wahl der Delegierungstechnologie möglich. Spezialisierte Hochsicherheits-Spediteure könnten dann ebenfalls für die Datensicherheit in ihren eigenen Reihen garantieren. Dieser Zusatzdienst würde für Spediteure mit besonderen Absicherungen eine neue Marktnische eröffnen. Bei Verwendung von AMANDA könnte der Spediteur überdies nicht einmal selber die Leserechte besitzen und so gefeit gegen Angriffe sein.

7 Abgrenzung zu bestehenden Systemen

Das einzige mit MASC vergleichbare System, welches sich bereits im Feldversuchsstatus befindet, ist die „Intelligent Trade Lane“ (ITL) [27] von IBM und Mærsk. ITL wird bereits aktiv beworben. Demnach wird die endgültige Produktreife vermutlich in absehbarer Zukunft erreicht werden.

7.1 Intelligent Trade Lane

Die Basis der ITL ist eine Containereinheit, die von IBM (und der ETH Zürich) entwickelt wurde [27]. Diese Containereinheit ist portabel, wird vor dem Transport an den Container gesteckt und nach dem Transport wieder entfernt. Dabei besteht die Containereinheit aus zwei mit einem Scharnier verbundenen Einheiten. Durch Aufklappen des Scharniers kann die größere Recheneinheit von der schmalen Sendeeinheit entfernt werden. Im aufgeklappten Zustand lässt sich die Containertür zwischen die Einheiten stecken und durch Zuklappen wird die Einheit an der Containertür arretiert.

Abbildung 7.1: TREC - ITL Containereinheit [27]

IBM nennt ihre Containereinheit *TREC*[1]. Abbildung 7.1 zeigt auf der linken Seite die zusammengeklappte Einheit, wobei sich die Sensorik oben und die Sendeeinheit auf der Unterseite befindet. Die befestigte Einheit ist auf der rechten Seite abgebildet. Die Sendeeinheit ist dabei sichtbar, da diese auf der Außenseite der Containertür verbleibt. Die Recheneinheit mit Sensorikansteuerung ist im Inneren des Containers verborgen.

[1] TREC (sprich: Ti-Räck) steht für „Tamper-Resistant Embedded Controller“.

Die Sendeeinheit des TREC besteht aus einer ZigBee-Einheit, einem Mobilfunkgerät und einem GPS-Transmitter. Per GPS berechnet die Einheit selbstständig ihren Aufenthaltsort. Gleichzeitig wird der GPS-Sender dazu verwendet, die Kommunikation mit der IBM-Gegenstelle aufzubauen. Dieses ist zumindest der Fall, wenn die Mobilfunkeinheit keine Verbindung aufbauen kann. [27]

Für den Fall, dass die Einheit weder Kontakt zu Satelliten noch zu einer Mobilfunkbasisstation aufbauen kann, versucht sie über ZigBee[2] andere TRECs zu erreichen und indirekt mit Hilfe von Meshing eine Verbindung herzustellen. Dieses ist eine wichtige Ergänzung, da auf Schiffen nur wenige Containerstellplätze eine GPS-Verbindung herstellen können.

Die Gegenstelle wird von IBM verwaltet. IBM ist der exklusive Dienste-Anbieter, der die Daten in seinem Rechenzentrum verwaltet und dem Shipper zur Verfügung stellt. ITL ist ein Joint-Venture zwischen IBM und Mærsk. Da der aktuelle TREC nur auf Containertüren der Firma Mærsk passt, gibt es bei ITL keine freie Wahl von Spediteur und Frachtführer.

Der bisher größte Feldversuch erfolgte zusammen mit Heineken im Biertransport von den Niederlanden in die USA. Heinekens Transportversicherung war dabei bereit, beim Einsatz dieses Systems die Versicherungssummen so weit zu senken, dass diese die Mehrkosten für den Shipper aufwiegen. Dieses ist zwar von der Versicherung, der Ware und dem Warenwert abhängig, zeigt aber, wie interessiert die Versicherungen an solchen Systemen sind. Im Besonderen ist es bei ITL möglich, dass die Container ihre eigene Position aus GPS-Daten ermitteln und bei einer falschen Weiterleitung selbst mitteilen können, wo sie sich befinden.

7.2 Vergleich zwischen MASC und ITL

MASC und ITL haben die Gemeinsamkeit, dass sie Sensoren verwenden, um die Qualität des Transports zu erhöhen, indem sie Informationen über die Frachtbedingungen preisgeben. Sie unterscheiden sich aber in der Wahl der Infrastruktur und der Kommunikation voneinander.

7.2.1 Infrastruktur

MASC benötigt eine Infrastruktur beim Containerstellplatz und umgeht damit die Notwendigkeit einer Satellitenverbindung. Der Nachteil ist, dass MASC-Einheiten nur online sein können, wenn eine Infrastruktur vorhanden ist. Besonders interessant ist die Betrachtung der beiden Verfahren, wenn erst wenige Container mit entsprechenden Einheiten ausgestattet sind. Bei MASC wären bei einer flächendeckenden Infrastruktur die Kosten pro Container extrem hoch. Es wäre vermutlich auf einige Anfangsstrecken begrenzt und würde sich langsam mit einer höheren Containerdurchsetzung entwickeln. Bei der ITL sind die Kosten pro Container konstant. Es ist allerdings notwendig, dass die Container nach Möglichkeit am Ende des Schiffes oder oben in den Stapeln platziert werden, um eine Satellitenverbindung aufbauen zu können.

[2] ZigBee ist ein stromsparendes Sendeverfahren nach IEEE 802.15.4. [37]

Je mehr überwachte Container es gibt, desto günstiger wird im MASC-Fall die nötige Infrastruktur pro Container (bei konstanter Ausbaustufe). Bei vorhandener Infrastruktur ist es egal, wo der MASC-Container in einem Containerstapel steht. Aber auch bei ITL weichen die Anforderungen an die Positionierung der TREC-ausgerüsteten Container auf, wenn viele von Ihnen vorhanden sind. Wenn nämlich in jedem Stapel mindestens ein Container eine Satellitenverbindung aufbauen kann, dann kann durch Meshing die Erreichbarkeit auf alle erweitert werden.

7.2.2 Meshing

Da die Lebenszeit der TRECs auf maximal acht Wochen begrenzt sein muss, können energieaufwändige Satellitenverbindungen überhaupt erst aufgebaut werden und die Container sind nicht auf Push-Verfahren angewiesen. Wegen dieser Grundvoraussetzung kann Meshing aus technischer Sicht zwischen den TRECs genutzt werden. Die wichtigste Bedingung beim Meshing ist aber, dass die anderen Geräte vertrauenswürdig sein müssen. Dies ist unter der Annahme der Fall, dass nur IBM die Geräte produziert. Wenn es jedoch Angreifern gelingt, gefälschte TRECs einzuschleusen oder IBM ihr Monopol bezüglich der TREC-Herstellung verliert, wäre das Meshing angreifbar.

Bei MASC sind diese beiden Eigenschaften nicht gegeben, weswegen MASC auf Meshing verzichtet. Dies ist aber wegen der Infrastruktur auch nicht erforderlich.

7.2.3 Kundenbindung

Mærsk hat das Interesse, mit dem ITL die potenziellen Shipper als Spediteur an sich zu binden. IBM will ebenfalls exklusiver Ausrüster der TRECs sein. Betriebswirtschaftliches Ziel ist das Erreichen einer Marktführerrolle. Nicht von dieser Kundenbindung profitierende staatliche Behörden haben kein Interesse daran, ein System mit bedingter Kundenbindung als Standard vorzuschreiben. ITL adressiert lediglich Kunden von teurer Ware, die eine verbesserte Frachtüberwachung wünschen.

MASC versucht keine marktpolitischen Verschiebungen der aktuellen Logistikbranche zu erzielen, sondern den Status quo zu erhalten. Jede Partei soll durch Mitkonkurrenten ersetzbar sein, sogar der Bau der MASC-Einheiten könnte auf mehrere Zulieferer verteilt werden. So besteht auch innerhalb des Systems eine Konkurrenzsituation zwischen den RSOs oder zwischen den Spediteuren, die alle das System verwenden können. Auf diese Weise kann eine effiziente Selbstkontrolle des Systems erfolgen.

7.2.4 Terrorabwehr

Das ITL-System ist nicht zur Terrorabwehr entwickelt worden und lässt sich nicht gut erweitern. Die leicht zu entfernenden Einheiten widersprechen der Anforderung einer adäquaten Containerüberwachung. Ein Terrorist könnte die Einheit abnehmen und in einen Stahlbehälter einschließen. Die Einheit könnte wegen der Abschirmung keine Alarmierung mehr aussenden. Dieser Angriff wäre sehr leicht möglich, wenn der Ausführende Zugang zum Container erhält. Des Weiteren müsste jeder Container

mit einem TREC ausgerüstet werden. Die Kosten wären dabei sehr viel höher als bei den günstigeren MASC-Einheiten.

Bei flächendeckender Einführung der Basisstationen sänken die Infrastrukturkosten pro Container stark. Die Kosten für die Einheiten würden die Infrastrukturkosten dominieren und der Kostenvergleich beschränkte sich auf Preise der Einheiten.

7.2.5 Kosten

Ohne die jeweiligen Systeme kostenmäßig genau abschätzen zu können, erlauben die beiden Geschäftsmodelle Rückschlüsse auf die Systeme, da sich verschiedene Kosten pro Containereinheit amortisierten. Da die ITL wertvolle Fracht ausrüsten will und nur hier ihren Einsatzbereich sieht, kann sie wesentlich teurer sein als die MASC-Einheit, die auch für günstige Fracht Rentabilität erzielen will. MASC möchte die sehr günstigen Transportkosten absolut wenig erhöhen und auch bei günstiger Fracht mit den entsprechend geringeren Versicherungseinsparungen mithalten können.

Für einen kostenabschätzenden Vergleich können die Systeme komponentenweise verglichen werden. Die Sensorik von MASC und ITL wird ähnlich teuer sein. Die Prozessoren der ITL müssen etwas leistungsfähiger sein, da wesentlich mehr Datenverarbeitung durchgeführt wird. Dies gilt ebenfalls für den Arbeitsspeicher. Da die Speicherdauer der Ereignisse des ITL nicht bekannt ist, kann nur die Vermutung angestellt werden, dass die Ereignisse nach dem bestätigten Versenden nicht mehr vorgehalten werden. Deswegen könnte eine MASC-Einheit mehr Langzeitspeicher anfordern.

Im Bereich der Kommunikation unterscheiden sich beide Systeme deutlich. Da die MASC-Einheit die Kosten für die Großdistanzkommunikation auf die MASC-Infrastruktur auslagert, teilen sich alle MASC-Container diese Kosten. Bei der ITL müssen diese Kosten von jedem Container selbst getragen werden. Alleine die Existenz der drei Funkmodule erhöht die Kosten für den TREC deutlich.

Kostenabschätzung Die Kosten der Einheiten sind natürlich stückzahlabhängig. Bei Grundlage der Preise, die zurzeit für standardisierte Sensornetzwerkknoten aufzuwenden sind, und deren Hochrechnung für die zwei Modelle läge ein TREC am oberen Ende des dreistelligen Eurobereichs, während die MASC-Einheit € 500 bis 600 kostete. Bezüglich des TRECs wurde diese Kostenabschätzung von einem IBM-Mitarbeiter bestätigt. Dass die MASC-Einheit durch den Einsatz wesentlich genügsamerer Hardware und den Verzicht auf Satellitenverbindung, GPS-Empfänger und Mobilfunkeinheit (bei vergleichbarer Stückzahl) günstiger als der TREC sein kann, liegt nahe.

8 Schluss

Das Design der MASC-Anwendung ist auf drei Zielkriterien abgestimmt:

1. Ein **Sicherheitskonzept** einer verteilten Anwendung ist nur effektiv wirksam, wenn es in der grundlegenden Architektur verankert ist (vgl. Abschnitt 2.2.5).

2. Eine **effektive und effiziente** Sicherheitsarchitektur muss auf die speziellen Anforderungen der Anwendung abgestimmt sein (vgl. Abschnitt 2.2).

3. Der Zielkonflikt zwischen Sicherheit und effektiver Implementierbarkeit muss stets einen **angemessenen Sicherheitslevel** gewährleisten.

Jede Abwägung dieser Arbeit versucht die optimale Erfüllung dieser drei zentralen Anforderungen zu erreichen. Eine absolute Sicherheit ist bei einem angestrebten Massenprodukt wie diesem leider nicht zu erzielen. Diese ist nicht bezahlbar und praktisch nicht existent. Es ist daher wichtig, eine angemessene Sicherheit zu gewährleisten.

Um in realen Anwendungen den Spagat zwischen einem sicheren Systementwurf und einer sowohl bezüglich Entwicklungskosten als auch Energieverbrauch kostengünstigen Umsetzung zu schaffen, müssen die Sicherheitsanforderungen sinnvoll abgestimmt werden. Eine betriebswirtschaftliche Definition ist die Folgende.

Definition 8.1 (Angemessene Sicherheit)**.**

$$\textit{Angemessene Sicherheit} \Leftrightarrow \textit{Aufwand} > \textit{Nutzen}$$

Ein System ist angemessen sicher, wenn der zu investierende Aufwand für die Angreifer größer ist als der zu erwartende Nutzen, den sie durch diesen Angriff erzielen.

Die Schwierigkeit bei dieser allgemein gefassten Definition ist die Bewertung des Aufwandes und vor allem des Nutzens. Diese Formel darf daher nicht als mathematische Gleichung aufgefasst werden, in die lediglich ein Einsetzen der Variablen erfolgen muss. Sie soll vielmehr formalisieren, welche Abhängigkeiten sich bei der Abstimmung der Sicherheitsmechanismen ergeben. Die in dieser Arbeit angeführten Bedrohungen zeigen die Vielseitigkeit der Angriffsszenarien mit völlig anderen Anforderungen und auch unterschiedlichem Nutzen für den Angreifer.

8.1 Angreifer-Nutzen in der MASC-Anwendung

Abschnitt 3.4 führt eine Sicherheitsbetrachtung der MASC-Infrastruktur mit ihren Bedrohungen durch. Exemplarisch werden drei Angreifertypen mit einem hohen Nutzen am Brechen des Systems zusammengefasst, wie sie an verschiedenen Stellen in dieser Arbeit diskutiert wurden.

Frachtführer werden als mögliche Angreifer angeführt. Ihr Ziel ist es, eine falsche Containerbeförderung zu verschleiern oder in den Zuständigkeitsbereich eines anderen Frachtführers zu verschieben. Der finanzielle Nutzen, dass für eventuell entstandene Frachtschäden nicht aufgekommen werden muss, ist abhängig von der Fracht und kann sehr groß werden.

Terroristen, die versuchen, sich Zugang zum Container zu verschaffen, könnten ein Interesse daran haben, dass die MASC-Einheit nicht mehr verfügbar ist und somit ihre Anwesenheit nicht preisgibt. Ein Angriff auf die Basisstation im Catwalk oder ein Störsignal in Containerreichweite kann dies erzielen. Die Zollbehörden könnten nicht zuordnen, ob sich der auftretende Offlinestatus auf einen technischen Defekt oder einen Angriff zurückführen lässt. Im Nachhinein würde sich zwar der Speicher der MASC-Einheit auslesen lassen, aber diese Bedrohungserkennung erfolgt unter Umständen zu spät. Der Nutzen von terroristischen Anschlägen lässt sich rational nicht ermessen. Die Vergangenheit lehrte aber, dass die verfügbaren finanziellen Mittel von Terroristen sehr hoch sein können. Auf jeden Fall handelt es sich bei Terroristen ebenfalls um sehr potente Angreifer.

Auch bei der **Industriespionage** ist der Nutzen potenzieller Angreifer groß. Zwar sind Informationen über den logistischen Warenstrom nicht ganz so geheim wie neueste wissenschaftliche Errungenschaften, doch gibt es auch hier ein Bedrohungspotenzial. Wenn noch geheime Produkte vor der Markteinführung transportiert werden, dann könnte ein Konkurrent interessiert sein, die Ware zu stehlen. Wenn aber mehrere tausend Container das Werk verlassen, ist der Angriff auf einzelne Container so aussichtslos wie die sprichwörtliche Suche der Nadel im Heuhaufen. Dies ändert sich aber, wenn der Angreifer die Frachtbeschreibung leicht auslesen kann. Darüber hinaus ist die Information über einen aufgetretenen Frachtschaden ebenfalls ein Firmengeheimnis, welches zukünftige marktstrategische Entscheidungen beeinflussen kann.

8.2 Sicherheitskonzepte

Die Sicherheitskonzepte der MASC-Architektur werden hier noch einmal zusammengefasst. Es soll dabei deutlich werden, dass die Konzepte effizient an die Bedingungen angepasst sind und die Sicherheitsanforderungen nicht vernachlässigt wurden.

Die Datenübertragung wurde durch das MASC-SecureTunnel-Protokoll (MASC-ST, Kapitel 4) abgesichert. Erfolgreiche Angriffe auf die verschlüsselten Daten sind nach jetzigem Ermessen nicht möglich. Integrität und Vertraulichkeit sind adäquat gewahrt. Der Frachtführer oder ein Mann-in-der-Mitte könnten zwar die unverschlüsselten Zusatzinformationsdaten des Gateways verändern und somit kurzfristig

Verwirrung stiften, allerdings sind diese Daten keine verlässlichen Informationen, auf denen das Sicherheitskonzept fußt. Der MASC-ST basiert wegen der geringen Ressourcen und der erforderlichen Energieeffizienz auf einem symmetrischen Verschlüsselungsverfahren. Die Verwendung eines reinen symmetrischen Verfahrens auf Basis von gemeinsamen Geheimnissen war im Besonderen nur möglich, weil es mit den RSOs bereits vertrauenswürdige Dritte in den Transportketten gibt. In der Vergangenheit sind viele Sicherheitskonzepte an der Einführung vertrauenswürdiger Dritter gescheitert, weil diese nicht existierten und deren künstliche Erzeugung nicht akzeptiert wurde[1]. Dadurch konnte Sicherheitseffizienz drastisch gesteigert werden.

Bei der Weiterverteilung der Daten lag das Hauptaugenmerk auf der Integrierbarkeit in die aktuellen Auftragsabläufe. Der MASC-Informationsdienst (MASC-ID, Kapitel 5) löst dieses durch Einführung eines delegierungsbasierten Zugriffsschemas. Weil die Systembenutzer nur über diese Schnittstelle auf das Transportüberwachungssystem zugreifen, sind Benutzerfreundlichkeit und Verständlichkeit hier besonders wichtig. Ein daraus entstandener Synergieeffekt ist, dass der MASC-ID alle Transportbeteiligten zusammenführt. Dadurch lassen sich Zusatzverfahren wie das automatisierte Erstellen der Transportdokumente durch das System leicht ergänzen. Die beiden zu Grunde liegenden Autorisierungstechnologien SPKI/SDSI und AMANDA erfüllen die Anforderung an die korrekte Authentifizierung der Parteien. Somit ist die Datenintegrität gesichert. Die Vertraulichkeit wird durch eine zusätzliche Transportverschlüsselung erreicht (siehe Abschnitt 6.2). Die Sicherheit dieses Verfahrens ist durch die Verlässlichkeit der Vertragspartner gegeben und auf die Wahl dieser hat der Auftraggeber (Shipper) direkten Einfluss.

Ein angesprochener Angriffspunkt ist die MASC-Einheit selber. Da Angreifern physischer Zugang nicht verwehrt werden kann, muss diese Einheit gegen Zugriffe gesichert werden. Diese Arbeit schlägt einen Angriffserkennungsmechanismus vor (Abschnitt 6.1). Da Angriffe nicht verhindert werden können, sollen sie erkannt und deren Auswirkung gemildert werden. Durch das Löschen der Schlüsselinformationen bei einem entdeckten Angriff bleiben Vertraulichkeit und Datenintegrität erhalten. Die Selbstzerstörung des Gerätes ist jedoch irreversibel. Die Einheit muss danach durch eine neue ausgetauscht werden. Allerdings ist der Zerstörungszeitpunkt bekannt und die vorher ermittelten Daten lassen sich rekonstruieren. Die gewählten Sicherungsmaßnahmen sind dabei ausgesprochen günstig und verteuern die Kosten für die Einheit unwesentlich. Ob die vorgeschlagene physische Zugriffssicherung ausreicht, kann wegen fehlender Expertise nicht abschließend geklärt werden.

Bei der Absicherung der Serversysteme muss auf die bekannten Techniken zurückgegriffen werden, die bei Webservern und Datenbanken bereits in jedem Rechenzentrum verwendet werden. Da die meiste Software Fehler enthält, besteht hier die Abhängigkeit von Firewalls und Gastsystemen. Gleiches gilt für die Klientensysteme (Abschnitt 6.3). Allerdings hat es der Shipper selbst in der Hand, wie er sein System absichert und welche anderen Transporteure welche Informationen einsehen dürfen.

[1] Gerade die Einführung von PKIs hat in den späten 90er Jahren viele Erwartungen geweckt. Durchgesetzt haben sich diese nur in beschränktem Umfeld (Firmen, wissenschaftliche Gruppen ...), der erwartet große Wurf blieb aus.

Die Zusammenführung der einzelnen Konzepte sichert den gesamten Informationsfluss ab. Die schlussendliche Implementierung muss diese Konzepte korrekt umsetzen und sollte keine neuen Angriffspunkte einführen. Diese Anforderung ist leider nicht trivial, da es heute kein komplexeres System gibt, welches keine Softwareschwachstellen aufweist.

8.3 Einführungsphase

Ein Problem, welches die MASC-Anwendung lösen muss, bevor sie auf den Markt kommt, ist die Überbrückung der Einführungsphase. Das Versehen aller Transportmittel und Containerstellplätze mit MASC-Infrastrukturen ist nicht ohne Weiteres möglich. Ausgangspunkt müssten Partnerschaften zwischen RSOs und Spediteuren/Frachtführern sein, die auf gezielten Transportketten die MASC-Überwachung anbieten und ausbauen. Dabei könnten einige Stellplätze auf Schiffen und ausgewählten LKWs oder Zügen das System unterstützen. Besonders die Umschlagplätze mit MASC-Architekturen zu bestücken erscheint sinnvoll, um die Übergabestellen abzudecken und parallel zu Interchanges auch eine Auswertung der Containereinheiten zu erzielen.

Es wäre überdies denkbar, mit **Handlesegeräten** zu arbeiten. Diese müssten aber online sein und als eine Art portable MASC-Infrastrukturen fungieren. Dies wäre mit einem PDA-ähnlichen System dort möglich, wo ein Mobilfunknetz besteht. Anderenfalls müsste es direkt eine Satellitenkommunikation aufbauen.

