

Network Architectures
and Services
NET 2012-08-1

**FI & IITM & AN
SS 2012**

**Proceedings of the Seminars
Future Internet (FI),
Innovative Internet Technologies and Mobile
Communications (IITM), and
Aerospace Networks (AN)**

Summer Semester 2012

Munich, Germany, 12.04.-13.07.2012

Editors

Georg Carle, Corinna Schmitt

Organisation

Chair for Network Architectures and Services
Department of Computer Science, Technische Universität München

Technische Universität München 





Network Architectures
and Services
NET 2012-08-1

FI & IITM & AN SS 2012

Proceedings zu den Seminaren Future Internet (FI), Innovative Internettechnologien und Mobilkommunikation (IITM), und Aerospace Networks (AN)

Sommersemester 2012

München, 13.04.-13.07.2012

Editoren: Georg Carle, Corinna Schmitt

Organisiert durch den Lehrstuhl Netzarchitekturen und Netzdienste (I8),
Fakultät für Informatik, Technische Universität München

Proceedings of the Seminars
Future Internet (FI) , Innovative Internet Technologies and Mobile Communications (IITM), and
Aerospace Networks (AN)
Summer Semester 2012

Editors:

Georg Carle
Lehrstuhl Netzarchitekturen und Netzdienste (I8)
Technische Universität München
D-85748 Garching b. München, Germany
E-mail: carle@net.in.tum.de
Internet: <http://www.net.in.tum.de/~carle/>

Corinna Schmitt
Lehrstuhl Netzarchitekturen und Netzdienste (I8)
Technische Universität München
D-85748 Garching b. München, Germany
E-mail: schmitt@net.in.tum.de
Internet: <http://www.net.in.tum.de/~schmitt/>

Cataloging-in-Publication Data

Seminars FI & IITM & AN SS2012
Proceedings zu den Seminaren „Future Internet“ (FI), „Innovative Internettechnologien und
Mobilkommunikation“ (IITM), und „Aerospace Networks“ (AN)
München, Germany, 12.04.-13.07.2012
Georg Carle, Corinna Schmitt
ISBN: 3-937201-27-0

ISSN: 1868-2634 (print)
ISSN: 1868-2642 (electronic)
DOI: 10.2313/NET-2012-08-1
Lehrstuhl Netzarchitekturen und Netzdienste (I8) NET 2012-08-1
Series Editor: Georg Carle, Technische Universität München, Germany
© 2012, Technische Universität München, Germany

Vorwort

Wir präsentieren Ihnen hiermit die Proceedings zu den Seminaren „Future Internet“ (FI), „Innovative Internettechnologien und Mobilkommunikation“ (IITM) und „Aerospace Networks“ (AN), die im Sommersemester 2012 an der Fakultät Informatik der Technischen Universität München stattfanden.

Im Seminar FI wurden Beiträge zu unterschiedlichen Fragestellungen aus den Gebieten Internettechnologien und Mobilkommunikation hinsichtlich Future Internet vorgestellt. Die folgenden Themenbereiche wurden abgedeckt:

- Prefix Hijacking-Angriffe und Gegenmaßnahmen
- Traceroute Anomalien
- Jenseits von WEP - WLAN Sicherheit
- „Device Fingerprinting“ mit dem Web-Browser
- Wassermarken in Sensordatensätzen
- TLS Lösungen für Sensornetze
- Wie funktioniert ein App Store/Markt?

Im Seminar IITM wurden Vorträge zu verschiedenen Themen im Forschungsbereich Internettechnologien und Mobilkommunikation vorgestellt. Der folgende Themenbereich wurde abgedeckt:

- Smart Energy Grids

Das Seminar AN ist eine Kooperationsveranstaltung mit EADS, Innovation Works, Dept. IW-SI - Sensors, Electronics & Systems Integration, On-Board Architectures & Networks. In diesem Seminar wurden Beiträge zu unterschiedlichen Fragestellungen aus den Gebiet „Aerospace Networks“ vorgestellt. Der folgende Themenbereich wurden abgedeckt:

- Evolution von Avionics Netzwerken von ARINC 429 bis AFDX

Wir hoffen, dass Sie den Beiträgen dieser Seminare wertvolle Anregungen entnehmen können. Falls Sie weiteres Interesse an unseren Arbeiten habe, so finden Sie weitere Informationen auf unserer Homepage <http://www.net.in.tum.de>.