In letzter Instanz wäre es wünschenswert, dass Behörden dieses System fördern. So wie es bei der C-TPAT der Fall ist, könnten Benutzer des Systems Erleichterungen bei der Einfuhr erhalten. Alle in die USA einreisenden Schiffe müssten dann mit MASC-Architekturen ausgestattet sein. So verbreitete sich schnell eine Infrastruktur und die Kosten pro Einheit reduzierten sich schlagartig.

8.4 Offene Punkte

Bis aber eine Einführungsphase aktuell würde, müsste die technische Plattform für eine Anwendung gefunden werden. Ein Anstoßen der die Anforderungen an die Containereinheit erfüllenden Hardwareentwicklung muss erfolgen. Die Anpassung des MASC-Systems gepaart mit Feldversuchen wäre notwendig, um die letzten Fragen der technischen Realisierbarkeit zu klären.

Es ist zurzeit unklar, wie gut aus Sensorschwankungen reale Ereignisse abgebildet werden können. Dazu müsste eine Analyse mit Feldtesteinheiten durchgeführt werden. Ebenfalls ist die Fehleranfälligkeit der Sensormessungen zu analysieren und zu minimieren. Des Weiteren könnte eine Bedarfsanalyse herausfinden, ob neben den genannten weitere Sensoren angeschlossen werden sollten.

Je nach verwendetem Datenübertragungsstandard sind die Basisstationen speziell anzupassen. Die Entwicklung der Gateways müsste transportmittelspezifisch erfolgen. Eine Kooperation mit einem GSM-Betreiber sollte eingegangen werden, um die drahtlose Kommunikation auf dem Festland zu gewährleisten.

Natürlich müssten RSOs diese Anwendung unterstützen und die Server bereitstellen. Die Versicherer wären ebenfalls potenzielle Unterstützer des Systems. Diese müssten bei einer Überwachung ihre Tarife senken, um so die Profitabilität an die Shipper weiterzugeben. Der Dienst würde über eine Gebühr von den Spediteuren finanziert, die ihn ihrerseits den Shippern und Versicherern anbieten. Laut IBM ist das ITL (bei teuren Gütern) selbsttragend, da die Versicherungssummen stärker als die von ihnen erzeugten Kosten gesenkt werden. Für die MASC-Anwendung sollte ähnliches gelten. Zwar ist die durchschnittliche Fracht unter Umständen günstiger, dafür sollten die durch das MASC-System erzeugten Kosten auch geringer sein.

8.5 Fazit

Ziel dieser Arbeit ist die Erstellung einer sicheren Architektur für eine Containerüberwachung, ohne dabei die konkrete Hardware und die genaue Implementierung vorzugeben. Die Arbeit schlägt bei einigen Optimierungsfragen bewusst mehrere Varianten vor, da eine endgültige Entscheidung erst bei finaler Hardwarewahl durchgeführt werden sollte. Als Beispiel sei die nicht getroffene Auswahl des symmetrischen Verschlüsselungsverfahrens genannt. Die Arbeit tendiert zu der Empfehlung von AES, diskutiert aber auch die Verwendung von RC6 und lässt das Ergebnis bewusst offen.

Ohne eine genaue Kostenabschätzung machen zu können, scheint die Umsetzung durchaus realistisch. Abhängig von der Stückzahl sollte sich eine Containereinheit für unter € 1 000 produzieren lassen. Bei einer Lebensdauer von drei Jahren und jeweils achtwöchigen Transporten würde eine Einheit 18 Transporte begleiten. Neben den Kosten, die für die Architektur und die Bereitstellung der Dienste anzusetzen sind, wären die Kosten für die Einheit pro Transport geringer als € 100. Die Kosten pro Transportkilometer wären bei dieser Abschätzung geringer als ein €-Cent. Der Mehrwert für die beteiligten Vertragspartner sollte die Kosten damit überwiegen.

Die Arbeit hat gezeigt, dass eine angemessen sichere Containerüberwachung machbar ist. Die Tatsache, dass IBM ein ähnliches System (siehe Abschnitt 7.1) auf den Markt bringt und weitere Forschungsgruppen an vergleichbaren Anwendungen arbeiten, ist ein Indiz für die zukünftige Marktpräsenz von Containerüberwachungsanwendungen.

A Kryptographische Primitive, Methoden und Verfahren in der Informationssicherheit

A.1 Der kryptographische Hashwert

Ein wichtiger Bestandteil für fast alle Sicherheitsprotokolle ist der kryptographische Hashwert (in dieser Arbeit als Hash bezeichnet). Dieser ist ein komprimierendes Abbild fester Länge eines unbestimmt großen Urbildes. Bei MD5 ist diese Länge 128 Bit, bei SHA-1 160 Bit und bei SHA-256, wie der Name schon sagt, 256 Bit. Die erzeugende Funktion wird als Hashfunktion bezeichnet. Eine wichtige Eigenschaft dieser Funktion ist die einfache Berechenbarkeit. Im Allgemeinen sollte der Berechnungsaufwand einer Hashfunktion im Gesamtkontext der umrahmenden Anwendung zu vernachlässigen sein. Die Besonderheit von kryptographischen Hashs ist der Verlust der Ähnlichkeitsrelation und die Quasi-Zufälligkeit. Ersteres bedeutet, dass ähnliche Urbilder nicht auf ähnliche Bilder abgebildet werden. Zweiteres besagt, dass die Ausgaben von Hashfunktionen keine Entropie mehr enthalten und somit wie Zufallszahlen aussehen. Die wichtigste Eigenschaft ist aber die Nicht-Umkehrbarkeit (siehe folgende Definitionen).

Definition A.1 (Nicht-Durchführbarkeit)**.** *Mit nicht durchführbar (not feasible) ist eine realistische Betrachtung der Durchführbarkeit gemeint. Wenn ein Angriff theoretisch denkbar ist, aber extrem großen Lösungsaufwand benötigt, der außerhalb des mittelbar Schaffbaren liegt, wird dieser als* ***nicht durchführbar*** *betrachtet.*

Definition A.2 (Nicht-Umkehrbarkeit)**.** *Die Nicht-Umkehrbarkeit bezeichnet in der Kryptographie, dass es nicht durchführbar ist, von einem bestimmten Bild auf ein gültiges Urbild zu schließen.*

Eine Hashfunktion lässt sich zwar einfach berechnen, um aber vom Bild auf alle möglichen zugehörigen Urbilder zu schließen, müssen alle Urbilder ausprobiert und deren Hash berechnet werden, ehe der Umkehrschluss erfolgen kann[1]. Da der Urbildraum (im Allgemeinen) unbegrenzt ist, dauert es unendlich lange, alle möglichen Urbilder zu finden, die sich auf einen Hash abbilden lassen. Der Bildraum ist allerdings begrenzt, und so ist es möglich, in endlicher Zeit irgendein Urbild zu finden, welches auf einen Hash abgebildet wird. Je größer der Bildraum, desto aufwändiger wird dieser Angriff. *Dauert*

[1] Dieses Ausprobieren aller Möglichkeiten nennt man Brute-Force-Angriff.

eine Hashfunktionsberechnung beispielsweise eine Millisekunde[2], benötigt das sequenzielle Durchprobieren von 2^{160} *Varianten (alle möglichen Variationen des Bildraumes bei 160 Bit) ungefähr* $4,6 * 10^{35}$ *Jahrhunderte.*

Es gibt nach dem *Handbook of Applied Cryptography* (HAC) [60] drei Klassen, in die Hashfunktionen eingeteilt werden:

- **Preimage-Resistance** Es ist nicht durchführbar, für einen vorausbestimmten Bildwert irgendeinen Urbildwert zu finden, der auf diesen Bildwert abgebildet wird: Bei gegebenem y ist es nicht durchführbar, einen Wert x zu finden, für den $h(x) = y$ gilt. (Hashfunktion sei h.)
- **2nd-Preimage-Resistance** Es ist nicht durchführbar, ein beliebiges zweites Urbild zu finden, welches auf den gleichen Hash abgebildet wird, als irgendein gegebenes Urbild: Bei einem gegebenen x ist es nicht durchführbar, ein 2nd-Preimage x' zu finden, so dass gilt: $x \neq x'$, $h(x) = h(x')$.
- **Collision-Resistance** Es ist nicht durchführbar, zwei beliebige, unterschiedliche Urbilder x,x' zu finden, die auf denselben Bildwert abgebildet werden $h(x) = h(x')$. Diese Anforderung unterscheidet sich von der 2nd-Preimage-Resistance in der Form, dass hier beide Urbildwerte frei wählbar sind.

A.2 MAC – Message Authentication Code

Die schlüsselbasierte Hashfunktion wird auch als *Message Authentication Code* (MAC) bezeichnet. Ein MAC ist dabei eine Einwegfunktion in Abhängigkeit eines Schlüssels. Es wird das Urbild x mit Schlüssel K auf das Bild y abgebildet.

Definition A.3 (MAC-Funktion)**.** *Eine MAC-Funktion* $h_K(x)$ *ist eine Funktion mit zwei Parametern. K bezeichnet den Schlüssel und x eine Nachricht. Bei Kenntnis der Nachricht x ist es dabei unmöglich, auf den Schlüssel K zu schließen. Nur wer in Kenntnis beider Parameter ist, kann die Funktion erneut ausführen und so deren Richtigkeit überprüfen.*

Es gibt zwei gebräuchliche Arten, MACs zu berechnen. Eine Variante ist der CBC-MAC: Hierbei wird das Bild bei Verwendung eines symmetrischen Verschlüsselungsalgorithmus (siehe A.3.1) mit dem Schlüssel K per CBC (siehe Anhang A.7) chiffriert. Der Ausgang des letzten CBC-Schrittes wird dann als MAC-Wert verwendet. Wegen der höheren Geschwindigkeit werden jedoch hauptsächlich Hashfunktionen als MAC-Basis verwendet. Die so genannten HMACs basieren darauf, dass der Schlüssel vor der MAC-Berechnung per spezieller Konkatenation (vgl. Definition A.4) mit dem Eingangswert verrechnet und das Ergebnis per Hashfunktion abgebildet wird.

[2]Die Berechnungszeit ist natürlich prozessorabhängig. Auf Smartcards benötigt SHA-1 gemäß [71, Tabelle 25] ungefähr 1 000 Zyklen pro Byte. Bei einem Prozessortakt von 100 MHz entspräche dies einer Ausführungsdauer von 100 Byte Eingabetext pro Millisekunde. Auf den ScatterWeb-Sensorknoten dauert MD5 über 16 Byte Testdaten $1,76\ ms$ [80].

Definition A.4 (HMAC). *$h_K(x)$ sei die MAC-Funktion, $h(x)$ die Hashfunktion. p_1 und p_2 sind vordefinierte, verschiedene Bitmuster ausreichender Länge, um den Eingang der Hashfunktion auf Blocklänge zu verlängern:*

$$h_K(x) = h(K|p_1|h(K|p_2|x))$$

Haben zwei Parteien einen geheimen (niemanden anders bekannten) Schlüssel, können sie eine Nachricht authentisieren, indem sie diese dem anderen zukommen lassen und den MAC unter Verwendung des geheimen Schlüssels anhängen. Nur wer den Schlüssel kennt, kann bei erhaltener Nachricht diese MAC-Berechnung nachvollziehen. Auf diese Weise kann eine Partei ihrem Gegenüber beweisen, dass sie in Besitz des Schlüssels ist, ohne diesen aufzudecken. Glaubt der Gegenüber dem Sender, dass er diesen Schlüssel nicht weitergegeben hat, ist die Authentifizierung erfolgt.

A.3 Verschlüsselung

In der Kryptographie existieren zwei grundsätzliche Arten der Verschlüsselung: Zum einen gibt es die symmetrische Verschlüsselung. Sie basiert auf der Existenz eines geheimen Schlüssels, den zwei oder mehrere Parteien teilen. Die Daten werden mit diesem Schlüssel verschlüsselt. Jeder, der im Besitz des Schlüssels ist, kann die Daten wieder entschlüsseln (Symmetrie zwischen Verschlüsselungs- und Entschlüsselungsschlüssel). Zum anderen existiert die asymmetrische Verschlüsselung. Hierbei unterscheiden sich die Schlüssel für das Chiffrieren und Dechiffrieren. Mit dem öffentlichen Schlüssel eines Teilnehmers verschlüsselte Nachrichten können nur mit Hilfe seines privaten entschlüsselt werden. Wird einer Benutzergemeinde der öffentliche Schlüssel bekannt gemacht, können alle Mitglieder Nachrichten verschlüsseln, die bei Geheimhaltung des privaten Schlüssels nur von dessen Besitzer entschlüsselt werden können.

A.3.1 Die symmetrische Verschlüsselung

Einfachstes Beispiel für eine symmetrische Verschlüsselung ist das Anwenden der Entweder-Oder-Funktion. Der binäre Klartext (Nachricht) sei in diesem Fall N, der Schlüssel sei K. Die Verschlüsselung der Daten ist $N \oplus K$, wobei $\oplus$ für die Entweder-Oder-Operation steht.

$$N = 00101011$$

$$K = 11100100$$

$$C = N \oplus K = 11101111 \tag{A.1}$$

Das Chiffrat C gibt ohne Kenntnis des Schlüssels K keine Information über den Klartext preis. Dies ändert sich, wenn der Schlüssel bekannt ist. Es ist dann ein Leichtes, den Klartext wieder zu dechiffrieren:

$$C = 11101111$$

$$K = 11100100$$

$$N = C \oplus K = 00101011 \tag{A.2}$$

In diesem Beispiel sind Verschlüsselungs- und Entschlüsselungsfunktion gleich. In beiden Fällen ist es ein Entweder-Oder mit dem Schlüssel. Verallgemeinern lässt sich diese Symmetrieeigenschaft bezüglich des Chiffrierverfahrens jedoch nicht. Der symmetrische Schlüssel wird **geheimer Schlüssel** genannt.

Zwei wissenschaftliche Tests haben sich in den vergangenen Jahren intensiv mit der Sicherheit von Blockchiffren befasst. Die AES-Challenge [61] kürte im Rahmen eines Standardisierungswettbewerbs den Rijndael-Algorithmus zum Sieger, der seitdem als AES geführt wird. Eine europäische Initiative namhafter Kryptographen hat eine Sicherheits- [70] und Geschwindigkeitsanalyse [71] der bekanntesten kryptographischen Primitive durchgeführt.

A.3.2 Die asymmetrische Verschlüsselung

Ein mächtiges Konzept ist die asymmetrische Verschlüsselung. Bei ihr gibt es zwei Arten von Schlüsseln, den **privaten Schlüssel** und den **öffentlichen Schlüssel**. Wenn A an B eine Nachricht schicken will, die nur B entschlüsseln kann, dann verschlüsselt A die Nachricht mit dem öffentlichen Schlüssel von B. Entschlüsselt werden kann die Nachricht nur mit dem privaten Schlüssel von B und dieser sollte nur B bekannt sein.

Die Funktion f sei eine asymmetrische Verschlüsselungsfunktion. Sie wird in zwei Modi betrieben, f_V sei dabei der Verschlüsselungs- und f_E der Entschlüsselungsmodus. Das Schlüsselpaar von B sei (K_{PUB}, K_{PRV}). Das Chiffrat C wird mit dem öffentlichen Schlüssel K_{PUB} erzeugt. Der Klartext N kann mit f_V und K_{PRV} aus C wieder gewonnen werden:

$$C = f_V(N, K_{PUB})$$

$$N = f_E(C, K_{PRV}) \tag{A.3}$$

Das bekannteste Verfahren für die asymmetrische Verschlüsselung ist RSA. Allerdings muss die Umsetzung mit Vorsicht durchgeführt werden, weil es viele Angriffe gegen fehlerhafte Implementierungen gibt. Das Verfahren als solches kann jedoch als sicher betrachtet werden.

Im Gegensatz zur symmetrischen Verschlüsselung kann derselbe öffentliche Schlüssel einer ganzen Benutzerschaft zugänglich gemacht werden, ohne dass die damit erzeugten Nachrichten von den Nutzern

entschlüsselt werden können. Während bei der symmetrischen Verschlüsselung ein Schlüssel paar-(oder gruppen-)weise die Kommunikation in beide (alle) Richtungen absichert, kann bei der symmetrischen Verschlüsselung ein Schlüsselpaar verwendet werden, um jede Kommunikation zu einem Teilnehmer abzusichern. Um eine Antwort verschlüsselt senden zu können, wird jedoch ein weiteres Schlüsselpaar benötigt.

A.3.3 Vergleich zwischen asymmetrischen und symmetrischen Verfahren

Arjen Lenstra beschreibt in seinem Papier [57] die relative Geschwindigkeit von RSA bezüglich AES, sprich um wie viel langsamer sich RSA gegenüber AES bei gleicher Sicherheit verhält. Bei der Verschlüsselung gibt Lenstra den Faktor 2200–2900 an, für EC-RSA (RSA unter Verwendung von elliptischen Kurven) immerhin noch 960. Für die Entschlüsselung ist der geringste Wert 870–1200, für EC-RSA 490. Tests mit dem Motorola MSP430 Prozessor haben ergeben, dass sich der Stromverbrauch linear zur Laufzeit des Algorithmus verhält (siehe Abschnitt 4.5.1). Dies kann zwar bei anderen Prozessoren etwas variieren, dennoch ist diese Aussage als Annäherung richtig. Damit ist deutlich, wie viel komplexer asymmetrische Verfahren gegenüber den symmetrischen Pendants sind.

A.3.4 Die digitale Signatur

Die Verschlüsselung von Nachrichten sorgt für Vertraulichkeit, die digitale Signatur für Echtheit/Integrität. Mit Hilfe einer digitalen Unterschrift kann der Absender einer Nachricht deren Echtheit bestätigen. Eine digital signierte Nachricht kann ausschließlich von einer Partei erstellt worden sein. Es darf nicht möglich sein, bei gleicher Signatur die Nachricht zu ändern.

Die digitale Signatur ist ebenfalls eine asymmetrische Technik, denn es gibt zwei verschiedene Schlüssel für die beiden Arbeitsschritte: Um die Signatur zu erstellen, wird der **Signaturschlüssel** benötigt, zur Überprüfung der Signatur der **Verifizierungsschlüssel**. Der Verifizierungsschlüssel sollte dem Empfänger zur Verfügung gestellt, der Signaturschlüssel hingegen niemals preisgegeben werden.

Der Nutzer kann im Folgenden eine Nachricht verschicken und diese digital signieren. Jeder, der diese Signatur überprüfen möchte, benötigt dafür den Verifizierungsschlüssel des Ausstellers.

Die digitale Signatur und die MAC-Funktion (vgl. Anhang A.2) stehen zueinander wie die asymmetrische und die symmetrische Verschlüsselung. Beide dienen dem gleichen Zweck, sind aber konzeptionell artverschieden. Sowohl die MAC-Funktion als auch die digitale Signatur sorgen für eine Teilnehmer- oder Nachrichtenauthentisierung. Bei der MAC-Funktion wird aber eine symmetrische Technik genutzt, weswegen sich Sender und Empfänger einen geheimen Schlüssel teilen müssen. Bei der digitalen Signatur kann der Verifizierungsschlüssel einer Benutzergruppe mitgeteilt und eine Nachricht von jedem in dieser Gruppe authentifiziert werden, ohne dass sie paarweise mit einem MAC versehen werden muss.

A.3.5 Zusammenfassung

Bei der symmetrischen Kryptographie müssen beide (alle) Kommunikationspartner ein Interesse daran haben, das Geheimnis nicht preis zu geben. Es ist dabei nicht möglich, im Nachhinein nachzuweisen, welcher Gesprächspartner welche Nachricht gesendet hat. In beide (alle) Richtungen wird die gleiche Verschlüsselung verwendet. Bei der asymmetrischen Kryptographie hat jeder sein eigenes Schlüsselpaar. Es ist also auch eine vertrauliche Punkt-zu-Punkt-Kommunikation zwischen mehr als zwei Parteien möglich und im Besonderen eine Kommunikation zwischen unbekannten (sich nicht vertrauenden) Gesprächspartnern.

Wenn die Nachrichten nicht verschlüsselt, sondern lediglich die Herkunft der Nachrichten nachgewiesen (authentisiert) werden soll, gibt es bei der symmetrischen Kryptographie die Möglichkeit, einen MAC hinzuzufügen. Dieser weist nach, dass der Sender im Besitz des geheimen Schlüssels ist. Bei der asymmetrischen Kryptographie ist das Äquivalent die digitale Signatur. Auch hier ist der Vorteil, dass diese nicht nur für den jeweils anderen Schlüsselinhaber gilt, sondern auch in der Eins-zu-viele-Kommunikation funktioniert.

Die asymmetrische Kryptographie bietet zahlreiche Möglichkeiten und ist meistens das bessere Konzept, wenn es viele Parteien gibt, die unabhängig miteinander kommunizieren wollen. Allerdings ist es auch wesentlich rechenaufwändiger und die Schlüssellängen für vergleichbar sichere Verschlüsselungen sind ungefähr um den Faktor zehn größer.

Ein Problem bei der Verwendung von Verschlüsselungen oder Signaturen ist das Binden eines Schlüssels an Entitäten. Bei symmetrischen Schlüsseln einigen sich zwei Parteien auf einen geheimen Schlüssel. Hierbei ist ein Vertrauensverhältnis notwendig, welches beide dazu bringt, diesen nicht aufzudecken. Der Schlüsselaustausch geschieht meist im Rahmen dieser Vertrautheit und sichert die Bindung des Schlüssels an Entitäten ab. Asymmetrische Techniken werden wegen ihres Konzeptes meistens eingesetzt, um unbekannte Parteien oder Entitäten miteinander zu verbinden. Jeder Kommunikationspartner benötigt dabei den öffentlichen Schlüssel beziehungsweise den Verifizierungsschlüssel des jeweils anderen. Um die Echtheit der Schlüssel nachzuweisen, werden meist vertrauenswürdige Dritte eingeführt, die diese Aufgabe übernehmen. Der nachfolgende Abschnitt beschreibt, wie dieses mit Hilfe von PKIs abläuft.

A.4 Public-Key-Infrastructure

Die Public-Key-Infrastructure ist dem Namen nach eine Infrastruktur, die Public-Key-Cryptography (PKC) ermöglicht. Diese PKC ist der englische Begriff für Verfahren auf Basis asymmetrischer Kryptographie, bei denen jeder Teilnehmer einen privaten und einen öffentlichen Schlüssel (engl. *public key*) beziehungsweise einen Signatur- und Verifizierungsschlüssel besitzt.

Die große Schwierigkeit bei Verwendung asymmetrischer Kryptographie ist, dem Gesprächspartner seinen öffentlichen Schlüssel (oder Verifizierungsschlüssel) beglaubigt mitzuteilen. Dies ist im Beson-

deren problematisch, wenn sich Sender und Empfänger nicht initial kennen. Das Problem ist vergleichbar mit der Telefonnummern-Personen-Zuordnung. Wenn sich eine Person potenziell als eine andere ausgibt und dem Gegenüber seine Telefonnummer mitteilt, kann dieser auch nicht sicher sein, ob diese Angabe korrekt ist.

Beim klassischen Telefonbuch ist der Herausgeber ein Telekommunikationskonzern. Diesem kann zwar nicht grundsätzlich vertraut werden, doch bezüglich der Richtigkeit von Telefonbüchern besteht im Regelfall ein Vertrauensverhältnis zwischen Benutzern und Herausgebern. Gleiches ist notwendig, wenn öffentliche Schlüssel dem Gegenüber mitgeteilt werden sollen. Es muss einen oder mehrere vertrauenswürdige Dritte geben, die die Bindung von öffentlichen Schlüsseln an Teilnehmer zertifizieren. Mit Zertifikaten bekundet eine Partei, dass einem Teilnehmer x der öffentliche Schlüssel/Verifizierungsschlüssel y gehört, und signiert diesen. Allerdings muss dazu der Verifizierungsschlüssel des Zertifizierenden bekannt sein. Dieser muss auf einem anderen vertrauten Weg zu den Teilnehmern gelangen.

Die vertrauenswürdigen Dritten können dabei verschiedene Rollen übernehmen. Bei der weitverbreiteten X.509-Zertifikatsstruktur sind vertrauenswürdige Dritte eigene Parteien, die ausschließlich diese Rolle einnehmen. Dieses Alleinstellungsmerkmal wird als Vertrauensbasis verwendet. Bei PGP können andere Teilnehmer (zum Beispiel Bekannte) diese Vertrauensstellungen einnehmen und als Dritte, selbst vertraute Teilnehmer zertifizieren. Auf diese Weise entstehen transitive Vertrauensrelationen („Ich kenne jemanden, der kennt wen, der kennt . . . “).

Bei X.509 stellen die Zertifizierungsstellen Zertifikate aus, nachdem sie die Identität der Teilnehmer (gemäß ihrer Richtlinien) überprüft haben. Als Aussteller signieren sie das Dokument digital, so dass jeder Benutzer das Zertifikat auf seinen Aussteller hin untersuchen kann. Die Benutzer benötigen dafür noch die Zertifikate der Herausgeber. Bei modernen Betriebssystemen sind die renommiertesten bereits im System gespeichert. Für weitere Herausgeber muss der Schlüssel initial dem eigenen Schlüsselspeicher hinzugefügt werden.

Bei SPKI (siehe Abschnitt 5.3.1) gibt es keine Identitätszertifikate, die einen öffentlichen Schlüssel einer Entität zuordnen. Deswegen ist SPKI keine PKI im klassischen Sinn. Allerdings sind viele Anwendungen, für die PKIs verwendet werden, auch mit SPKI lösbar, weswegen es oft PKIs ersetzen kann.

A.5 Kerberos

Bei Kerberos handelt es sich nicht um ein kryptographisches Primitiv, sondern um eine Zusammensetzung aus verschiedenen Primitiven. Kerberos ist eine weitverbreitete Methode, um Zugriffskontrolle in größeren Netzwerken zu betreiben.

Genau genommen ist es nichts anderes als ein verteilter Authentifizierungsdienst, der oberhalb der Transportschicht angesiedelt ist. Er baut also auf TCP- oder UDP-Nachrichten auf. Es ist eine Erweite-

rung des Needham-Schroeder-Protokolls, welches im Jahr 1978 entworfen wurde. Weil der Vorgänger für Replay-Angriffe empfänglich ist, wurde Kerberos entwickelt, welches diese Schwachstelle nicht mehr enthält. Die aktuelle Version ist Kerberos 5 nach RFC 4120 [62].

Die Basis von Kerberos ist ein Server, der zentral die Autorisierung für die dezentralen Dienste regelt (Single-Sign-On). Beispiele sind Firmen- oder Hochschulnetzwerke, bei denen ein Server als Authentifizierungsserver auserkoren wird. Der Benutzer meldet sich bei dem Server an und authentifiziert sich. Die Aufgabe des Kerberos-Servers besteht darin, die einmalige Authentifizierung für andere Dienste im System gültig zu machen. Die mythologische Anlehnung an den Höllenhund, der den Eingang zur Unterwelt bewacht und den Zugang regelt, passt also.

Der Benutzer möchte einen Serverdienst nutzen, wobei der Server keine Nutzerdatenbank kennt. Also meldet er sich mit einem Nonce n_A beim Kerberos-Server an und teilt ihm mit, wer er ist und auf was er zugreifen möchte. Er erhält ein Ticket (T_B) für den gewünschten Serverdienst (zum Beispiel für den Druckerserver B). Mit diesem Ticket kann er sich im Folgenden beim Drucker anmelden und diesen für einen Auftrag im gewissen Zeitintervall nutzen.

Dies funktioniert folgendermaßen: Der Kerberos-Server generiert den Schlüssel K_{SA} und verschlüsselt diesen zum einen mit dem Schlüssel K_{BA}, den nur er und der Benutzer teilen, zum anderen mit dem Schlüssel K_{SB}, den nur Kerberos-Server und Diensteserver kennen. Zweiteres ist das Ticket und kann weitere Metainformationen bezüglich der Rechte des Benutzers beinhalten. Der Benutzer zeigt beim Anmelden am Server das Ticket vor. Der Server entschlüsselt das Ticket und extrahiert die Benutzerrechte und den geheimen Kommunikationsschlüssel. Da auch der Benutzer diesen erhalten hat, können beide jetzt mit diesem Schlüssel kommunizieren. Der Diensteserver antwortet mit dem extrahierten Zeitstempel und weist dem Benutzer nach, dass er den Kommunikationsschlüssel korrekt erhalten hat, wenn der Benutzer diesen entschlüsseln kann. Abbildung A.1 stellt den Ablauf von Kerberos schematisch dar.

Bei Kerberos übernimmt also ein Server im Netzwerk die Authentifizierung und die Autorisierung. Durch die Verwendung von symmetrisch verschlüsselten Tickets kommuniziert der Benutzer nur unter Verwendung symmetrischer Kryptographie mit den Serverdiensten. Die Basis des Verfahrens ist dabei, dass jede Partei einen geheimen Schlüssel initial mit dem Kerberos-Server teilt.

A.6 Schlüsseletablierung

Definition A.5 (Schlüsseletablierung)**.** *Eine Schlüsseletablierung ist ein Prozess, bei dem ein Shared Secret für zwei oder mehr Parteien für nachfolgenden Gebrauch generiert wird.* [60, nach Definition 12.2]

Definition A.6 (Dynamische Schlüsseletablierung)**.** *Bei der dynamischen Schlüsseletablierung wird ein dynamischer Sitzungsschlüssel vor jeder Sitzung generiert. Im Regelfall wird dieser aus einem Shared Secret errechnet.*[60, siehe Definition 12.5]

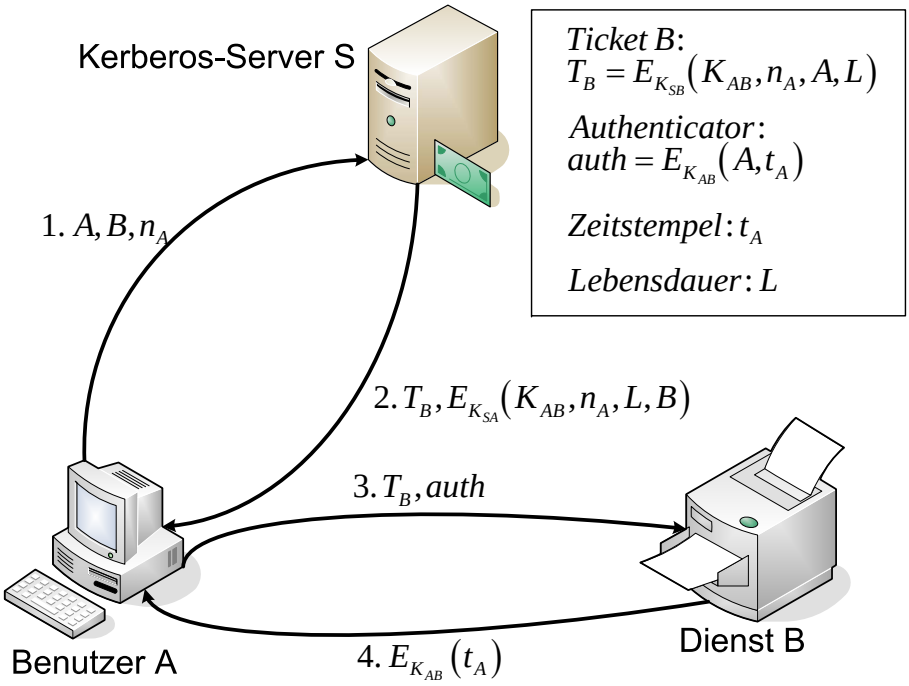


Abbildung A.1: Kerberos

Dieses Schlüsseletablierungsverfahren soll gleichzeitig für eine Authentifizierung sorgen. Die erforderliche Schlüsselauthentifizierungseigenschaft wird folgend definiert:

Definition A.7 (Schlüsselauthentifizierung)**.** *Schlüsselauthentifizierung nennt man die Eigenschaft, bei der eine Partei sicherstellen kann, dass niemand außer der zu identifizierenden Partei im Besitz des gemeinsamen Schlüssels ist.* [60, nach Definition 12.6]

Definition A.8 (Dynamisches und authentifiziertes Schlüsseletablierungsprotokoll)**.** *Ein dynamisches und authentifiziertes Schlüsseletablierungsprotokoll ist ein Protokoll, bei dem die vorigen drei Definitionen gelten.*

Durch ein dynamisches und authentifiziertes Schlüsseletablierungsprotokoll einigen sich zwei Parteien, die ein Shared-Secret besitzen, auf einen Sitzungsschlüssel. Dabei wird das gemeinsame Geheimnis verwendet, um sicherzustellen, dass es sich bei der jeweils anderen Seite auch um den Partner handelt, der das Shared-Secret mitbesitzt.

A.7 Operationsmodi

Bei Blockchiffren wird der Klartext zunächst in Klartextblöcke aufgeteilt und dann einzeln verschlüsselt. Um die Eigenschaft zu erzielen, dass nicht gleiche Blöcke auf gleiche Chiffrate abgebildet werden, werden Operationsmodi verwendet. Diese fügen Informationen des vorherigen Blocks in den nachfolgenden Block hinzu und verändern so dessen Urbild. Da der Empfänger den Vorblock kennt, kann er

diesen leicht wieder herausrechnen. Neben dem ECB (siehe folgende Definition) gibt es vier weitverbreitete Möglichkeiten, dieses durchzuführen.

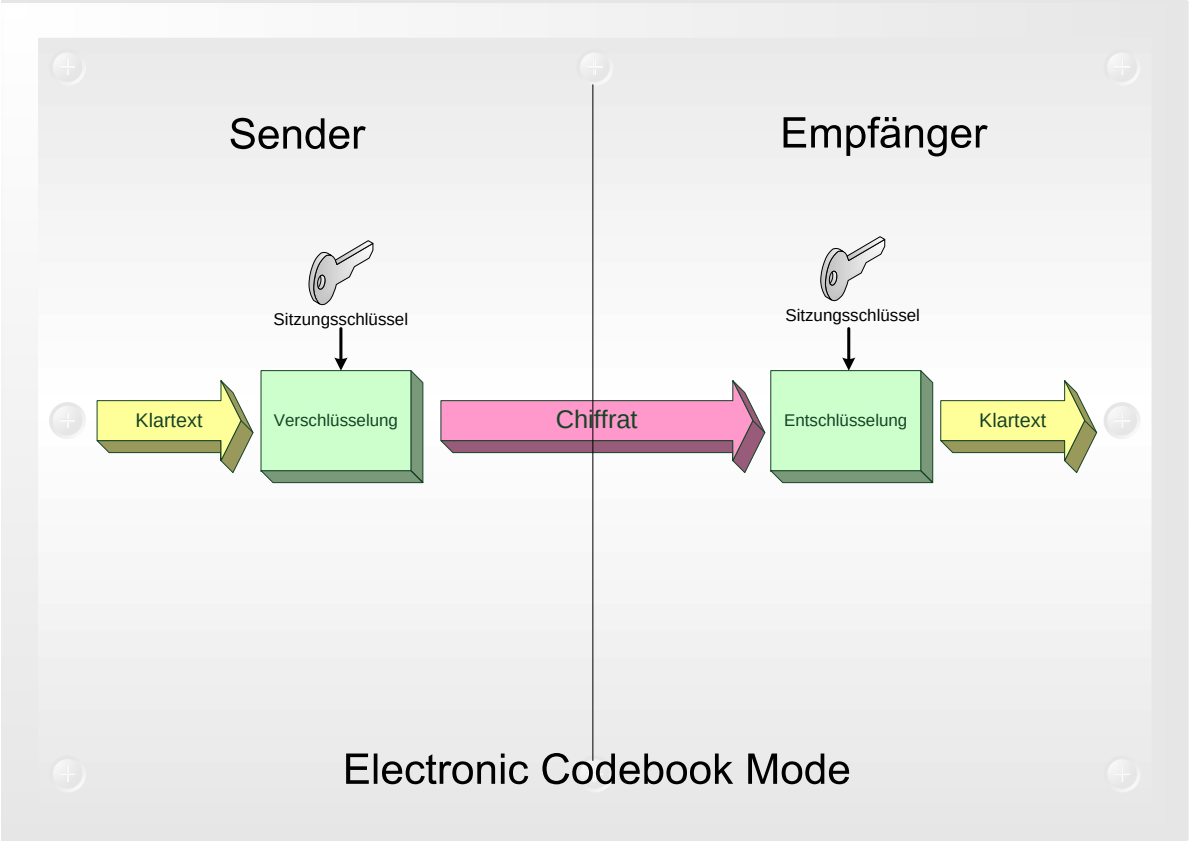


Abbildung A.2: Electronic Codebook Mode (ECB)

Definition A.9 (ECB-Operationsmodus)**.** *Der „Electronic Codebook Mode“ (ECB) wandelt (bei gleichem Schlüssel) identische Klartextblöcke in identische Chiffrate um.* [60, nach 7.11]

Beim elektronischen Codebuch-Modus (ECB) wird keine Veränderung der Codeblöcke vorgenommen. Dieser Operationsmodus bildet gleiche Klartextblöcke immer auf gleiche Chiffrate ab. Keine Information des vorherigen Blocks wird zur Verschleierung genutzt. Jeder Block ist damit komplett unabhängig. Übermittlungsfehler wirken sich jeweils nur auf den aktuellen Block aus.

Definition A.10 (CBC-Operationsmodus)**.** *Der „Cipher-Block Chaining Mode“ (CBC) verwendet einen Initialisierungsvektor (IV), der vor der Verschlüsselung mit dem ersten Klartextblock Entweder-Oder-verknüpft wird. Jeder folgende Klartext wird mit dem vorherigen Chiffrat verknüpft.* [60, nach 7.13]

Bei der Chiffrat-Verkettung (CBC) wird also das vorherige Chiffrat immer auf den Klartext addiert. Der Empfänger muss demnach das vorherige Chiffrat eine Nachricht lang speichern, um ihn nach dem Entschlüsseln vom Resultat wieder abzuziehen. Dadurch besteht eine Abhängigkeit von jedem gesendeten Block zu seinem Vorgänger. Ein Übertragungsfehler wirkt sich also immer auf den aktuellen und den

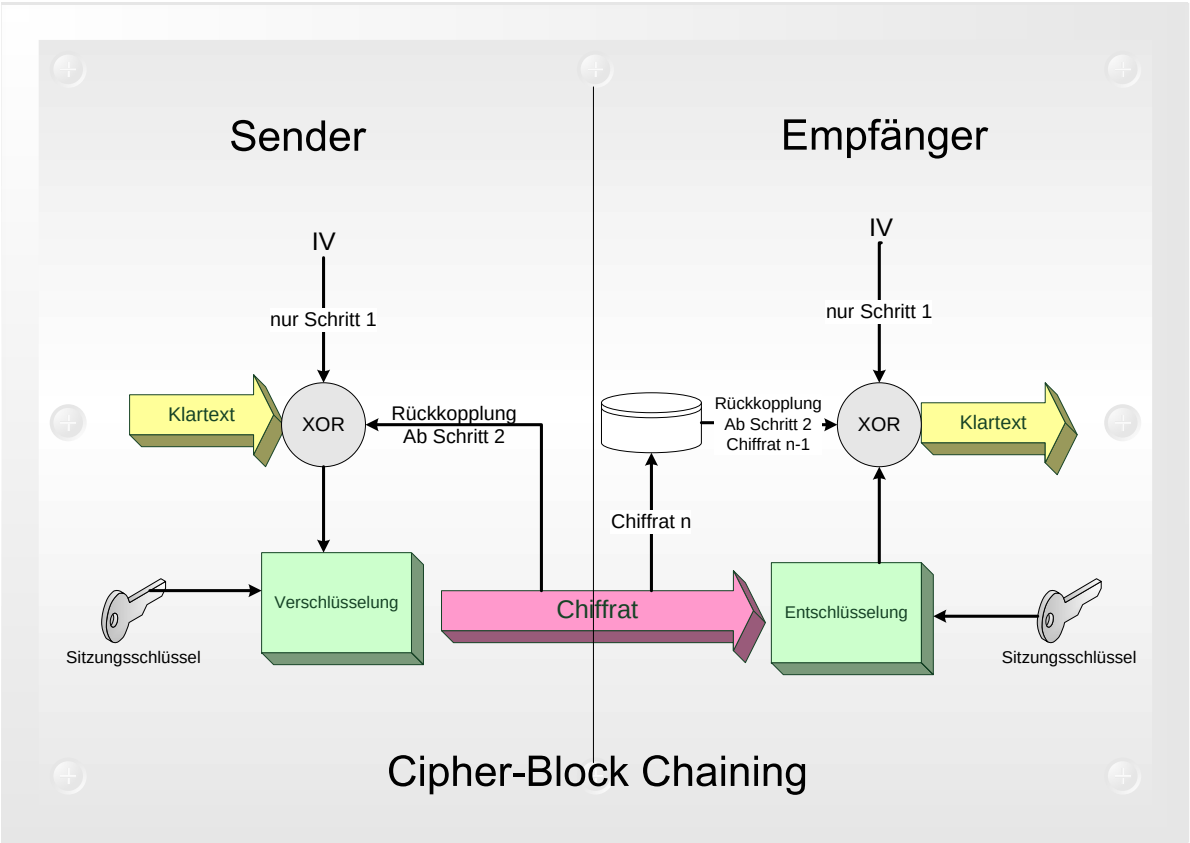


Abbildung A.3: Cipher Block Chaining Mode (CBC)

nachfolgenden Block aus. Wenn aber beim Senden des nachfolgenden Blocks kein Übertragungsfehler geschieht, steht für dessen Nachfolger das korrekte Chiffrat wieder bereit. So ist die Fehlerfortpflanzung begrenzt. Da beim ersten Paket noch kein Vorgänger-Chiffrat zur Verfügung steht, wird ein IV generiert, der zufällige Werte enthält. Dieser muss allerdings gesondert übertragen werden. Im MASC-Szenario gibt es aber bereits sitzungsabhängige Nonces, die beim Tunnelaufbau generiert wurden. Diese könnten für diesen Fall genutzt werden – es entstünde also keine Nachrichtenexpansion.

Definition A.11 (CFB-Operationsmodus)**.** *Beim „Cipher Feedback Mode“ (CFB) wird im ersten Schritt der IV verschlüsselt und anschließend mit dem ersten Klartextblock Entweder-Oder-verknüpft. Dieser verknüpfte Block ist das erste Chiffrat. Dieses wird im zweiten Schritt erneut verschlüsselt und der zweite Klartextblock per Entweder-Oder-Relation damit verknüpft und geschickt.* [60, nach 7.17]

Der Vorteil des rückgekoppelten Chiffrat-Modus (CFB) ist, dass Empfänger und Sender die identische Funktion verwenden. Der Empfänger verschlüsselt genauso wie der Sender das vorherige Chiffrat, zieht das Ergebnis vom aktuellen Chiffrat ab und erhält so den Klartext. Zum Entschlüsseln wird lediglich die Verschlüsselungsfunktion der Blockchiffre benötigt. Bei der Fehlerfortpflanzung verhält es sich wie beim CBC. Ein Bit-Fehler wirkt sich also nur auf den aktuellen und den folgenden Block aus. Beim CFB ist das Verwenden von Padding zwingend notwendig, was für CBC und ECB gleichermaßen gilt.