München, August 2012



Georg Carle



Corinna Schmitt

Preface

We are very pleased to present you the interesting program of our main seminars on “Future Internet” (FI), “Innovative Internet Technologies and Mobile Communication” (IITM), and “Aerospace Networks” which took place in the summer semester 2012.

In the seminar FI we deal with issues of Future Internet. The seminar language was German, and the majority of the seminar papers are also in German. The following topics are covered by this seminar:

- Prefix Hijacking-Angriffe und Gegenmaßnahmen
- Traceroute Anomalies
- Beyond WEP - WLAN Security Revisited
- Device Fingerprinting with Web-Browser
- Watermarking in Sensor Data Sets
- TLS Solutions for Wireless Sensor Networks
- How does an App Store / Market work?

In the seminar IITM talks to different topics in innovative internet technologies and mobile communications were presented. The seminar language was German, and also the seminar papers. The following topic is covered by this seminar:

- Smart Energy Grids

The seminar „Aerospace Networks“ (AN) took place in cooperation with EADS, Innovation Works, Dept. IW-SI - Sensors, Electronics & Systems Integration, On-Board Architectures & Networks. The seminar language was German, and also the seminar papers. The following topic is covered by this seminar:

- The Evolution of Avionics Networks from ARINC 429 to AFDX

We hope that you appreciate the contributions of these seminars. If you are interested in further information about our work, please visit our homepage <http://www.net.in.tum.de>.

Munich, August 2012

Seminarveranstalter

Lehrstuhlinhaber

Georg Carle, Technische Universität München, Germany

Seminarleitung

Corinna Schmitt, Technische Universität München, Germany

Betreuer

Ralph Holz, *Technische Universität München, Wiss. Mitarbeiter I8*
Holger Kinkelin, *Technische Universität München, Wiss. Mitarbeiter I8*
Alexander Klein, *Technische Universität München, Wiss. Mitarbeiter I8*
Andreas Müller, *Technische Universität München, Wiss. Mitarbeiter I8*

Marc-Oliver Pahl, *Technische Universität München, Wiss. Mitarbeiter I8*
Johann Schlamp, *Technische Universität München, Wiss. Mitarbeiter I8*
Corinna Schmitt, *Technische Universität München, Wiss. Mitarbeiterin I8*
Stefan Schneelee, *EADS*

Kontakt:

{carle,schmitt,holz,kinkelin,klein,mueller,pahl,schlamp}@net.in.tum.de

Seminarhomepage

<http://www.net.in.tum.de/de/lehre/ss12/seminare/>

Inhaltsverzeichnis

Seminar Future Internet

Session 1: Sicherheit

| | |
|--|----|
| Prefix Hijacking-Angriffe und Gegenmaßnahmen | 1 |
| <i>Rafael Fedler (Betreuer: Johann Schlamp)</i> | |
| Traceroute Anomalies | 9 |
| <i>Martin Erich Jobst (Betreuerin: Johann Schlamp)</i> | |
| Beyond WEP - WLAN Security Revisited | 15 |
| <i>Rolf Sotzek (Betreuer: Holger Kinkel)</i> | |
| Device Fingerprinting mit dem Web-Browser | 23 |
| <i>Thomas Pieronczyk (Betreuer: Ralph Holz)</i> | |
| TLS Solutions for Wireless Sensor Networks | 31 |
| <i>Sebastian Wöhl (Betreuerin: Corinna Schmitt)</i> | |

Session 2: Anwendungen

| | |
|---|----|
| Watermarking in Sensor Data Sets | 39 |
| <i>Sebastian Wiendl (Betreuerin: Corinna Schmitt)</i> | |
| How does an App Store / Market work? | 47 |
| <i>Thomas Behrens (Betreuer: Marc-Oliver Pahl)</i> | |