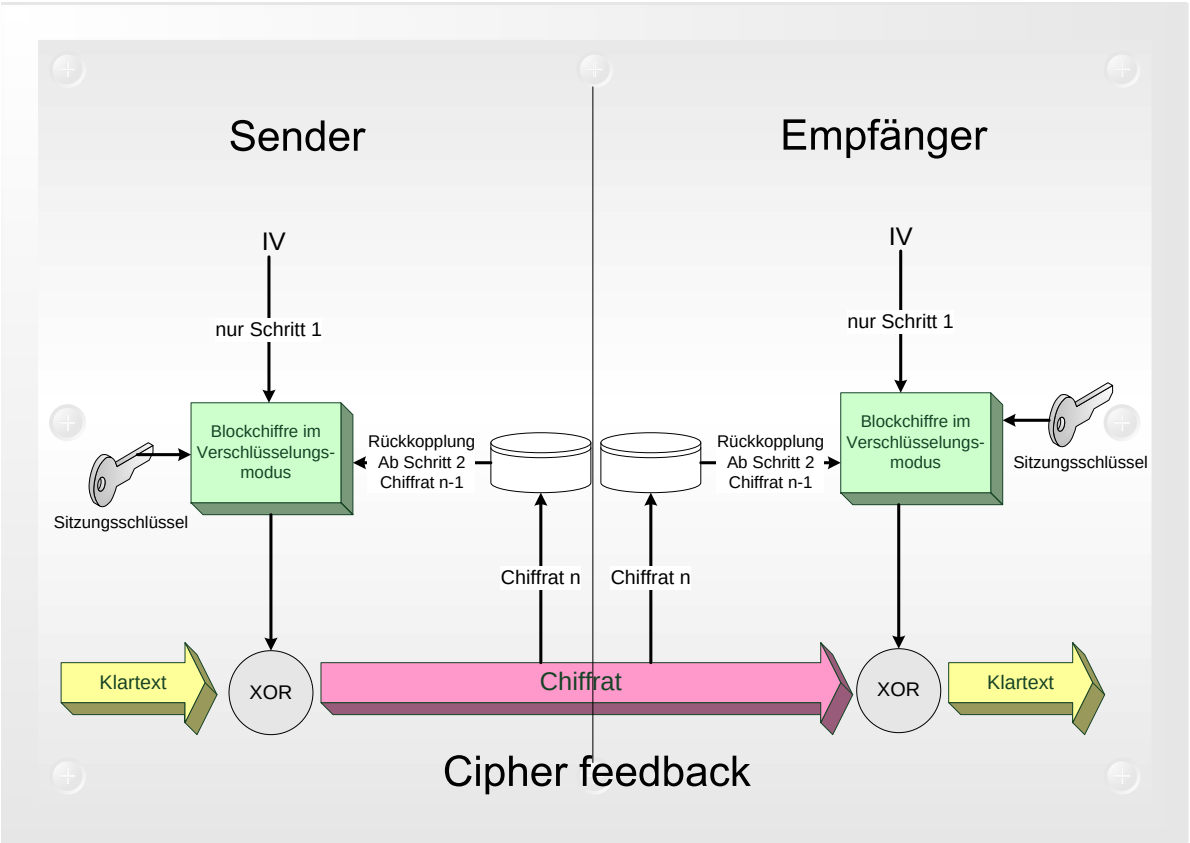


Abbildung A.4: Cipher Feedback Mode (CFB)

Definition A.12 (OFB-Operationsmodus)**.** *Beim „Output Feedback Mode“ (OFB) geschieht der erste Schritt analog zum CFB. Beim zweiten Schritt wird jedoch nicht das Chiffrat des ersten Durchgangs verwendet, sondern nur das Teilchiffrat, bevor es mit dem Klartext verknüpft wurde.* [60, nach 7.20]

Wie beim CFB wird beim OFB der Klartext nicht mit der Blockchiffre verschlüsselt, sondern per Entweder-Oder-Verschlüsselung mit einem Schlüssel (Teil-Chiffrat) gleicher Länge verknüpft. Lediglich zur Berechnung des Schlüssels wird die Blockchiffre herangezogen. Auch hier wird die Blockchiffre nur als Einwegfunktion genutzt, weswegen ausschließlich die Implementierung der Verschlüsselung der Blockchiffre benötigt wird. Im Gegensatz zum CFB wird jedoch das TEIL-CHIFFRAT[3] der vorherigen Berechnung, nicht das vorherige Chiffrat, zur Rückkopplung verwendet. Dadurch ist die Schlüsselberechnung unabhängig vom Klartext und kann vorausberechnet werden. Da ein unabhängiger Schlüsselstrom berechnet wird, der durch bitweises Entweder-Oder mit dem Klartext verknüpft wird, existiert keine Fehlerfortpflanzung.

Definition A.13 (CTR-Operationsmodus)**.** *Beim „Counter Mode“ (CTR) handelt es sich um eine Variante des OFB, bei dem keine Rückkopplung verwendet wird, sondern ein iterierender Zähler. Der IV ist also 1, im zweiten Schritt wird die 2 als Rückkopplungsersatz verwendet, et cetera.* [60, nach 7.23]

[3] Das Teil-Chiffrat und der Sitzungsschlüssel ergeben dabei den aktuellen Schlüssel des Schlüsselstroms (laufender Schlüssel/running key).

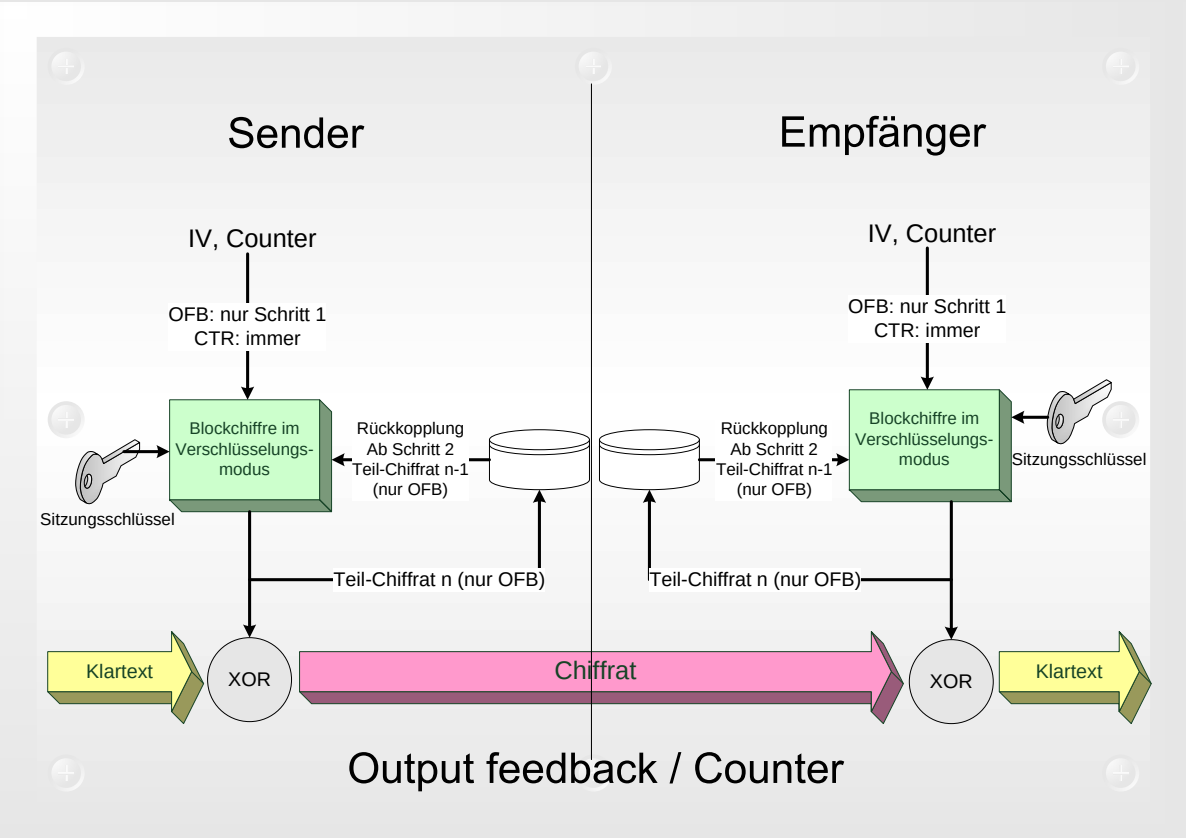


Abbildung A.5: Output Feedback Mode (OFB) / Counter Mode (CTR)

In der Praxis wird der Zähler im Zählermodus (CTR) allerdings nicht als reine $0, 1, 2$ interpretiert, sondern er besteht aus zwei Teilen. Die erste Hälfte ist dabei eine Zufallszahl, quasi ein IV, und der hintere Teil ein Zähler. Auf diese Weise wird bei jeder Sitzung ein neuer Zählerstartwert generiert. Sowohl beim OFB als auch beim CTR muss darauf geachtet werden, dass es Zyklen gibt. Je nach Länge des Rückkopplungsregisters beziehungsweise des Zählers steht nach einer Reihe von Iterationen der gleiche Wert wieder an. Die Sitzung muss vor dem Ende des ersten Zyklus beendet werden. Genauso wie beim OFB gibt es beim CTR keine Fehlerfortpflanzung.

B Begrifflichkeiten

B.1 Fachbegriffe des Transportwesens

Catwalk Begehbarer Freiraum zwischen zwei Containerstapeln auf einem Schiff.

Consolidation-Container Teilen sich mehrere Spediteure einen Container durch die Versendung von Stückgut (*LCL*), spricht man von einem *Consolidation-* oder *Groupage-Container*.

Containerdepot Ein Containerdepot ist ein Stellplatzareal für Leercontainer. Ein *Reeder* oder Container-Vermieter hat in der Regel Container auf verschiedenen Depots platziert und weist diese *Speditionen* zu, die den Container anmieten.

Containerslot Containerstellfläche auf einem Schiff, entspricht einem TEU.

Containerterminal Ein Containerterminal ist für den Waren-Umschlag verantwortlich. Hier werden die Container von einem Transportmittel auf ein anderes umgeschlagen.

Containeryard Bei einem Containeryard handelt es sich um eine Stellfläche, auf der ein Container für längere Zeit abgestellt wird.

FCL – Full Container Load FCL beschreibt eine gesamte Containerladung. Wenn ein Spediteur einen Container eines Reeders anmietet, kann dieser entweder einen Container komplett beziehen (FCL) oder nur einen Teil des Containers beanspruchen (LCL).

Frachtführer Der Frachtführer (engl. *Carrier*) führt den tatsächlichen Transport mit einem Transportmittel durch.

Groupage-Container Siehe *Consolidation-Container*.

Interchange Übergabe von Containern zwischen zwei *Transporteuren*. Hierbei ist ein bestimmtes Protokoll zu wahren. Äußere Beschädigungen am Container oder Siegel sind dabei zu untersuchen. Ist der Container leer, muss er unter anderem besenrein übergeben werden.

ISO-Container Die ISO hat die Außenmaße von Containern standardisiert. Hauptverbreitung finden dabei der 20-Fuß-Container und der 40-Fuß-Container, wobei der 20-Fuß-Container etwas kleiner als 20 Fuß ist, damit ein 40-Fuß-Container auf zwei Stellflächen passt (inklusive *Catwalk* zwischen den Containern).

LCL – Less then Container Load LCL kann auch als Stückgut bezeichnet werden. Ein Spediteur bestellt bei LCL nur den Platz in einem Container, den er wirklich benötigt. Bei LCL teilen sich Spediteure einen Container: Diese heißen *Groupage-* oder *Consolidation-Container*.

Logistiker Mit Logistiker werden in dieser Arbeit all jene Parteien bezeichnet, die mit dem Transport oder der Transportorganisation zu tun haben. Dabei handelt es sich um die *Transporteure* ergänzt um die *Spediteure*.

LSP – Logistic Service Provider siehe Logistiker

Owner (Ship-Owner) Die Owner- oder Master-Reederei besitzt ein Schiff und verkauft Container-Slots an andere Reedereien, die als *Slot-Charter* fungieren.

Reederei/Reeder Eine *Reederei* (auch *Reeder* genannt) ist ein Transportunternehmen im Schifffahrtsbereich. Ein Reeder besitzt Containerstellflächen auf Schiffen. Er kann dabei auch der Schiffseigner sein. Bei großen Containerschiffen besitzen meist mehrere Reeder die Containerstellplätze. Ferner gehören die Container meist *Reedereien*.

Shipper Bei dem *Shipper* handelt es sich um das Unternehmen, das den Transport in Auftrag gibt. Es ist das Unternehmen, welches den Warentransport von *A* nach *B* beauftragt.

Slot-Charter *Reederei*, die *Containerslots* von einem *Owner* chartert.

Spedition Organisator/Vermittler des Warentransports. Eine *Spedition* bringt Frachtgut, Container und Frachtführer zusammen. Sie hat im Regelfall keine eigenen Frachtmittel. Ein Spediteur kann auch eine Transportkette über verschiedene Frachtführer organisieren.

TEU TEU steht für „Twenty-feet Equivalent Unit“ und entspricht dem standardisierten 20-Fuß-*ISO-Container*. Containerschiffsgrößen werden in TEU angegeben. Ein 40-Fuß-Container passt auf den Stellplatz von zwei TEU hintereinander.

Transporteur Diese Arbeit definiert Transporteur wie folgt: Alle Parteien, die den Container physisch transportieren oder ihn lagern, werden Transporteur genannt. Gelagert wird der Container auf dem Containerdepot und dem Containeryard. Umgeschlagen (und somit gelagert und transportiert) wird er auf dem Containerterminal. Transportiert wird er von Frachtführern und Reedern. Nicht zu den Transporteuren gehören *Speditionen* (ansonsten *Logistiker*), Shipper und Versicherer.

Transportmittel Transportmittel sind zum Beispiel Seeschiffe, Binnenschiffe, Eisenbahn, LKW und Flugzeuge.

B.2 Abkürzungen des Transportwesens

AMS „Automated Manifest System“ - Elektronisches Meldesystem, bei dem sämtliche ladungsbezogenen Informationen eines Schiffes 24 Stunden vor Ladefrist an den US-Zoll elektronisch gemeldet werden müssen.

BoL Bei der Bill of Lading handelt es sich um die internationale Bezeichnung der Ladungspapiere.

BRZ Bruttoraumzahl - „Maßeinheit nach IMO-Vermessung für die Verdrängung eines Schiffs; sie ist eine dimensionslose Zahl, basiert aber auf dem Gesamtvolumen eines in Kubikmetern vermessenen Schiffs, multipliziert mit einem von der Schiffsgröße abhängigen Faktor. Mit dem 1994 in Kraft getretenen Schiffsvermessungsabkommen ersetzt die BRZ die Bruttoregistertonne.“ (Quelle: `http://www.tis-gdv.de/tis/taz/b/bruttoraumzahl.htm`)

DA „Designated Authority“ – Hierbei handelt es sich um eine verantwortliche Stelle auf Seiten des Staates, die den ISPS-Code im Hafen durchsetzen darf.

IMO „International Maritime Organisation“ – Unterorganisation der Vereinten Nationen, die für die Regelung der Handelsschifffahrt zuständig ist.

ISPS-Code „International Ship and Port Facility Security Code“ – Dokument, das die Sicherheit von Schiffen und Häfen aus Angst vor Terrorismus regelt. (siehe 1.2.1)

RSO „Recognised Security Organisation“ – Eine Stelle mit Fachwissen in Sicherheitsangelegenheiten und einschlägigen Kenntnissen über betriebliche Vorgänge auf Schiffen und in Häfen, die ermächtigt ist, eine nach Teil A des ISPS-Codes vorgeschriebene Bewertung, Überprüfung, Genehmigung oder Zeugniserteilung durchzuführen. (Quelle: `http://www.tis-gdv.de/tis/tagungen/workshop/2004/eggers1.pdf`

SOLAS „Safety Of Life At Sea Convention“ – Von der IMO entwickelter internationaler Vertrag, der das menschliche Lebens auf See schützen soll; der ISPS-Code ist eine Erweiterung von SOLAS.

PFSO „Port Facility Security Officer“ – Verantwortlicher für die Gefahrenabwehr in Hafenanlagen nach dem ISPS-Code.

PFSA „Port Facility Security Assesments“ – Im Verantwortungsbereich der DA zu erstellende Gutachten zur Risikobewertung der Hafenanlagen nach dem ISPS-Code.

PFSP „Port Facility Security Plan“ – Analog zum SSP ist der PFSP ein Plan, wie im Ernstfall die Gefahr für die Hafenanlage abgewehrt wird, die im PFSA erörtert wurde.

SSA „Ship Security Assesments“ – Risikobewertung, die für jedes Schiff von der verantwortlichen Reederei durchzuführen ist. [14]

SSO „Ship Security Officer“ – Verantwortlicher für die Umsetzung des ISPS-Codes an Bord, der vom zuständigen Reeder bestimmt wird.

SSP „Ship Security Plan“ – Plan zur Gefahrenabwehr. Basierend auf den Gefahren, die der SSA bewertet hat, ist der SSP der zugehörige Plan, wie im Ernstfall mit der Gefahr umzugehen ist.

B.3 Abkürzungen der Computersicherheit

ACL Eine ACL ist eine Access Control List, also eine Zugriffskontrollliste, die am Ort des Zugriffs die Zuordnung von Ressourcen an Benutzerrecht verwaltet. Bei einer ACL handelt es sich um eine „White List“ von Richtlinien.

AES AES steht für „Advanced Encryption Standard“ und ist die amerikanische Standardisierung der Blockchiffre (siehe Anhang A.3.1) Rijndael.

AMANDA AMANDA ist ein delegierungsfähiges Zugriffskontrollsystem, welches in Abschnitt 5.3.2 erklärt wird.

Hash Hashwerte und Hashfunktionen werden in Anhang A.1 erklärt.

MAC Message Authentication Code wird in Anhang A.2 erklärt.

MD5 MD5 ist eine Hashfunktion, die Schwachstellen bei der Kollisionsresistenz aufweist.

RSA RSA ist ein Algorithmus zur asymmetrischen Verschlüsselung (siehe A.3.1) und wurde von Ron Rivest, Adi Shamir und Leonard Adleman erfunden.

SHA SHA-1 ist eine Hashfunktion, die noch als sicher gilt. Allerdings glauben Experten, dass es in absehbarer Zeit durchführbar sein wird, Kollisionen zu generieren. Sie empfehlen die Verwendung von SHA-256 und SHA-512 für Anwendungen mit einer längeren Laufzeit.

SPKI Abkürzung für SPKI/SDSI (Simple Public Key Infrastructure/Simple Distributed Security Infrastructure) – Erklärung siehe 5.3.1

XACML XACML steht für „eXtensible Access Control Markup Language“ und ist ein XML-Schema, welches die Darstellung und Verarbeitung von Autorisierungsrechten standardisiert. Der vom OASIS-Konsortium definierte Standard dient dazu, Regeln aufzustellen, durch deren Auswertung der Zugriff von Subjekten auf Ressourcen eines Systems gesteuert werden kann.

B.4 Authentifizierung

Als Basis für die deutsche Sprachrichtigkeit definiert der Duden die Begriffe *authentisieren* und *authentifizieren* wie folgt.

authentisieren – laut Duden: gehoben für glaubwürdig/rechtskräftig machen

authentifizieren – laut Duden: die Echtheit von etwas bezeugen/beglaubigen

Das englische Verb *to authenticate* unterscheidet nicht zwischen beiden Wortbedeutungen. Es bedeutet sowohl „glaubwürdig machen“ als auch „beglaubigen“.

In der Computerwelt hat der englische Begriff „to authenticate“ eine besondere Wortbedeutung erlangt; diese ergänzt im Englischen die Bedeutung der beiden deutschen Übersetzungen.

to authenticate – Die NSA definierte den Begriff auf ihrer INFOSEC [63] folgendermaßen: „Verify the identity of a user, user device, or other entity, or the integrity of data stored, transmitted, or otherwise exposed to unauthorized modification in an automated information system, or **establish the validity of a transmitted message**“. Der hervorgehobene Teil dieser Definition ist der für diese Arbeit relevante. Übersetzt bedeutet dies: „Gültigkeit einer übertragenen Nachricht nachweisen.“ Die Gültigkeit bezieht sich auf den Anspruch, dass sich der Senderprozess für etwas ausgibt, wofür er sich ausgeben darf. Wenn der Benutzer einer Bankcard an einem Bankautomaten seinen PIN eingibt und dieser zu der Karte gehört, ist die Gültigkeit festgestellt. Der Benutzer kann sowohl der rechtmäßige Besitzer dieser Karte als auch ein befreundeter PIN-Mitwisser als auch ein bösartiger Trickbetrüger sein. Auch wenn sich Personen mit digitalen Signaturen (siehe A.3.4) identifizieren wollen, wird in Wirklichkeit nur ihr privater Schlüssel identifiziert.

Entity authentication wird vom HAC folgendermaßen definiert: „EA is the process whereby one party is assured (through acquisition of corroborative evidence) of the identity of a second party involved in a protocol, and that the second has actually participated (i.e., is active at, or immediately prior to, the time the evidence is acquired)“ [60]. Diese Quelle bezieht sich auf eine Entität und bleibt somit allgemeingültig. Im Gegensatz zur Definition von der NSA wird aus gutem Grund das Wort „user“ (Benutzer) umgangen.

Weder die Erklärung des Duden für authentifizieren („die Echtheit von etwas bezeugen/beglaubigen.“) noch die für authentisieren („glaubwürdig machen“) reicht aus, um das Verb „to authenticate“, wie es in den englischen Definitionen erklärt ist, eindeutig zu übersetzen. Abhilfe schafft das Bundesamt für Sicherheit und Informationstechnik (BSI). Im E-Government-Handbuch [15] werden die zugehörigen Substantive folgend definiert:

Authentisierung – Unter einer Authentisierung (engl. authentication) wird die Vorlage eines Nachweises eines Kommunikationspartners verstanden, in dem bestätigt wird, dass er tatsächlich derjenige ist, der er vorgibt zu sein.

Authentifizierung – Unter einer Authentifizierung (engl. authentication) wird die Prüfung einer Authentisierung verstanden, das heißt die Überprüfung, dass ein Kommunikationspartner tatsächlich derjenige ist, der er vorgibt zu sein.

Die Authentisierung beschreibt somit das senderseitige Berechnen und Anhängen der Authentisierungsinformation. Authentisieren ist auf den Sender beschränkt und beschreibt den Vorgang, der für das Ausführen einer Authentisierung notwendig ist. Der Empfänger wiederum, der die Authentisierungsinformation verarbeitet, kann diese für gültig oder nicht gültig erklären. Im ersten Fall spricht man von einer Authentifizierung. Diese kann nur vom Empfänger durchgeführt werden. Das erfolgreiche Prüfen wird authentifizieren genannt.

C Beispiele für den MASC-ID

Dieses Kapitel führt begleitende Beispiele für das bessere Verständnis des Kapitels 5, im Besonderen des Abschnitts 5.3 auf.

C.1 SPKI-Beispiele

Um die SPKI-Konzepte besser zu verstehen, sind folgend begleitende Beispiele aufgeführt.

C.1.1 Delegierung mit SPKI

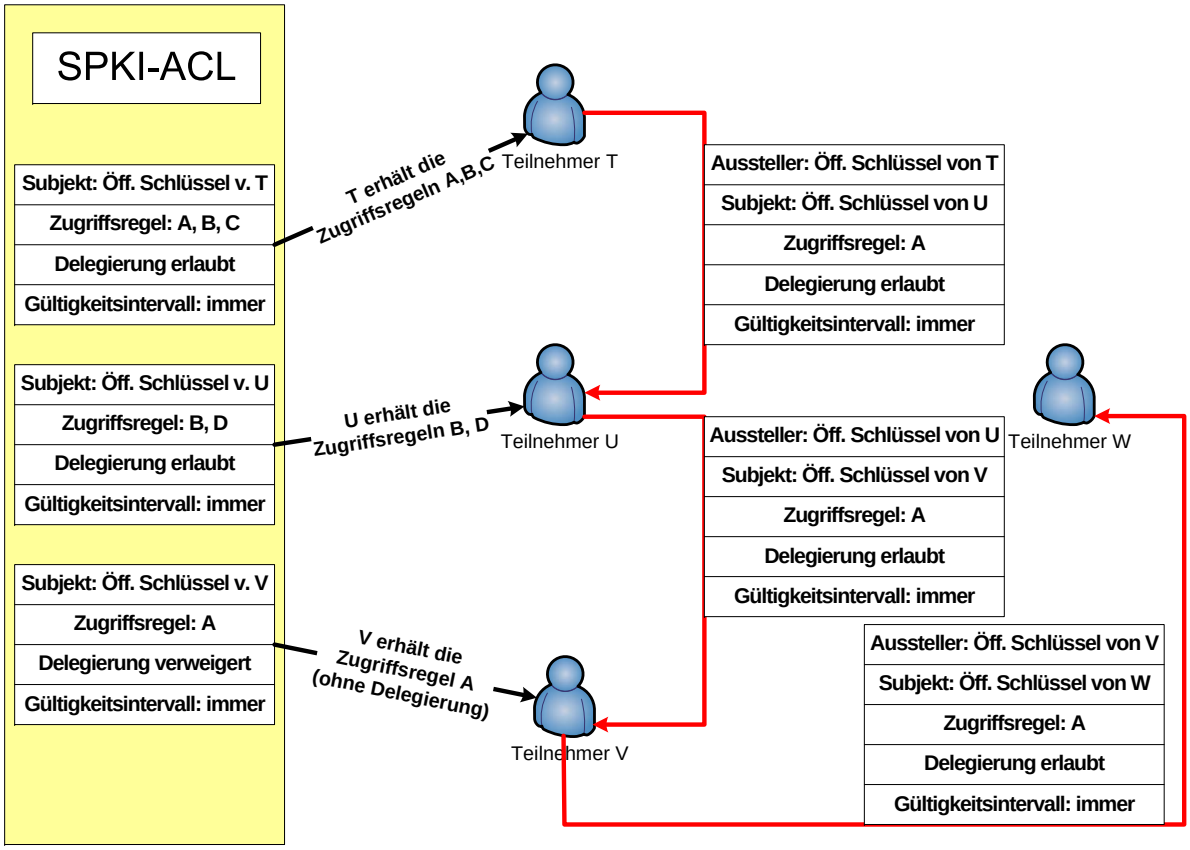


Abbildung C.1: Delegierungsbeispiel mit SPKI

Beispiel C.1 (Delegierung (nach Abbildung C.1))**:**

Es existiert eine SPKI-ACL, die dem Teilnehmer T die Regeln A, B und C zuweist. Der Teilnehmer U bekommt die Regeln B und D zugewiesen und V erhält auf diese Weise A. Alle außer V dürfen die

Regeln an andere Teilnehmer delegieren. Mit einem Zertifikat delegiert T die Regel A an U. Somit besitzt U folgend die Zugriffsrechte A, B und D. Alle darf er delegieren. Er delegiert folgend A an den Teilnehmer V, der bereits das Zugriffsrecht A hat. Allerdings darf er A nicht delegieren, weil das Delegierungsbit nicht gesetzt ist. Dies ändert sich jedoch durch den Umstand, dass er zusätzlich die delegierungsfähige Erlaubnis A von den Teilnehmern T und U delegiert bekommt. Jetzt darf Teilnehmer V das Zugriffsrecht A auch an den Teilnehmer W weitergeben. Abbildung C.1 stellt schematisch diesen Sachverhalt dar und zeigt die notwendigen Zertifikate.

Zur Überprüfung der Zugriffsrechte muss ein Teilnehmer beim Zugriffskontrollsystem (Kombination aus PEP und PDP) seinen Zugriffswunsch äußern und seine gesamte ihn autorisierende **Zertifikatskette** vorlegen. Gemäß dem Beispiel erhält W in einer Kette von drei Delegierungen die Zugriffsrecht A zugewiesen.

Falls der Teilnehmer W auf A zugreifen möchte und W dem Wächter des Dienstes (PDP/PEP) lediglich das Zertifikat von V an W vorzeigt, kann dieser ihm keinen Zugriff gewähren. Bei einem Blick in die SPKI-ACL ist V nämlich keine Delegierung von A erlaubt. Der Teilnehmer W muss also beim Zugriff nachweisen, dass V dieses Recht der Delegierung besitzt. Dies ist nur möglich, wenn er alle drei dargestellten Zertifikate vorlegt.

C.2 AMANDA-Beispiele

Zum besseren Verständnis von AMANDA seien einige Beispiele beigefügt.

C.2.1 Privilegien

Beispiel C.2 (Privilegien)**:**

Teilnehmer a besitzt die Ressource R. Der lesende Zugriff auf R sei durch die Autorisierung α ausgedrückt. Teilnehmer a erteilt Teilnehmer b die Berechtigung, α einem anderen Teilnehmer zuzuteilen. Teilnehmer b erhält somit das Managementprivileg für α. Darauf erteilt b Teilnehmer c ein Zugriffsprivileg und d ein weiteres Managementprivileg jeweils für α. Teilnehmer d erteilt schließlich e ein Zugriffsprivileg. Es dürfen am Ende zwei Parteien auf R lesend zugreifen und somit die Autorisierung ausüben. Beide Zugriffsberechtigungen folgen einer Delegierungskette (siehe nachfolgende Definition 5.7). Einmal wird die Autorisierung von a über b zu c delegiert und die zweite Delegierungskette lautet $(abde)$. Teilnehmer a als Besitzer der Ressource darf natürlich auf R zugreifen, b und d dürfen bezüglich der genannten Privilegienvergabe α ausüben.

C.2.2 Constraints

Constraints bestehen aus Gruppen, Delegierungsketten aus Teilnehmern. Dennoch enthalten folgend auch Constraints kleine Buchstaben. Kleine Buchstaben in Constraints stehen für einelementige Gruppen, die in der Praxis einzelnen Teilnehmern entsprechen. Constraints sind nur für Gruppen definiert, damit die $\in$-Relation zwischen Teilnehmern und Gruppen definiert ist.

Beispiel C.3 (Einfacher Constraint)**:**

Der Constraint *AbC* schreibt vor, dass nur Gruppenmitglieder von *C* die betreffende Autorisierung ausüben dürfen, wenn Teilnehmer *b* die Richtlinie dafür erstellt hat. Allerdings muss ein Gruppenmitglied von *A* vorher *b* dazu berechtigt haben. Der Constraint schreibt also vor, wie die Delegierungskette durchlaufen werden muss. Die Delegierungsketten (a_1bc_1) oder (a_2bc_5) sind Beispiele, die sich von dem Constraint *AbC* repräsentieren lassen.

Beispiel C.4 (Kleene-Stern)**:**

Der Constraint mit dem Kleene-Stern ABC^* umfasst Delegierungsketten, die sowohl mit einem Gruppenmitglied von *B* als auch mit Teilnehmern von *C* enden. Die drei Delegierungsketten (a_1b_2), $(a_1b_1c_2)$, $(a_1b_1c_1c_2)$ sind Beispiele für durch den erweiterten Constraint erlaubte Inkarnationen.

C.2.3 Beispielszenario

Kleine Buchstaben in Constraints entsprechen zur Vereinfachung einelementigen Gruppen (siehe C.2.2).

Angenommen, *b* ist ein Zwischenhändler für Pressefotographien: Seine Mitarbeiter gehören der Gruppe *C* an. Ihre Aufgabe ist es, die Fotos zu verkaufen. Jedoch ist *b* kein Fotograph, weswegen er die Bilder einkaufen muss. Fotograph *a* hat zwei brandheiße Fotos gemacht, bei denen *b* glaubt, dass er sie verkaufen kann. Er erwirbt sie beim Fotographen für den Weiterverkauf an die Presse. Der Fotograph will sichergehen, dass die Fotos nur für diesen Zweck eingesetzt werden. Er gibt die Fotos nicht aus der Hand, sondern delegiert die Zugriffsrechte. α, β sind die Autorisierungen, die die Leserechte den Bildern zuordnen. Er möchte aber nicht, dass der Fotomakler und seine Abteilung das Bild kopieren können, deswegen erlaubt er nur die Delegierung des Leserechtes. Ausschließlich Pressevertreter (Mitglieder von *D*) dürfen im Endeffekt auf das Bild zugreifen. Außerdem darf zwischen den Mitarbeitern *C* und der Presse kein weiterer Makler zwischengeschaltet sein.

Das Subjekt seines Zertifikats an *b* ist daher bC^*D. Dies erlaubt nur *b* die Weitergabe der Zugriffsrechte an die Presse direkt oder mit beliebig vielen Zwischenschritten in seiner Abteilung. Das abgetretene Recht gilt für immer.

Mitarbeiter c_1 ist für den Kontakt mit dem Zeitungsverleger d_1 zuständig. Makler *b* weist ihm mit Zertifikat 2 das Delegierungsrecht für das mutmaßlich für die Zeitung besser geeignete Bild α zu. Der

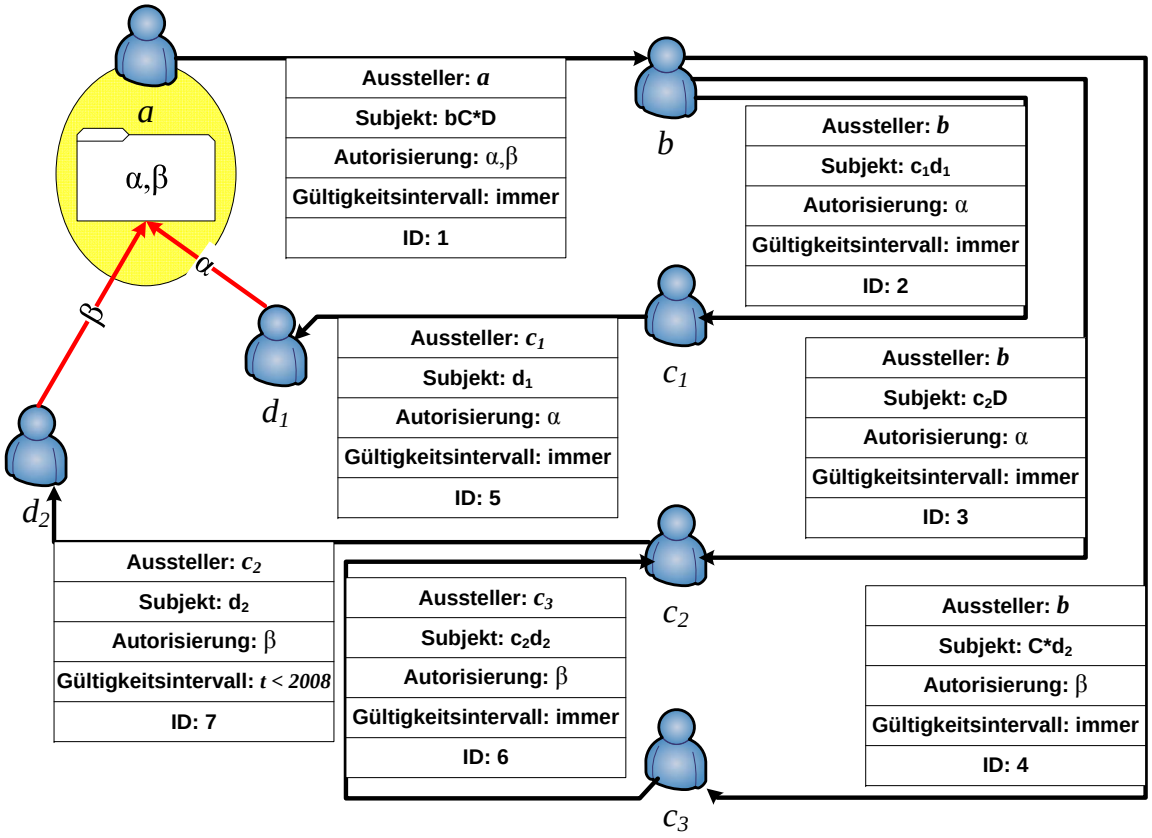


Abbildung C.2: Delegierungsbeispiel AMANDA

Mitarbeiter c_2 erhält mit Zertifikat 3 das Recht, α an einen beliebigen Pressevertreter zu delegieren. Da c_2 gute Kontakte zu Nachrichtenredaktionen hat, soll er probieren, das Bild an diese Parteien zu verkaufen. Das Bild hinter der Autorisierung β sei eher für die Regenbogenpresse geeignet. Deshalb übergibt der Makler seinem Spezialisten c_3 dieses Zugriffsrecht zur Delegierung. Allerdings hat c_3 nicht viele direkte Kontakte, weswegen er das Bild über andere Mitarbeiter in seiner Abteilung delegieren darf. Letztendlich ist es doch eine Nachrichtenredaktion, die das Bild haben möchte. Deshalb delegiert c_3 das Zugriffsrecht an c_2 weiter, der dem Verleger d_2 das Zugriffsrecht β erteilt. Gleichzeitig ist es auch c_1 geglückt, α an d_1 zu verkaufen. Mit Zertifikat 5 erteilt c_1 d_1 den Zugriff.

Es sind in den zwei Delegierungsketten einige weitere Einschränkungen geschehen. Makler b hat bei den Zertifikaten 2 und 3 die Delegierungskette auf spezielle Mitarbeiter beschränkt, während er im Zertifikat 4 seinem Mitarbeiter c_3 erlaubt hat, weitere Delegierungen im eigenen Betrieb vorzunehmen. Ferner hat er die Zugriffsrechte auf die Zertifikate aufgeteilt und in den ersten beiden Zertifikaten nur die Delegierung von α erlaubt, während er im Zertifikat 4 nur β zur Delegierung freigibt. Das Zertifikat 7 erhält dazu eine feinere Einschränkung im Gültigkeitsintervall. Es ist dem Verleger d_2 also nur erlaubt, auf das Bild in der Zeit vor dem Jahr 2008 zuzugreifen.

C.2.4 Widerruf

Beispiel C.5 (Widerruf)**:**

Eine Firma hat einige Teilaufgaben ausgelagert und deswegen ein Delegierungsschema gewählt, mit dem Aufgaben und Autorisierungen an diese Fremdfirma abgegeben wurde. Als Gültigkeitszeitraum wurde dabei ein ganzes Jahr gewählt ($t \leq 365$). Allerdings erfolgt nach 200 Tagen eine Firmenfusion und der Fusionspartner stellt eine Abteilung, die die delegierte Teilaufgabe in Zukunft abwickeln soll. Nach 100 Tagen ist die Zwischenphase beendet und die neue Partnerabteilung übernimmt die delegierten Aufgaben. Sie bekommt für die gesamte Zwischenphase und den Rest des Jahres die Delegierungsrechte erteilt ($200 < t \leq 365$). Gleichzeitig soll der nicht mehr benötigte Unterauftragnehmer die Erlaubnis für das zukünftige Ausführen der Teilaufgaben aberkannt bekommen, ohne dass die bis dahin erledigte Arbeit ungültig wird. Daher wird das PZert, welches die Delegierung der Teilaufgaben durchführte, für das Zeitintervall $t > 300$ widerrufen. In der Datenbank der Zertifikate wird neben dem PZert auch das WZert eingefügt. Für diesen Fall ist der resultierende Gültigkeitszeitraum für die Delegierungen des Unterauftragnehmers $t \leq 300$.

C.3 Beispiele für den MASC-ID

C.3.1 Generelle Funktionsweise des MASC-ID

Beispiel C.6 (Consolidation-Container)**:**

Gesetzt den Fall, der Spediteur hat einen Consolidation-Container zusammengestellt und von zwei Lieferanten und einer weiteren Spedition Güter im Container, dann bietet er diesen die Sensorinformationen an. Angenommen, einer der Shipper transportiert ausschließlich verzinkte Metallwaren, denen weder zu hohe Temperaturen noch Luftfeuchtigkeitsschwankungen etwas anhaben können: Wegen potenzieller Beschädigungen ist der Shipper lediglich daran interessiert, ob der Container fällt oder gestaucht wird. Er verzichtet darauf, weitere Sensordienste bei der Spedition zu buchen. Der zweite Shipper liefert Baumwollprodukte, die nur durch Nässe zerstört werden können. Daher abonniert dieser den Feuchtegehalt des Containers. Der Subspediteur möchte alle Informationen haben, da er seinen Teil des Containers wiederum an andere Shipper vermietet hat. Die Transportversicherung für den gesamten Transport möchte ebenfalls alle Sensorinformationen bekommen. Durch initiale Richtlinien ist der Spediteur berechtigt, Delegierungen für die Sensordaten zu erstellen. Daher generiert er vier verschiedene Delegierungszertifikate und ordnet auf diese Weise jeder betroffenen Partei die benötigten Autorisierungen zu.

C.3.2 Lokale Gruppen in AMANDA

Neben den globalen Gruppen werden zu jedem Transport lokale Gruppen definiert. Diese werden in einer weiteren Datenbanktabelle in der Transportdatenbank erstellt (siehe Abbildung 5.4). Diese Gruppen

beziehen sich nicht wie die globalen Gruppen auf Rollen, sondern auf „Teiltransport-Beteiligungen“. Die gesamte Transportkette wird gemäß den Angaben der Logistiker in Teiltransporte aufgeteilt. Die lokalen Gruppen werden über die eindeutige Identifizierung der Transportmittel (Transportmittel-ID) referenziert. Auf diese Weise kann beim Datenzugriff die Sicht der Sensordaten auf diese Teiltransporte beschränkt werden, indem die Daten ausgewertet werden, die der Gateway den Containerdaten hinzufügt. Es ergibt sich die Datenbankstruktur bezüglich der AMANDA-Zugriffsrechtverwaltung gemäß Abbildung 5.4.

A: Container Depot bei Peking
Export-Erich (Shipper)
Supersped (Spedition A)
China Carriages (Spedition B)
Peking Parking (Depotbetreiber)
Chinesischer Zoll

B: LKW-Transport in Peking
Export-Erich (Shipper)
Supersped (Spedition A)
China Carriages (Spedition B)
Lu Lings Lasterladen (Frachtführer)
Chinesischer Zoll

C: Schienentransport China
Export-Erich (Shipper)
Supersped (Spedition A)
China Carriages (Spedition B)
China Schienenverkehr (Frachtführer)
Chinesischer Zoll

D: Verladung in Shanghai
Export-Erich (Shipper)
Supersped (Spedition A)
China Carriages (Spedition B)
Shanghai Shipping (Terminalbetreiber)
Chinesischer Zoll

E: Schiffstransport
Export-Erich (Shipper)
Supersped (Spedition A)
Container-Claus (Frachtführer)
Chinesischer Zoll
Deutscher Zoll (weitere Zollbehörden?)

F: Verladung in Hamburg
Export-Erich (Shipper)
Supersped (Spedition A)
Hamburger Hafenhalde (Terminalbetreiber)
Deutscher Zoll

G: LKW-Transport Deutschland
Export-Erich (Shipper)
Supersped (Spedition A)
Deutscher Zoll

H: Containerdepot Berlin
Export-Erich (Shipper)
Supersped (Spedition A)
Charlottenburg Container (Depotbetreiber)
Deutscher Zoll

Abbildung C.3: Lokale Gruppen (Beispiel)

Beispiel C.7 (Lokale Gruppen)**:**

Shipper EXPORT-ERICH *beauftragt Spediteur* SUPERSPED *mit dem Transport seiner Ware von Peking nach Berlin.* SUPERSPED *beauftragt den Unterspediteur* CHINA CARRIAGES *mit dem Transport von Peking nach Shanghai.* CHINA CARRIAGES *beauftragt den LKW-Frachtführer* LU LINGS LASTERLADEN *mit dem Teiltransport vom Containerdepot* PEKING PARKING *zum Shipper und von dort zum Verladebahnhof des Eisenbahnverladers* CHINA SCHIENENVERKEHR *in Peking.* CHINA SCHIENENVERKEHR *liefert den Container selbständig zu Containerterminal* SHANGHAI SHIPPING*, wo der Container umgeladen wird. Das Containerschiff, auf das der Container geladen wird, gehört dem Reeder* CONTAINER-CLAUS*, der den Container bis in den Hamburger Hafen bringt.* CONTAINER-CLAUS *wurde durch* SUPERSPED *selber beauftragt. Da* SUPERSPED *selber auch LKWs*

besitzt, bringt er die Container zum Ziel, dem Depot CHARLOTTENBURG CONTAINER*, nachdem er ihn am Containerterminal* HAMBURGER HAFENHALDE *in Empfang genommen hat.*

Die acht Teiltransporte werden lokalen Gruppen zugeordnet und alle Beteiligten entsprechend zugewiesen. Abbildung C.3 stellt dies exemplarisch dar. Dabei wurden keine Containeryards und keine Versicherungen berücksichtigt und die Schachtelungstiefe des Baums auf drei begrenzt. Beliebig tiefere Bäume und dementsprechend mehr Einträge pro Gruppe sind denkbar.

Die Gruppenliste wird sukzessive vom MASC-ID erstellt. Wenn ein neuer Eintrag in der Datenbank einem echten Teiltransport entspricht, dann wird das Feld *Transportmittel-ID* (TM-ID) ausgefüllt. Der MASC-ID legt anschließend einen Eintrag in der lokalen Gruppenliste an, indem er den TM-ID als Identifizierungselement der lokalen Gruppenliste wählt. Über einen rekursiven Algorithmus werden alle Auftraggeber des Frachtführers ermittelt, indem das Tabellenelement *Auftraggeber* ausgewertet wird. Alle Auftraggeber und die Auftraggeber der Auftraggeber werden der lokalen Gruppe hinzugefügt. Auf diese Weise entstehen in der Generierungsphase der *Transportkette*-Tabelle die lokalen Gruppen automatisch. Die Lokale-Gruppen-Tabelle ist berechenbar und somit redundant, wegen des einfachen Verständnisses und der geringeren Berechnungsdauer beim Zugriff wird sie hier vorberechnet. Es wäre aber genauso möglich, diese Berechnung erst beim Zugriff durchzuführen und auf diese Tabelle zu verzichten.

Die gesamte Transportkette wird gemäß der lokalen Gruppenliste auf n Teiltransporte aufgeteilt, wobei $n =$*Anzahl lokaler Gruppen*. Über die Informationen, die die Transportmittel bei der Datenübermittlung zum MASC-ST hinzufügen, kann eine Zuordnung zwischen dem aktuellen Aufenthaltsort des Containers und der Teiltransporte geschehen. Das bedingte Constraint *Frachtführer(teiltransportbeteiligt)* bezüglich der Autorisierung *Sensordaten: Türöffnung, Temperatur, Luftfeuchte, Rauchalarm* erlaubt dem Frachtführer den lesenden Zugriff auf die vier Sensordaten nur dann, wenn er am aktuellen Transport beteiligt ist.

Auf diese Weise kann das System durch bedingte Constraints die Sicht auf die Sensordaten sehr fein einschränken. Beschränkungen in der Autorisierung müssen aber von den Delegierenden selbst durchgeführt werden. Da keine Gruppendelegierungen möglich sind, beschreibt der exemplarische Constraint nur den Rahmen der Delegierungsmöglichkeiten, der schlussendlich bezüglich eines bestimmten Frachtführers präzisiert werden muss. Mit Hilfe der bedingten Constraints kann nicht nur die Sicht auf die Sensordaten eingeschränkt werden, sondern auch der Datenbankzugriff für die Transportdaten.

D ScatterWeb-Hardware

Das ScatterWeb-Projekt der Computer Systems & Telematics Gruppe der FU Berlin bietet die Sensorknoten kommerziell an. Der Name ScatterWeb ist dabei das Betriebssystem, welches auf den Sensorknoten läuft, die an der FU Berlin entwickelt wurden. Zurzeit gibt es lediglich Entwicklerversionen, die noch nicht in großer Stückzahl gefertigt werden und die daher noch nicht die volle Leistungsfähigkeit entwickeln. Dies muss berücksichtigt werden, wenn man die gemessenen Werte interpretiert.

Als Beispiel sei die Lebensdauer genannt. Mit herkömmlichen Batterien schaffen die vorliegenden Sensoren im Ruhemodus maximal zwei bis drei Wochen. Die ScatterWeb-Webseite spricht von einer Lebensdauer von fünf Jahren, bei einem Verhältnis zwischen Übertragung und Ruhemodus von einem Prozent. Diese Daten sind allerdings nicht realisierbar, wie [80] nachweist. Bei Verwendung von Akkus muss zusätzlich auf die höhere Selbstentladung hingewiesen werden. Es kann aber davon ausgegangen werden, dass man mit besseren Produktionsbedingungen durchaus eine höhere Lebenserwartung erreichen kann.

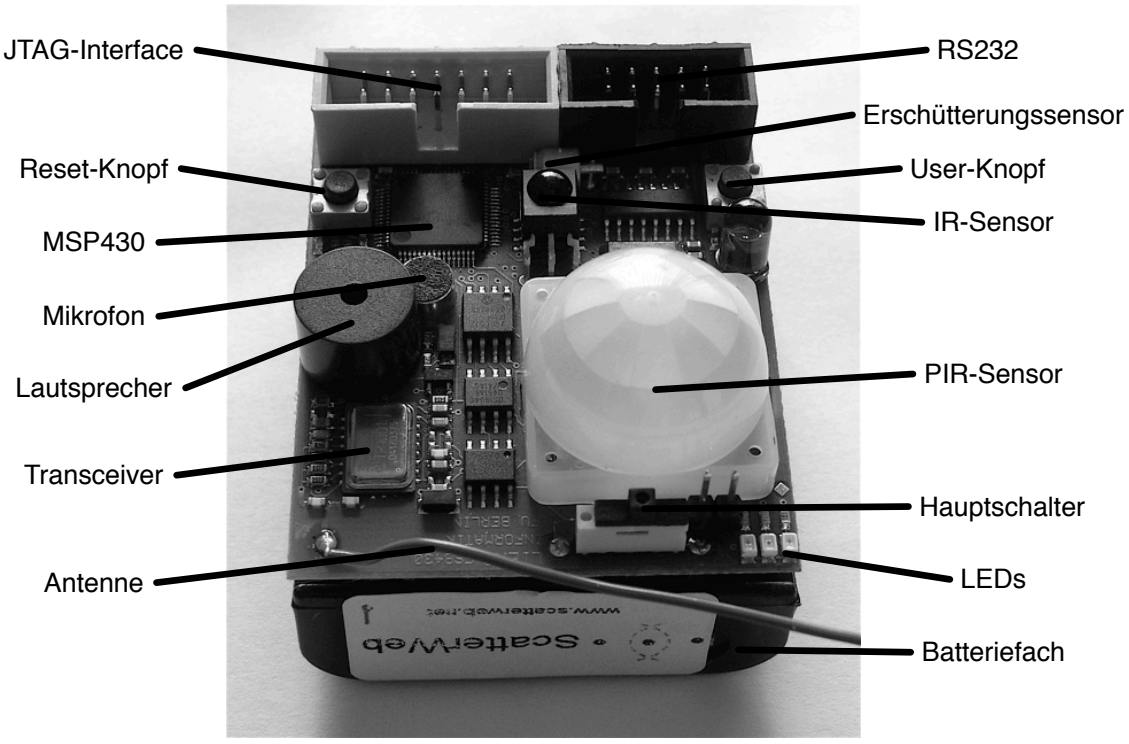


Abbildung D.1: ScatterWeb ESB (Embedded Sensor Board) [80]

[80] beschreibt die Hardware wie folgt:

- Die ESBs haben mit dem Batteriefach, der Platine und den Sensoren Gesamtabmessungen von $58mm * 47mm * 40mm$ (Länge $*$ Breite $*$ Höhe, inkl. Fresnellinse) und wiegen ca. 35 g (115 g inkl. 3 AA-Zellen).
- Die Energieversorgung erfolgt durch 3 Mignon-Zellen (AA) oder Akkus, die in einem Batteriefach auf der Unterseite des Sensorknotens untergebracht sind. Die Sensorknoten können alternativ auch durch Solarzellen u. ä. betrieben werden.
- Der Mikrocontroller ist ein MSP430 von Texas Instruments. Es handelt sich um einen 16-Bit-Mikrocontroller mit 2 kByte RAM und 60 kByte Flash-ROM, welcher zur Aufnahme der Firmware und der Anwendungen dient. Der Prozessor kann mit einer Geschwindigkeit von bis zu 8 MHz betrieben werden. Der MSP430 verfügt weiterhin über A/D-Wandler, einige Timer und einen Watchdog-Timer.
- Dem Mikrocontroller steht neben dem internen Flash-ROM noch ein 64 kByte großes EEPROM zur Verfügung, in dem beliebige Daten (zum Beispiel Sensormesswerte oder Routingtabellen) gespeichert werden können.
- Die Kommunikation innerhalb des Sensornetzes erfolgt mittels Funk im (in Europa) lizenzfreien Frequenzband bei 868,35 MHz. Es sollen Funkreichweiten auf freier Fläche von bis zu 300 m erreicht werden, in Gebäuden soll die Kommunikation durch bis zu zwei Wände möglich sein. Der Sende- und Empfangsbaustein ist ein TR1001 von RF Monolithics (RFM).
- Der ESB ist mit einer seriellen RS232-Schnittstelle zur Kommunikation mittels einer Konsole ausgerüstet. Mit der ScatterWeb-Firmware können auf dieser Konsole beispielsweise die Knoten konfiguriert, viele Funktionen der Knoten aufgerufen und auch die Ausgaben betrachtet werden. Weiterhin besitzen die Knoten eine parallele JTAG-Schnittstelle, über die neue Firmware auf den Sensorknoten überspielt werden und Debugging der Knoten durchgeführt werden kann.
- 3 LEDs in rot, gelb und grün dienen als Statusanzeigen.
- Über einen Lautsprecher können Signale ausgegeben werden, zum Beispiel zur Standortfeststellung oder als Fehlermeldung.
- Die ESBs sind mit folgenden Sensoren ausgestattet:
  - Ein Mikrofon zur Feststellung der Umgebungslautstärke.
  - Ein Bewegungssensor zur Feststellung von Erschütterungen des Knotens.
  - Ein Infrarotsensorzur Aufnahme von Signalen von Infrarotfernbedienungen. Dies kann zur Steuerung der Sensorknoten genutzt werden.
  - Ein passiver Infrarotsensor zur Bewegungserkennung, der sich unter einer weißen Fresnellinse aus Kunststoff befindet.

  - Ein Temperatursensor zur Feststellung der Umgebungstemperatur. Dieser Sensor ist im Uhrenbaustein integriert.
  - Ein Spannungssensor zur Kontrolle der eigenen Energiequelle.

- Weiterhin besitzen die ESBs die Möglichkeit, weitere Geräte über das Serial Peripheral Interface (SPI) oder den Inter-Integrated Circuit (I^2C) anzuschließen. Bei SPI handelt es sich um ein von Motorola entwickeltes Bussystem, beim I^2C um eine Entwicklung von Philips. Beide Bussysteme sind darauf ausgelegt, eine einfache serielle Verbindung herstellen zu können.

Neben den EBSs gibt es weitere Knoten, die die ScatterWeb-Gruppe vertreibt. Alle haben eine gemeinsame Firmware, die sich ScatterWeb nennt. Diese bietet die Basis für weitere Entwicklungen. Sie abstrahiert eine logische Trennung zwischen Anwendungs- und Systemteil. Nach dem Kompilieren wird diese Trennung jedoch nicht beibehalten, weswegen das Aufteilen nur als Hilfe für den Entwickler zu sehen ist. Die ScatterWeb-API bietet eine Menge komplexer Funktionen, die eine Reihe von Basisfunktionsaufrufen zusammenfasst. Als Kompiler wird .NET-Technologie von Microsoft verwendet. Die kompilierte Binärdatei kann dann auf die Sensorknoten übertragen werden. Für die Übertragung wird die JTAG-USB-Schnittstelle verwendet. Die Firmware liegt fast komplett im Quelltext offen, was sich positiv auf die Änderbarkeit und Nachvollziehbarkeit auswirkt.

E FAQ zur Frachtdokumentation

Folgend sei eine FAQ[1] zum Thema Frachtdokumentierung angeführt. Diese FAQ ist im Folgenden angeführt, um deutlich zu machen, welche Informationen von dem automatischen Manifestsystem des MASC-ID verarbeitet werden müssen. [103]

E.1 Bei Transporten in die USA müssen Manifeste in das AMS-System des amerikanischen Zolls eingegeben werden. Aus welchen Daten besteht dieses Manifest?

AMS ist ein Akronym und steht für „Automated Manifest System“. Es handelt sich hierbei um eine Sicherheitsmaßnahme auf EDV-Basis für das amerikanische Zollsystem. Seit Dezember 2002 ist weltweit allen Frachtführern im Containerverkehr mit den USA eine 24-Stunden-Vorabmeldefrist über die komplette in die USA zu verschiffende Ladung im Rahmen der „24-Stunden-Regel“ (vgl. Abschnitt 1.2.2) auferlegt. Die Reederei muss mindestens 24 Stunden vor der Beladung des Schiffes dem US-Zoll die Ladungsdaten elektronisch übermitteln. Das gilt sowohl für den letzten Hafen vor dem Anlaufen des ersten amerikanischen Hafens als auch für alle Zwischenhäfen, in denen Güter für die USA beladen wurden. Vor Beladung des Schiffes im Hafen muss die Manifestierung abgeschlossen sein und eine positive Bestätigung zum Beladen der Güter seitens der amerikanischen Zollbehörden vorliegen.

Der Grund für diese 24-Stunden-Meldefrist ist, dass ohne die rechtzeitige Übermittlung der Daten an die jeweiligen Initiativen SCTI[2] und ACE[3] keine termingerechte Überprüfung der Ladung möglich ist.

Weitere erforderliche Daten für das AMS sind:

1. Carrier SCAC Code
2. Reisenummer
3. Ankunftsdatum 1. Löschhafen in USA
4. Ladungsangaben gemäß BoL
5. Erster europäischer Ladungshafen für USA-Ladung

[1] Bei einer FAQ (Frequently Asked Questions) handelt es sich um eine Fragen-/Antwortliste.
[2] SCTI steht für Sea Cargo Targeting Initiative
[3] ACE steht für Automated Commercial Environment

6. Genaue Beschreibung der Ladung
7. Vollständige Liste aller Ablader
8. Vollständige Liste der Empfänger oder Vertreter des Eigentümers
9. Schiffsname/Nationalität des die Dokumente ausstellenden Landes/IMO-Registernummer
10. Angabe der jeweiligen Ladehäfen
11. Bei Gefahrgut Angabe der UN-Nummer
12. Containernummer
13. Siegelnummer[74]

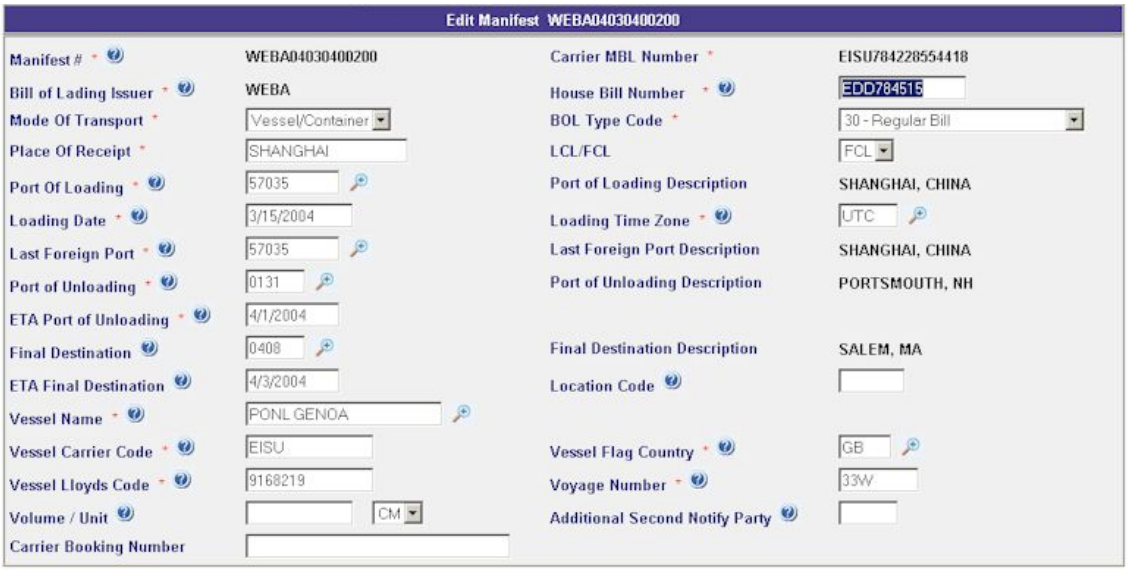


Abbildung E.1: Beispiel für das Format einer elektronischen Datenübermittlung via AMS [85]

E.2 Was ist ein Manifest und welches Format hat es? Wie und von wem wird es ausgestellt?

Ein Manifest (bezogen auf das Transportwesen) ist ein Verzeichnis für Frachtgut, das für den grenzübergreifenden Warenverkehr bestimmt ist. Manifeste werden für Administrationszwecke verwendet und von den Agenten in den jeweiligen Häfen anhand des Frachtbriefs zusammengestellt. Allgemein wird es auch als Ladeliste beziehungsweise Warenbegleitpapier bezeichnet. Im Manifest sind unter anderem sämtliche auf einem Transportmittel geladenen Gegenstände dokumentiert.

In der Logistik werden durch das Manifest alle Dokumente, die zu einer Frachtsendung gehören, zusammengefasst. Im Einzelnen sind dies:

- Der **Frachtbrief** stellt ein Beförderungsversprechen dar. In ihm werden Lade- und Entladestellen dokumentiert, die Frankierung festgelegt und der Ablieferbeleg gegengezeichnet.

- Auf der **Rechnung** können ebenfalls alle Merkmale des Frachtbriefes stehen. Die Rechnung definiert den Eigentumsübergang der Ware unter bestimmten Voraussetzungen wie der Zahlung oder dem Erlöschen des Eigentumvorbehalts.
- Der **Lieferschein** als Warenbegleitpapier dient zur Identifizierung der Ware, dem Eigentumsvorbehalt und beinhaltet Hinweise auf die AGB des Verkäufers. Dem Lieferschein muss stets die Rechnung folgen. [90]

E.3 Was ist die Bill of Lading?

Bei der Bill of Lading (BoL) handelt es sich um einen Ladeschein im See- und Binnenschifftransportwesen. Im deutschen Sprachgebrauch wurde in der Vergangenheit für BoL häufig die Bezeichnung Konnossement[4] oder Frachtbrief verwendet.

Der Frachtbrief ist nach [38] in Deutschland ein Transportpapier und regelt das Rechtsverhältnis zwischen dem Frachtführer und dem Lieferanten. Über die Verpflichtung zur Ablieferung des Gutes kann ebenso ein Ladeschein beziehungsweise ein Konnossement ausgestellt werden, wodurch sich der Frachtführer zur Auslieferung der Güter an den legitimierten Besitzer des Scheines verpflichtet. Er wird vom Frachtführer für den Folgelieferanten ausgestellt. Der Besitz des Frachtbriefes bedeutet nicht, die rechtliche Verfügungsgewalt über die Güter zu besitzen. Der Frachtbrief ist unabhängig vom Frachtvertrag und dient in erster Linie der Sicherstellung, dass die Maße oder Gewichte der Güter, die Anzahl der Frachtstücke und die Verpackung bei der Übernahme durch den neuen Frachtführer äußerlich in gutem Zustand sind [59].

Der Ladeschein hat unter anderem die im Frachtbrief angeführten Punkte zu enthalten und muss vom Frachtführer unterschrieben werden. Er stellt somit nicht nur eine Beweisurkunde über den Abschluss des Frachtvertrages dar, sondern dient gleichzeitig als

- Empfangsbescheinigung für die übernommenen Güter in Zahl, Maß und Gewicht,
- Beförderungsversprechen,
- Ablieferungsversprechen gegenüber dem legitimierten Besitzer des Ladescheines.

Es existieren folgende Formen von BoLs:

- Das **Namenspapier** ist auf den Namen des Kunden ausgestellt und nur er ist zum Empfang der Güter berechtigt.
- Der **Orderladeschein** ist an eine Order, an einen Empfänger oder an dessen Order ausgestellt. Es ist derjenige berechtigt, der zuerst namentlich in der Order aufgeführt ist, oder der, der durch den Berechtigten bestimmt wird. Der Inhaber des Papiers kann die im Orderladeschein bezeichneten

[4] Der Begriff Konnossement kommt vom französischen „connaissement“ (Anerkennung).

Güter an einen anderen Empfangsberechtigten als neuen Eigentümer übertragen. Die Übertragung erfolgt per Indossament. Das Indossament ist eine Erklärung, mit welcher der Inhaber des Ladescheines die Rechte aus dem Ladeschein auf den (in einem schriftlichen Übertragungsvermerk genannten) Berechtigten überträgt. Es befindet sich üblicherweise auf der Rückseite des Orderpapiers. In der Praxis ist der Orderladeschein das übliche Dokument.

- Der **Inhaberladeschein** erlaubt dem Besitzer, den Transportweg der Güter nachzuvollziehen. Beim Inhaberladeschein ist es nicht notwendig, dass der Empfänger genannt wird.
- Der **Durchladeschein** wird immer dann ausgestellt, wenn an einem Transport mehrere Frachtführer beteiligt sind. Dabei übernehmen alle Frachtführer die Haftung gemäß der jeweilig entsprechenden Beförderungsbedingungen.

Nach dem deutschen Handelsgesetzbuch (§ 643 HGB) muss ein BoL mindestens folgende Aspekte beinhalten:

- Aussteller des Dokuments: Reeder (Verfrachter)
- Name und Nationalität des Schiffes
- Ablieferer: Exporteur, Ablader, Spediteur
- Empfänger; ggf. Benachrichtigungsadresse (notify address)
- Benennung von Verlade- und Bestimmungshafen
- Güterart und Warendeklaration
- Gewicht, Anzahl, Masse etc.
- Verpackung und Beschaffenheit der Ware, soweit erkennbar
- Bestimmung über die Fracht
- Empfangsbestätigung und Auslieferungsversprechen mit Ort und Tag
- Zahl der Ausfertigungen (Originale) [59]

Duplikate des Frachtbriefs werden an den Lieferanten, den Frachtführer und den Kunden geschickt. Er kann vom Layout her unterschiedlich aufgebaut sein enthält gemäß HGB § 408 folgende Informationen.

(1) Der Frachtführer kann die Ausstellung eines Frachtbriefs mit folgenden Angaben verlangen:

1. Ort und Tag der Ausstellung
2. Name und Anschrift des Absenders
3. Name und Anschrift des Frachtführers
4. Stelle und Tag der Übernahme des Gutes sowie die für die Ablieferung vorgesehene Stelle

5. Name und Anschrift des Empfängers und eine etwaige Meldeadresse
6. Übliche Bezeichnung der Art des Gutes und die Art der Verpackung, bei gefährlichen Gütern ihre nach den Gefahrgutvorschriften vorgesehene, sonst ihre allgemein anerkannte Bezeichnung
7. Anzahl, Zeichen und Nummern der Frachtstücke
8. Rohgewicht oder die anders angegebene Menge des Gutes
9. Vereinbarte Fracht und die bis zur Ablieferung anfallenden Kosten sowie einen Vermerk über die Frachtzahlung
10. Betrag einer bei der Ablieferung des Gutes einzuziehenden Nachnahme
11. Weisungen für die Zoll- und sonstige amtliche Behandlung des Gutes
12. Vereinbarung über die Beförderung in offenem, nicht mit Planen gedecktem Fahrzeug oder auf Deck

In den Frachtbrief können weitere Angaben eingetragen werden, die die Parteien für zweckmäßig halten.

(2) Der Frachtbrief wird in drei Originalausfertigungen ausgestellt, die vom Absender unterzeichnet werden. Der Absender kann verlangen, dass auch der Frachtführer den Frachtbrief unterzeichnet. Nachbildungen der eigenhändigen Unterschriften durch Druck oder Stempel genügen. Eine Ausfertigung ist für den Absender bestimmt, eine begleitet das Gut, eine behält der Frachtführer [16].

Container	**Stückgut**
Operator/Reeder	Operator/Reeder
Schiff	Schiff
Containernummer	Marke/Nummer
EDIFACT-Nachrichten-Nummer	Versendungs-/Ausfuhrland
Gestellungsdatum	Bestimmungsort
Bezugsnummer	Zollrechtlicher Status der Ware
Beförderungsmittel	Art der Packstücke
Kennzeichen Beförderungsmittel	Stückzahl (der Packstücke)
Anzahl Container	Gewicht (Rohmasse)
Versendungs-/Ausfuhrland	Warenbeschreibung
Bestimmungsort	
Zollrechtlicher Status der Ware	
Spezifischer Ordnungsbegriff	
Art der Packstücke	
Stückzahl (der Packstücke)	
Gewicht (Rohmasse)	
Warenbeschreibung	

Tabelle E.1: Angeführte Daten müssen für die Zollbehandlung vorliegen [38]

E.4 Was muss dem deutschen Zoll bei der Einfuhr von Containern gegeben werden?

Container	**Stückgut**
Operator/Reeder	Operator/Reeder
Schiff	Schiff
Reisenummer	Marke/Nummer
Containernummer	Netto-Gewicht
ISO-Code	Hafen
Netto-Gewicht	Final Destination
Leer/Voll	IMO/UN-Nummer
VA (Verkehrsart)	Maße (Länge/Höhe/Breite)
Hafen	Bemerkungen
Final Destination	Buchungsnummer
Stauplatz	B/Z-Nummer
Siegelnummer	MRN-Nummer
IMO/UN-Nummer	
Temperatur	
OH (Überhöhe)	
OW (Überbreite)	
OL (Überlänge)	
Bemerkungen	
Buchungsnummer	
B/Z-Nummer	
Zubringerschiff (Call Sign)	
Bei der Anlieferung der Container per Fuhre wird ein Anlieferschein benötigt (HDS, A08, Frachtdokument), aus dem die obigen Informationen entnommen werden können. Eine Anlieferung ohne Buchungsnummer ist nicht zulässig. Bei Anlieferung per Bahn können die Daten per über das HABIS-System übermittelten Bahnfrachtbrief zur Verfügung gestellt werden. Alle Daten/Informationen müssen in der CAL enthalten sein. Es erfolgt keine Verladung ohne gültige B/Z-Nummer.	Bei der Anlieferung wird ein Anlieferschein benötigt (HDS, A08, Frachtdokument) und mit Einführung ZAPP AES die MRN-Nummer. Eine Anlieferung ohne MRN-Nummer ist nicht zulässig.

Tabelle E.2: Verpflichtende Angaben bei der Einfuhr [38]

Zunächst findet die Anmeldung der zu verzollenden Waren über ein schriftliches Formular statt. Je nach Beförderungsmittel sind zur Kennzeichnung der Güter bestimmte Angaben erforderlich. So ist zum Beispiel bei der Beförderung auf dem Seeweg oder auf Binnenwasserstraßen der Name des Transport-Schiffes anzugeben. Dabei ist es notwendig, dass die Spediteure bereits zu einem früheren Zeitpunkt die Angaben aus Tabelle E.1 bekannt gemacht haben. Für jeden Transport, der die deutschen Ländergrenzen übertritt, müssen die Angaben aus Tabelle E.2 dem deutschen Zoll übergeben werden.

Angefügt seien noch Beispiele der verschiedenen Frachtdokumente.

Shipper

REX International Forwarding Co., Ltd.

IATA FMC # 7010014

Consignee
TO THE ORDER OF FORTIS BANK AS
KURUMSAL BANKACILIK SUBESI

BILL OF LADING

for combined transport
and port to port shipment

Notify party

Delivery Agent
LATEK LOJISTIK TICARET A.S.
AHI EVREN CADDESI NO.:1 ATA CENTER
KAT.4 80870 MASLAK-ISTANBUL/TURKEY
TEL:902122866620
FAX:902122866618

Place of receipt	Pre-carriage by	Combined transport-Pre-carriage from*	B/L-No.
Vessel MSC FINLAND / F731A	Port of loading SHENZHEN,CHINA	Movement CY/CY	REX863071487
Port of discharge DUBAI,U.A.E.	Final destination(for merchantis reference only) DUBAI,U.A.E.	Combined transport-Onward carriage to*	Freight payable at

Marks and numbers	Number of packages	Description of goods	Gross weight kgs	Measurement cbm
	875 CTNS	SHIPPER LOAD STOW AND COUNT AS PER PROFORMA INVOICE DD 20 JUNE 2007 NO:HL070620 QUANTITY: 3500 SETS DELIVERY TERM AND PLACE:FOB SHENZHEN CHINA AS PER INCOTERMS 2000. L/C REF NR:2676860700046	10500.00KGS	70.000CBM
MSCU8080251/4382779				ON BOARD DATE: 2007-08-05
SAY TOTAL :(1X40'HQ)	CONTAINER ONLY.			

ORIGINAL

Above particulars as declared by shipper.

* If the Pre-carriage or onward carriage boxes are filled out, shipment will be treated as Through combined transport.

Freight and Charges	Rate	Prepaid	Collect
FREIGHT PAYABLE AT DESTINATION			
Total Charges			

NOTICE : The laws and/or international Conventions applicable to Port to port Shipments are limiting by virtue and/or by incorportion into this Bill of Lading the Carrieris liability to certain amounts per package or customary freight unit(for Carriage to or from ports in the United States to a Maximum of US$500. per package or customary freight unit as per U.S. Carriage of Goods by Sea Act, 1936), unless the Merchant declares a higher cargo value below and pays the Carrieris ad valorem freight rate.

Declared Cargo value_____

If merchant enters a value, Carrieris per package Limitation of Liability Shall not apply and the ad valorem rate will be charged.

PLACE OF ISSUE: SZX

DATE OF ISSUE: 2007-08-05

In wintness whereof, the undersigned, signing on behalf of Rex International Forwarding (HK) Limited (three (3) bills of lading of the same tenor and date, one of which accomplished, the others to stand void.

REX INTERNATIONAL FORWARDING CO., LTD.

As Carrier

Abbildung E.2: Beispiel einer BoL

Ram Pacific

RAM PACIFIC LTD
Suite 3603, The Center,
99 Queen's Road Central, H.K.
香港皇后大道中 99 號
中 環 中 心 3603 室
Tel : (852) 2956 1331
Fax: (852) 2956 1330

Customs Invoice

Date	13-Aug-07
Number	CISI-036-07

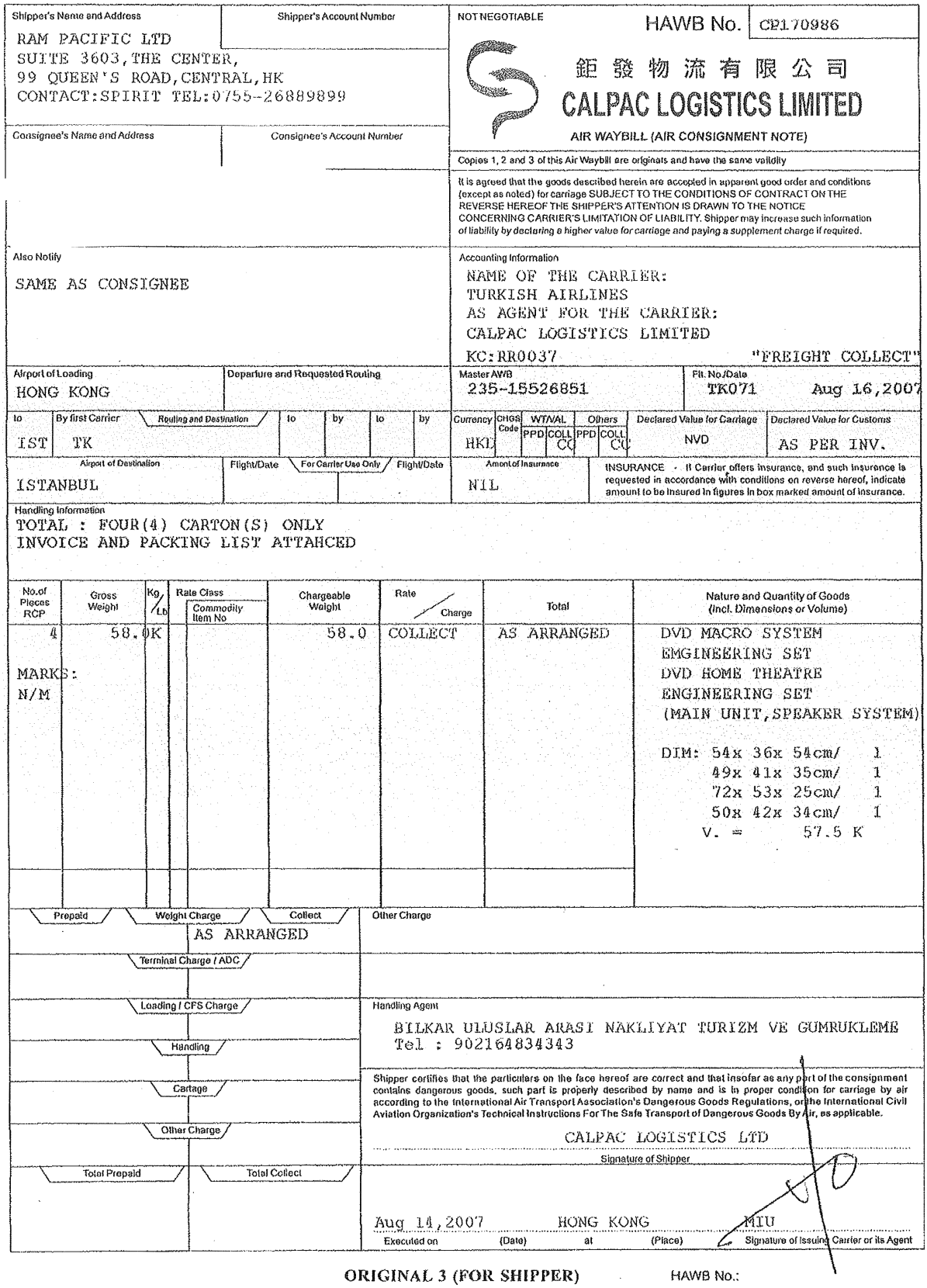
Description	Quantity	Unit Price (USD)	Total Amount (USD)
TOTAL: (4) CARTONS		ETD SHENZHEN, CHINA	
DVD MACRO SYSTEM ENGINEERING SET	2 PCS	$	$
DVD HOME THEATRE ENGINEERING SET			
MAIN UNIT	1 PCS	$	$
SPEAKER SYSTEM	1 PCS	$	$
Remark: Free of charge, The amount is for customs purpose only, RMA shipment			
TOTAL:			$

Free Of Charges -For Customs Purpose Only.

RAM PACIFIC LTD HONG KONG

KOC

Abbildung E.3: Beispiel einer Zolleinfuhr



Shipper's Name and Address
RAM PACIFIC LTD
SUITE 3603,THE CENTER,
99 QUEEN'S ROAD,CENTRAL,HK
CONTACT:SPIRIT TEL:0755-26889899

Shipper's Account Number

NOT NEGOTIABLE

HAWB No. CP170986

鉅發物流有限公司
CALPAC LOGISTICS LIMITED
AIR WAYBILL (AIR CONSIGNMENT NOTE)

Copies 1, 2 and 3 of this Air Waybill are originals and have the same validity

It is agreed that the goods described herein are accepted in apparent good order and conditions (except as noted) for carriage SUBJECT TO THE CONDITIONS OF CONTRACT ON THE REVERSE HEREOF THE SHIPPER'S ATTENTION IS DRAWN TO THE NOTICE CONCERNING CARRIER'S LIMITATION OF LIABILITY. Shipper may increase such information of liability by declaring a higher value for carriage and paying a supplement charge if required.

Consignee's Name and Address

Consignee's Account Number

Also Notify
SAME AS CONSIGNEE

Accounting Information
NAME OF THE CARRIER:
TURKISH AIRLINES
AS AGENT FOR THE CARRIER:
CALPAC LOGISTICS LIMITED
KC:RR0037 "FREIGHT COLLECT"

Airport of Loading	Departure and Requested Routing	Master AWB	Flt. No./Date
HONG KONG		235-15526851	TK071 Aug 16,2007

to	By first Carrier	Routing and Destination	to	by	to	by	Currency	CHGS Code	WT/VAL PPD	WT/VAL COLL	Others PPD	Others COLL	Declared Value for Carriage	Declared Value for Customs
IST	TK						HKD			CC		CC	NVD	AS PER INV.

Airport of Destination	Flight/Date	For Carrier Use Only	Flight/Date	Amount of Insurance	INSURANCE - If Carrier offers insurance, and such insurance is requested in accordance with conditions on reverse hereof, indicate amount to be insured in figures in box marked amount of insurance.
ISTANBUL				NIL	

Handling Information
TOTAL : FOUR(4) CARTON(S) ONLY
INVOICE AND PACKING LIST ATTAHCED

No.of Pieces RCP	Gross Weight	Kg/Lb	Rate Class / Commodity Item No	Chargeable Weight	Rate / Charge	Total	Nature and Quantity of Goods (Incl. Dimensions or Volume)
4	58.0K			58.0	COLLECT	AS ARRANGED	DVD MACRO SYSTEM ENGINEERING SET DVD HOME THEATRE ENGINEERING SET (MAIN UNIT,SPEAKER SYSTEM)
MARKS: N/M							DIM: 54x 36x 54cm/ 1 49x 41x 35cm/ 1 72x 53x 25cm/ 1 50x 42x 34cm/ 1 V. = 57.5 K

Prepaid | Weight Charge: AS ARRANGED | Collect

Other Charge

Terminal Charge / ADC

Loading / CFS Charge

Handling

Cartage

Other Charge

Total Prepaid | Total Collect

Handling Agent
BILKAR ULUSLAR ARASI NAKLIYAT TURIZM VE GUMRUKLEME
Tel : 902164834343

Shipper certifies that the particulars on the face hereof are correct and that insofar as any part of the consignment contains dangerous goods, such part is properly described by name and is in proper condition for carriage by air according to the International Air Transport Association's Dangerous Goods Regulations, or the International Civil Aviation Organization's Technical Instructions For The Safe Transport of Dangerous Goods By Air, as applicable.

CALPAC LOGISTICS LTD
Signature of Shipper

Aug 14,2007 (Date) HONG KONG (Place) MIU
Executed on at Signature of Issuing Carrier or its Agent

ORIGINAL 3 (FOR SHIPPER) HAWB No.:

Abbildung E.4: Beispiel einer Ladeliste

Ram Pacific

RAM PACIFIC LTD
Suite 3603, The Center,
99 Queen's Road Central, H.K.
香港皇后大道中 99 號
中 環 中 心 3603 室
Tel : (852) 2956 1331
Fax: (852) 2956 1330

To:

Date	13-Aug-07
Number	CISI-036-07

PACKING LIST

Shipping Marks	Mold No.	Mold No. & Description	Carton	Quantity	Gr. Wt. (KGS)	Meas. (cbm)
		DVD MACRO SYSTEM ENGINEERING SET	2	2 PCS	22.00	0.15
		DVD HOME THEATRE ENGINEERING SET				
		MAIN UNIT	1	1 PCS	15.00	0.11
		SPEAKER SYSTEM	1	1 PCS	22.00	0.10
			4	4 PCS	59.00	0.360

TOTAL: FOUR (4) CARTONS ONLY.

AS PER INVOICE NO. CISI-036-07 DATED 13 AUG., 2007

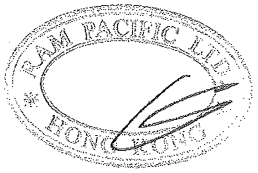


Abbildung E.5: Beispiel einer Packliste

Ram Pacific

To :
Attn
DATE : 16-Aug-07

SHIPPING ADVISE

BUYER NAME
SHIP MODE : Air
VESSEL/FLIGHT NAME : TK071
ETD HONGKONG : 16-Aug-07
TRANSIT TIME : 3-4DAY
MBL#/MAWB# : 235-15526851
HBL#/HAWB# : CP170986
FLIGHT CHARGE : COLLECT
PO/MOLD# : N/A
RAM INVOICE# : CISI-036-07
REMARK : N/A

Abbildung E.6: Beispiel einer Transportbenachrichtigung

Literaturverzeichnis

[1] ALAM ZAMORA, ANA LUISA: *XACML delegation for MASC information service*. Masterthesis, Technische Universität Hamburg-Harburg, Dezember 2007.

[2] ALOUSI, JALAL: *SPKI/SDSI basierte Zugriffskontrolle des MASC-Informationsdienst*. Studienarbeit, Technische Universität Hamburg-Harburg, April 2007. `http://www.sva.tu-harburg.de/projects/MASC/Paper/Alousi_SPKI.pdf`.

[3] BABIONE, ROBERT, CHANG KYUN KIM, EDDIE RHONE und ERIC SANJAYA: *Post 9/11 Security Cost Impact on Port of Seattle Import/Export Container Traffic*. Studienarbeit, University of Washington, Juni 2003. `http://depts.washington.edu/gttl/StudentPapersAbstracts/2003/SecurityCostImpactonPortofSeattle.pdf`.

[4] BANDMANN, OLAV, MADS DAM und BABAK SADIGHI FIROZABADI: *Constrained Delegations*. In: *Proceedings of 2002 IEEE Symposium on Security and Privacy*, Seiten 131–140, 2002. `http://www.sics.se/isl/pbr/papers/ConstraintDelegation.ps`.

[5] BARBIR, A., O. BATUNER, A. BECK, T. CHAN und H. ORMAN: *Policy, Authorization, and Enforcement Requirements of the Open Pluggable Edge Services (OPES)*. RFC 3838 (Informational), August 2004. `http://www.ietf.org/rfc/rfc3838.txt`.

[6] BARKER, ELAINE und JOHN KELSEY: *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*. NIST Special Publication 800-90, National Institute of Standards and Technology, März 2007. `http://csrc.nist.gov/publications/nistpubs/800-90/SP800-90revised_March2007.pdf`.

[7] BELLARE, MIHIR und PHILLIP ROGAWAY: *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols*. In: *1st ACM Conference on Computer and Communications Security*, Seiten 62–73, 1993. `http://citeseer.ist.psu.edu/bellare95random.html`.

[8] BIMCO, INTERCARGO, INTERNATIONAL CHAMBER OF SHIPPING, INTERNATIONAL SHIPPING FEDERATION und INTERTANKO: *ShippingFacts – Information about the international shipping industry*. Webseite, Mai 2006. `http://www.shippingfacts.com`.

[9] BLANCHET, BRUNO: *An Efficient Cryptographic Protocol Verifier Based on Prolog Rules*. In: *14th IEEE Computer Security Foundations Workshop (CSFW-14)*, Seiten 82–96, Cape Breton, Nova Scotia, Canada, Juni 2001. IEEE Computer Society.

[10] BLANCHET, BRUNO, MARTÍN ABADI und CÉDRIC FOURNET: *Automated Verification of Selected Equivalences for Security Protocols*. In: *20th IEEE Symposium on Logic in Computer Science (LICS 2005)*, Seiten 331–340, Chicago, IL, Juni 2005. IEEE Computer Society.

[11] BLEICHENBACHER, DANIEL: *Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1*. In: *Advances in Cryptology — CRYPTO '98*, Band 1462/1998 der Reihe *Lecture Notes in Computer Science*, Seiten 1–12. Springer, 1998. http://www.simovits.com/archive/pkcs.pdf.

[12] BLUNDO, CARLO, ALFREDO DE SANTIS, AMIR HERZBERG, SHAY KUTTEN, UGO VACCARO und MOTI YUNG: *Perfectly-Secure Key Distribution for Dynamic Conferences*. In: *Advances in Cryptology — CRYPTO '92*, Band 740/1993 der Reihe *Lecture Notes in Computer Science*, Seiten 471–486. Springer, 1993. http://citeseer.ist.psu.edu/blundo95perfectlysecure.html.

[13] BUNDESAMT FÜR SEESCHIFFFAHRT UND HYDROGRAPHIE: *Abwehr äußerer Gefahren auf See – Maritime Security*. Webseite, November 2005. http://www.bsh.de/de/Schifffahrt/Berufsschifffahrt/Gefahrenabwehr/.

[14] BUNDESAMT FÜR SEESCHIFFFAHRT UND HYDROGRAPHIE: *Risikobewertung für das Schiff*, November 2005. http://www.bsh.de/de/Schifffahrt/Berufsschifffahrt/Gefahrenabwehr/SSA%20und%20Anhaenge.pdf.

[15] BUNDESAMT FÜR SICHERHEIT UND INFORMATIONSTECHNIK: *E-Government-Handbuch*. Nummer Zweite Ergänzungslieferung. Bundesanzeigerverlag, 2003. http://www.bsi.bund.de/fachthem/egov/6.htm.

[16] BUNDESREPUBLIK DEUTSCHLAND: *Handelsgesetzbuch, §408 Frachtbrief*. Nummer Gesetz vom 10.5.1897, geändert durch Gesetz vom 5.1.2007. dejure.org, 2007. http://dejure.org/gesetze/HGB/408.html.

[17] BURDGE, RONALD: *4700 Mazda Cars Go Down the Drain*. Consumer Rights Law Blog, Webseite, September 2006. http://ohiolemonlaw.blogspot.com/2006/09/4700-mazda-cars-go-down-drain.html.

[18] CARTY, SHARON SILKE: *When cargo gets lost at sea firms can see big shortages, losses*. USA Today, 3. August: Seite 1, 2006. http://www.usatoday.com/money/world/2006-08-03-cargo-problems-usat_x.htm.

[19] CHAN, HAOWEN, ADRIAN PERRIG und DAWN SONG: *Random Key Predistribution Schemes for Sensor Networks*. In: *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, Seiten 197–215, 2003. http://citeseer.ist.psu.edu/chan03random.html.

[20] CHIPCON: *Chipcon CC2430 Data Sheet*. A True System-on-Chip solution for 2.4 GHz IEEE 802.15.4 / ZigBee, Gaustadalléen 21, N-0349 Oslo, Norway, Rev.no.: 1.02 Auflage, Oktober 2005. http://vast.uccs.edu/projects/traumagps_files/docs/chipcon/CC2430_Data_Sheet_1_02.pdf.

[21] CONTAINERHANDBUCH – FACHINFORMATION DER DEUTSCHEN VERSICHERER: *Der ISO-Standard 10368*. Webseite, 2005. http://www.containerhandbuch.de/chb/wild/index.html?/chb/wild/wild_07_02_09_03_03.html.

[22] COUNCIL, WORLD SHIPPING: *Liner Shipping: Facts and Figures*. Webseite. http://www.worldshipping.org/liner_shipping-facts\&figures.pdf.

[23] DAHLMAN, OLA, JENIFER MACKBY, BERNARD SITT, ANDRE POUCET, AREND MEERBURG, BERNARD MASSINON, EDWARD IFFT, MASAHIKO ASADA und RALPH ALEWINE: *Container Security – A Proposal for a Comprehensive Code of Conduct*. Online, Januar 2005. http://www.ndu.edu/ctnsp/Def_Tech/DTP9%20Container%20Security.pdf.

[24] DE HERTOGH, AUGUST A. und MARCEL LE NARD: *Flower Bulbs*. Agriculture Handbook, Nr. 66(United States Departement of Agriculture): Seiten 5–6, April 2004. http://usna.usda.gov/hb66/149flowerbulbs.pdf.

[25] DIERKS, T. und E. RESCORLA: *The Transport Layer Security (TLS) Protocol Version 1.1*. RFC 4346 (Proposed Standard), April 2006. http://www.ietf.org/rfc/rfc4346.txt. Updated by RFCs 4366, 4680, 4681.

[26] DMC'S CASENOTES: *The „Hyundai Fortune"*. Webseite, Singapore High Court decision, März 2004. http://www.onlinedmc.co.uk/the_hyundai_fortune.htm.

[27] DOLIVO, FRANCOIS: *The IBM Secure Trade Lane Solution*. ERCIM News 68, Traffic Planning and Logistics: Seiten 45–47, Januar 2007. http://ercim-news.ercim.org/content/view/98/256/.

[28] DOUCEUR, JOHN R.: *The Sybil Attack*. In: *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, Seiten 251–260, London, UK, März 2002. Springer.

[29] ELLISON, CARL: *RFC 2692 – SPKI Requirements*. Technischer Bericht, Network Working Group, September 1999. ftp://ftp.isi.edu/in-notes/rfc2692.txt.

[30] ELLISON, CARL, BILL FRANTZ, BUTLER LAMPSON, RON RIVEST, BRIAN THOMAS und TATU YLONEN: *RFC 2693 – SPKI Certificate Theory*. Technischer Bericht, Network Working Group, September 1999. ftp://ftp.isi.edu/in-notes/rfc2693.txt.

[31] ESCHENAUER, LAURENT und VIRGIL GLIGOR: *A Key Management Scheme for Distributed Sensor Networks*. Technischer Bericht, Electrical and Computer Engineering Department, University of Maryland, College Park, MD, USA, 2004. http://www.cs.umbc.edu/courses/graduate/CMSC691A/Spring04/papers/eschenauer02keymanagement.pdf.

[32] FIELDING, R., J. GETTYS, J. MOGUL, H. FRYSTYK, L. MASINTER, P. LEACH und T. BERNERS-LEE: *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616 (Draft Standard), Juni 1999. http://www.ietf.org/rfc/rfc2616.txt. Updated by RFC 2817.

[33] FRANZ, WALTER: *FleetNet – Internet on the Road*. Webseite, Januar 2002. http://www.et2.tu-harburg.de/fleetnet/pdf/FleetNet_Flyer.pdf.

[34] FREIE UND HANSESTADT HAMBURG: *Hafensicherheit: Fakten zu den internationalen Sicherheitsstandards, auf die sich alle Seeschiffahrt betreibenden Staaten verpflichtet haben*. Webseite, Pressemeldungen 26. Februar 2004. http://fhh.hamburg.de/stadt/Aktuell/pressemeldungen/2004/februar/26/2004-02-26-bfi-hafensicherheit.html.

[35] GERLACH, MATTHIAS, ANDREAS FESTAG, TIM LEINMÜLLER, GABRIELE GOLDACKER und CHARLES HARSCH: *Security Architecture for Vehicular Communication*. 4th International Workshop on Intelligent Transportation (WIT), Hamburg, März 2007. http://www.network-on-wheels.de/downloads/wit07secarch.pdf.

[36] GRUND-LUDWIG, PIA: *Sensoren schützen Container vor Diebstahl und Manipulation – Mit elektronischen Siegeln und Scannern begegnen Firmen den erhöhten Gefahren*. Handelsblatt, 15. Oktober 2007. Verlagsgruppe Handelsblatt – Economy.One.

[37] GUTIERREZ, J. A., M. NAEVE, E. CALLAWAY, M. BOURGEOIS, C. MITTER und B. HEILE: *IEEE 802.15.4: A developing standard for low-power low-cost wireless personal area networks*. IEEE Network, Seiten Seiten 12–19, September 2001. http://www.comsoc.org/livepubs/ni/Private/2001/Sep/gut.html.

[38] HAMBURGER HAFEN UND LOGISTIK AKTIENGESELLSCHAFT (HHLA): *Allgemeine Umschlagsbedingungen (AUB)*. Webseite, Februar 2007. http://www.hhla.de/de/Allgemeine_Umschlagsbedingungen.pdf.

[39] HERRY, MAX: *Transportpreise und Transportkosten der verschiedenen Verkehrsträger im Güterverkehr*. Studie, Österreichische Bundesarbeiterkammer, Büro Brüssel, 2001. http://eu.arbeiterkammer.at/pictures/d5/Herry.pdf. in german.

[40] HOUSLEY, R., W. POLK, W. FORD und D. SOLO: *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 3280 (Proposed Standard), April 2002. http://www.ietf.org/rfc/rfc3280.txt. Updated by RFCs 4325, 4630.

[41] HUANG, DIJIANG, MANISH MEHTA, DEEP MEDHI und LEIN HARN: *Location-aware key management scheme for wireless sensor networks*. In: *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, Seiten 29–42, New York, NY, USA, 2004. ACM Press.

[42] INTERNATIONAL MARITIME ORGANISATION: *Hinweise für den Zugang von Vertretern von Behörden, Notdiensten und Lotsen zu Schiffen.* Nachrichten für Seefahrer, Beilage

40: Seiten 1–5, 2005. http://www.bsh.de/de/Schifffahrt/Sportschifffahrt/Berichtigungsservice%20Karten/NfS/Schifffahrtsvorschriften/Beilagen-2005/Beilage40-2005.pdf. MSC/circ. 1156.

[43] INTERNATIONAL MARITIME ORGANISATION: *FAQ on ISPS code and maritime security*. Webseite, Januar 2006. http://www.imo.org/Newsroom/mainframe.asp?topic_id=897.

[44] JACOBSON, V., R. BRADEN und D. BORMAN: *TCP Extensions for High Performance*. RFC 1323 (Proposed Standard), Mai 1992. http://www.ietf.org/rfc/rfc1323.txt.

[45] KELSEY, JOHN, BRUCE SCHNEIER, DAVID WAGNER und CHRIS HALL: *Cryptanalytic Attacks on Pseudorandom Number Generators*. In: *Fast Software Encryption*, Band 1372/1998 der Reihe *Lecture Notes in Computer Science*, Seiten 168–188. Springer, März 1998. http://www.schneier.com/paper-prngs.html.

[46] KHALIL, ISSA und SAURABH BAGCHI: *SECOS: Key Management for Scalable and Energy Efficient Crypto On Sensors*. CERIAS Technical Report TR-2003-33, 2003. http://cobweb.ecn.purdue.edu/~dcsl/Publications/papers/secos_dsn04_submit.pdf.

[47] KLENSIN, J.: *Simple Mail Transfer Protocol*. RFC 2821 (Proposed Standard), April 2001. http://www.ietf.org/rfc/rfc2821.txt.

[48] KLÍMA, VLASTIMIL, ONDREJ POKORNÝ und TOMÁS ROSA: *Attacking RSA-Based Sessions in SSL/TLS*. In: *Cryptographic Hardware and Embedded Systems – CHES 2003*, Band 2779/2003 der Reihe *Lecture Notes in Computer Science*, Seiten 426–440. Springer, September 2003.

[49] LAUF, JENS OVE: *MASC – Monitoring and Security of Containers*. digma – Zeitschrift für Datenrecht und Informationssicherheit, Heft 4:160–163, Dezember 2007.

[50] LAUF, JENS OVE: *Containerüberwachung*. Spektrum – Das Magazin der Technischen Universität Hamburg-Harburg, Oktober, Juni 2008. (Geplante Veröffentlichung).

[51] LAUF, JENS OVE und DIETER GOLLMANN: *Monitoring and Securing of Containers*. In: *The First International Workshop on IT-Solutions for Physical Security*. Webseite, März 2006. http://www.sics.se/phys-sec/2006/dieterg.html.

[52] LAUF, JENS OVE und DIETER GOLLMANN: *MASC – Monitoring and Security of Containers*. In: *Proceedings of New Technologies, Mobility and Security (NTMS) 2007 Conference*, Band 1. Springer, Mai 2007.

[53] LAUF, JENS OVE, DIETER GOLLMANN und VOLKER TURAU: *MASC – Monitoring and Security of Containers*. In: *Informatik trifft Logistik (Band 1)*, Band Informatik 2007 der Reihe *GI Lecture Notes in Informatics (LNI)*. Gesellschaft für Informatik, 2007.

[54] LAUF, JENS OVE und HARALD SAUFF: *Secure Leightweight Tunnel for Monitoring Container Transports*. In: *IEEE Proceedings of the Third International Conference on Security and Privacy in Communication Networks*. IEEE, September 2007.

[55] LAW, YEE WEI, RICARDO CORIN, SANDRO ETALLE und PIETER HARTEL: *A Formally Verified Decentralized Key Management Architecture for Wireless Sensor Networks*. In: *Personal Wireless Communications*, 2003. http://citeseer.ist.psu.edu/law03formally.html.

[56] LAW, YEE WEI, JEROEN DOUMEN und PIETER HARTEL: *Survey and Benchmark of Block Ciphers for Wireless Sensor Networks*. ACM Transactions on Sensor Networks (TOSN), 2: Seite 24, 2006. http://www.ub.utwente.nl/webdocs/ctit/1/000000eb.pdf.

[57] LENSTRA, ARJEN K.: *Unbelievable Security Matching AES Security Using Public Key Systems*. In: *Advances in Cryptology — ASIACRYPT 2001*, Band 2248/2001 der Reihe *Lecture Notes in Computer Science*, Seiten 67–86, London, UK, 2001. Springer.

[58] LIU, DONGGANG und PENG NING: *Establishing Pairwise Keys in Distributed Sensor Networks*. In: *ACM Transactions on Information and System Security (TISSEC)*, Band 8, Seiten 52–61, 2003. http://citeseer.ist.psu.edu/liu03establishing.html.

[59] MAASSEN, C.: *Warenverkehr: Bill of Lading*. Webseite: Forum für Außenhandel, Juli 2005. http://www.trademind.de/modules/news/article.php?storyid=12.

[60] MENEZES, ALFRED J., PAUL C. VAN OORSCHOT und SCOTT A. VANSTONE: *Handbook of Applied Cryptography*, Band Fifth Printing (August 2001). CRC Press, Oktober 1996. http://www.cacr.math.uwaterloo.ca/hac/index.html.

[61] NECHVATAL, JAMES, ELAINE BARKER, LAWRENCE BASSHAM, WILLIAM BURR, MORRIS DWORKIN, JAMES FOTI und EDWARD ROBACK: *Report on the Development of the Advanced Encryption Standard (AES)*. Technischer Bericht, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Technology Administration, U.S. Department of Commerce, Oktober 2000. http://csrc.nist.gov/CryptoToolkit/aes/round2/r2report.pdf.

[62] NEUMAN, C., T. YU, S. HARTMAN und K. RAEBURN: *The Kerberos Network Authentication Service (V5)*. RFC 4120 (Proposed Standard), Juli 2005. http://www.ietf.org/rfc/rfc4120.txt. Updated by RFC 4537.

[63] NSA: *National Information Systems Security (INFOSEC) Glossary – NSTISSI*. National Security Telecommunications and Information Systems Security Committee, Nr. 4009, 1992. http://www.rsasecurity.com/rsalabs/faq/files/rsalabs_faq41.pdf.

[64] OASIS: *XACML v3.0 Administrative Policy Version 1.0*. Webseite, September 2007. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml#CURRENT.

[65] PADHY, PARITOSH, KIRK MARTINEZ, ALISTAIR RIDDOCH, H. L. ROYAN ONG und JANE K. HART: *Glacial Environment Monitoring using Sensor Networks*. In: *Online Proceedings of Real-World Wireless Sensor Networks*. Electronics and Computer Science, University of Southhampton, 2005.

[66] PALERO MONLLOR, ROSA ISABEL: *Key Management Protocols for Wireless Sensor Networks*. Diplomarbeit, Technische Universität Hamburg-Harburg, Oktober 2006. http://www.sva.tu-harburg.de/projects/MASC/Paper/rosabel_sensornetworks.pdf.

[67] PERRIG, ADRIAN, RAN CANETTI, J.D̃. TYGAR und DAWN SONG: *The TESLA Broadcast Authentication Protocol*. RSA CryptoBytes, Volume 5 (Summer), 2002. http://www.ece.cmu.edu/~adrian/projects/tesla-cryptobytes/tesla-cryptobytes.pdf.

[68] PERRIG, ADRIAN, ROBERT SZEWCZYK, VICTOR WEN, DAVID CULLER und J. D. TYGAR: *SPINS: Security Protocols for Sensor Networks*. Wireless Networks, Volume 8 (Nr. 5): Seiten 521–534, September 2002. http://sparrow.ece.cmu.edu/%7Eadrian/projects/mc2001/spins-wine-journal.pdf.

[69] POSTEL, J.: *Transmission Control Protocol*. RFC 793 (Standard), September 1981. http://www.ietf.org/rfc/rfc793.txt. Updated by RFC 3168.

[70] PRENEEL, B., A. BIRYUKOV, E. OSWALD, B. VAN ROMPAY, L. GRANBOULAN, E. DOTTAX, S. MURPHY, A. DENT, J. WHITE, M. DICHTL, S. PYKA, M. SCHAFHEUTLE, P. SERF, E. BIHAM, E. BARKAN, O. DUNKELMAN, J.-J. QUISQUATER, M. CIET, F. SICA, L. KNUDSEN, M. PARKER und H. RADDUM: *NESSIE security report*, Band 2.0. European Communities through the IST program under contract IST-1999-12324, Februar 2003. https://www.cosic.esat.kuleuven.be/nessie/deliverables/D20-v2.pdf.

[71] PRENEEL, B., A. BIRYUKOV, E. OSWALD, B. VAN ROMPAY, L. GRANBOULAN, E. DOTTAX, S. MURPHY, A. DENT, J. WHITE, M. DICHTL, S. PYKA, M. SCHAFHEUTLE, P. SERF, E. BIHAM, E. BARKAN, O. DUNKELMAN, J.-J. QUISQUATER, M. CIET, F. SICA, L. KNUDSEN, M. PARKER und H. RADDUM: *Performance of Optimized Implementations of the NESSIE Primitives*, Band 2.0. European Communities through the IST program under contract IST-1999-12324, Februar 2003. https://www.cosic.esat.kuleuven.be/nessie/deliverables/D21-v2.pdf.

[72] RADEMACHER, ROCHUS: *Container sind dauerwacht*. Computer Zeitung, Nr. 39 (Trends in der Forschung):Seite 15, Montag, 14. September 2007.

[73] RIES, JÜRGEN: *Haftungsfragen bei kombinierten Transporten*. GDV-Fachvortrag: Kombinierter Verkehr, Mai 1997. http://www.tis-gdv.de/tis/tagungen/workshop/kombi/ries/ries.htm.

[74] ROOS, HANS JÜRGEN: *Gefahrenabwehr (Terrorismus) im multimodalen internationalen Transport*. Transport-Fachvorträge des Gesamtverbandes der Deutschen Versicherungswirtschaft, International Ship and Port Facility Security Code (ISPS Code), Mai 2004. http://www.tis-gdv.de/tis/tagungen/workshop/2004/roos.pdf.

[75] RSA DATA SECURITY, INC.: *PKCS #5: Password-Based Encryption Standard*, Juni 1991. ftp://ftp.rsasecurity.com/pub/pkcs/ps/pkcs-5.ps.

[76] SADIGHI FIROZABADI, BABAK, MAREK SERGOT, und OLAV BANDMANN: *Using Authority Certificates to Create Management Structures*. In: *Security Protocols*, Band 2467/2002 der Reihe *Lecture Notes in Computer Science*, Seiten 134–145, Cambridge, UK, April 2001. Springer. http://www.sics.se/isl/pbr/papers/SecurityProtocol01.ps.

[77] SADIGHI FIROZABADI, BABAK und MAREK SERGOT: *Revocation in the Privilege Calculus*. In: *Proceedings of the 1st International Workshop on Formal Aspects in Security and Trust (FAST 2003)*, Seiten 39–51, September 2003. http://www.sics.se/isl/pbr/papers/fast03.pdf.

[78] SANTESSON, S.: *TLS Handshake Message for Supplemental Data*. RFC 4680 (Proposed Standard), Oktober 2006. http://www.ietf.org/rfc/rfc4680.txt.

[79] SANTESSON, S., A. MEDVINSKY und J. BALL: *TLS User Mapping Extension*. RFC 4681 (Proposed Standard), Oktober 2006. http://www.ietf.org/rfc/rfc4681.txt.

[80] SAUFF, HARALD: *Energieaufnahme von Kryptographiealgorithmen auf Sensorknoten*. Diplomarbeit, Technische Universität Hamburg-Harburg, September 2006. http://www.sauff.org/download/uni/diplomarbeit/da_hs06_energieaufnahme.pdf.

[81] SCHARNO, RENATE: *Ladungspflege und Risikofaktoren im Container Grundlagen und Schadenbeispiele*. Schadenverhütungstagung des Fachausschusses Transport im GDV vom 14. bis 16. Juni 2004, Juni 2004. http://www.tis-gdv.de/tis/tagungen/svt/svt03/09_scharnow.pdf.

[82] SCHIEDER, UWE-PETER und RENATE SCHARNOW: *Praktische Schadenbeispiele*. Fachtagung: Feuchtigkeit und ihre Auswirkungen auf den Warentransport, 1998. http://www.tis-gdv.de/tis/tagungen/svt/svt98/schieder/schieder.htm.

[83] SCHMITZ CARGOBULL: *Cargobull Telematics*. Pressemitteilung, Webseite, September 2004. http://www.schmitz-cargobull.de/images/120_Telematics_D_tcm10-492.pdf.

[84] SCHOOL OF ENGINEERING AND APPLIED SCIENCES, HARVARD UNIVERSITY: *Monitoring Volcanic Eruptions with a Wireless Sensor Network*. Webseite, November 2006. http://www.eecs.harvard.edu/~mdw/proj/volcano/.

[85] SECTRA, LLC.: *Ocean Web-AMS*. Webseite: WEB-Interface für das AMS-System, Juli 2007. http://www.tracking.com/Ocean-AMS.htm.

[86] SEREIKO, PAUL (SENSICAST SYSTEMS): *Monitoring Schools for IAQ During the Summer Months*. AutomatedBuilding.com, Webseite, Juni 2004. http://www.automatedbuildings.com/news/jun04/articles/sensicast/Sereiko.htm.

[87] SIMIC, SLOBODAN und SHANKAR SASTRY: *Distributed environmental monitoring using random sensor networks*. In: *Proceedings of Workshop on Information Processing in Sensor Networks*, Seiten 582–592, Palo Alto, California, April 2003. http://robotics.eecs.berkeley.edu/~simic/PDF/ipsn03.pdf.

[88] SIMIONESCU, MICHAEL: *Certificate Chain Discovery in SPKI/SDSI*. Diplomarbeit, Universität Dortmund, Fachbereich Informatik, Lehrstuhl VI, September 2003. http://ls6-www.informatik.uni-dortmund.de/issi/archive/literature/2003/Simionescu_2003.pdf#search=%22Certificate%20Chain%20Discovery%20in%20SPKI%2FSDSI%22.

[89] SOTO, JUAN: *Statistical Testing of Random Number Generators*. National Institute of Standards & Technology, 100 Bureau Drive, Stop 8930, Gaithersburg, MD, Proceedings of the 22nd National Information Systems Security Conference Auflage, Oktober 1999. http://csrc.nist.gov/rng/nissc-paper.pdf.

[90] SPEDITIONSBUCH: *Warenbegleitpapier*. Webseite, Juli 2007. http://speditionsbuch.info/spedition-show-1881.1.0-ladeliste.html.

[91] SUN MICROSYSTEMS: *XACML Implementierung*. Webseite, Sourceforge, Juni 2006. http://sunxacml.sourceforge.net/.

[92] SWEDISH INSTITUTE OD COMPUTER SCIENCE: *SICS's implementation of the XACML 3.0 draft*. Webseite, 2006. http://www.sics.se/spot/xacml_3_0.html.

[93] SWEDISH INSTITUTE OF COMPUTER SCIENCE: *Assertion Server Readme (Definitions)*. Webseite, September 2007. http://www.sics.se/spot/assertion_server/README.txt.

[94] TESKAN, SEVKAN: *Access Strategies for Monitoring Containers*. Masterthesis, Technische Universität Hamburg-Harburg, 21071 Hamburg, Germany, September 2007. http://www.sva.tu-harburg.de/projects/MASC/Paper/teskan_pullaccess.pdf.

[95] T-MOBILE DEUTSCHLAND: *T-Mobile showcases modern mobile communications solutions for the logistics sector at the IAA*. Webseite, September 2004. http://www.t-mobile.be/CDA/t-mobile_deutschland_newsdetails,1705,0,newsid-3368-yearid-2516-monthid-3333,en.html.

[96] TOLK, HENRIK: *Pseudozufallszahlen auf einem Scatterweb ESB Sensorknoten*. Bachelorarbeit, Technische Universität Hamburg-Harburg, 2007. Vorveröffentlichung.

[97] U.S. CUSTOMS AND BORDER PROTECTION: *Cargo security program aims to turn back a modern-day trojan horse*. Webseite: Customs Today, Dezember 2005. http://www.cbp.gov/xp/CustomsToday/2005/nov_dec/cargo_security.xml.

[98] U.S. CUSTOMS AND BORDER PROTECTION: *CSI In Brief*. Webseite, Januar 2006. http://www.cbp.gov/xp/cgov/border_security/international_activities/csi/csi_in_brief.xml.

[99] U.S. CUSTOMS AND BORDER PROTECTION: *Partnership to Secure the Supply Chain: Customs-Trade PartnershipAgainst Terrorism (C-TPAT)*. Webseite, Januar 2006. http://www.cbp.gov/xp/cgov/import/commercial_enforcement/ctpat/.

[100] U.S. DEPARTEMENT OF HOMELAND SECURITY – U.S. COAST GUARD: *Best Practice: 100% X-Ray Container Screening*. USCG Port Security Assessment – Best Practices Bulletin, Januar 2005. http://www.uscg.mil/hq/g-m/mp/pdf/Best%20Practice%20-%20Tunisia%20hundred%20percent%20x-ray%20screening.pdf. PFSO: Capt. Ben Hassan.

[101] WESTERINEN, A., J. SCHNIZLEIN, J. STRASSNER, M. SCHERLING, B. QUINN, S. HERZOG, A. HUYNH, M. CARLSON, J. PERRY und S. WALDBUSSER: *Terminology for Policy-Based Management*. RFC 3198 (Informational), November 2001. http://www.ietf.org/rfc/rfc3198.txt.

[102] YASAR, SIBEL: *Hardware-Zugriffsschutz für die MASC-Einheit*. Studienarbeit, Technische Universität Hamburg-Harburg, Juli 2007. http://www.sva.tu-harburg.de/projects/MASC/Paper/sibel_hardwaresecurity.pdf.

[103] YASAR, SIBEL: *Automatisierte und sichere Verwaltung von Ladepapieren im Containertransport*. Diplomarbeit, Technische Universität Hamburg-Harburg, Februar 2008.

[104] YAU, ARNOLD K. L., KENNETH G. PATERSON und CHRIS J. MITCHEL: *Padding Oracle Attacks on CBC-Mode Encryption with Secret and Random IVs*. In: *Fast Software Encryption*, Band 3557/2005 der Reihe *Lecture Notes in Computer Science*, Seiten 299–319. Springer, 2005. http://www.isg.rhul.ac.uk/~kp/padding.pdf.

[105] YAVATKAR, R., D. PENDARAKIS und R. GUERIN: *A Framework for Policy-based Admission Control*. RFC 2753 (Informational), Januar 2000. http://www.ietf.org/rfc/rfc2753.txt.

[106] ZESSACK, BJÖRN: *Implementierung und Optimierung von AES auf ScatterWeb-Sensorknoten*. Diplomarbeit, Technische Universität Hamburg-Harburg, 2007. (Vorveröffentlichung).

[107] ZHU, SENCUN, SANJEEV SETIA und SUSHIL JAJODIA: *LEAP: efficient security mechanisms for large-scale distributed sensor networks*. In: *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, Seiten 62–72, New York, NY, USA, 2003. ACM Press.

[108] ZIEHN, THOMAS: *Kriminalität im Containertransport*. Schadenverhütungstagung des Fachausschusses Transport im GDV, Juni 2004. http://www.tis-gdv.de/tis/tagungen/svt/svt03/08_ziehn.pdf.