Seminar Innovative Internettechnologien und Mobilkommunikation

| | |
|---|----|
| Smart Energy Grids..... | 57 |
| <i>Konrad Pustka (Betreuer: Andreas Müller)</i> | |

Seminar Aerospace Networks

| | |
|--|----|
| The Evolution of Avionics Networks from ARINC 429 to AFDX Network..... | 65 |
| <i>Christian M. Fuchs (Betreuer: Stefan Schnee, Alexander Klein)</i> | |

Prefix Hijacking-Angriffe und Gegenmaßnahmen

Rafael Fedler

Betreuer: Johann Schlamp, Dipl.-Inf.

Seminar Future Internet SS 2012

Lehrstuhl Netzarchitekturen und Netzdienste

Fakultät für Informatik, Technische Universität München

Email: fedler@in.tum.de

KURZFASSUNG

Das Border Gateway Protocol BGP, der de facto-Standard für Routing im Internet, wurde ohne Berücksichtigung von Sicherheitsaspekten entwickelt. Daher kann beim sogenannten Prefix Hijacking ein Angreifer sehr einfach ganze Adressbereiche Dritter übernehmen. Er ist dadurch in der Lage, große Teile des weltweiten Datenverkehrs für diese Adressbereiche zu sich umzuleiten. Als Folge hiervon leidet das Netz des Angegriffenen unter einem Verlust der Konnektivität, aber auch unerkannte Man-in-the-Middle-Angriffe gegen ganze Netze („Interception“) können durchgeführt werden. Diese Schwachstellen sind schon lange in der Theorie bekannt und wurden in der Praxis bereits viele Male ausgenutzt. Gegenmaßnahmen sind jedoch sehr schwer umzusetzen, weshalb bis heute keine Änderungen am BGP vorgenommen worden sind. Die vorliegende Arbeit gibt einen Einblick in die Schwachstellen des Border Gateway Protocol und stellt größere Vorfälle und Angriffe auf die weltweiten Routing-Systeme vor. Außerdem werden Ansätze zur Absicherung der globalen Routing-Infrastruktur vorgestellt. Hierbei wird sowohl auf Erkennungssysteme wie auch auf in Entwicklung befindliche Änderungen und Erweiterungen des BGP zur Beseitigung seiner Schwachstellen eingegangen.

Schlüsselworte

BGP, IP Prefix, Hijacking, Interception, MOAS, PHAS, iSPY, BGPsec, RPKI, S-BGP, soBGP, psBGP

1. EINLEITUNG

Das Internet besteht aus einer Vielzahl von eigenständigen Netzen, welche als Autonome Systeme (AS) bezeichnet werden. Bei der Datenübertragung zwischen Hosts in verschiedenen Netzen werden die Daten durch mehrere auf dem Weg liegende Netze geleitet, bevor sie ihr Ziel erreichen. Die Bestimmung der bevorzugten Route für die Datenpakete wird durch Routing-Protokolle wie das Border Gateway Protocol (BGP) vorgenommen. Dieses Protokoll ist der de facto-Standard für das Routing zwischen Autonomen Systemen (Exterior Gateway Protocol) [1]; für das Routing innerhalb von Autonomen Systemen werden Interior Gateway-Protokolle verwendet. Allerdings wurden bei der Entwicklung des BGP, ähnlich wie bei vielen frühen Netzwerkprotokollen, Sicherheitsaspekte unberücksichtigt gelassen. Das BGP bietet keinerlei Möglichkeiten, die Authentizität und Korrektheit übertragener Routing-Informationen zu überprüfen oder die Berechtigung für Urheberschaft von Adressbereichen festzustellen. Dies ermöglicht es Angreifern, die globale Routing-Tabelle zu manipulieren und fremde Adress-

bereiche zu kapern („Hijacking“). Dadurch sind sie in der Lage, den für ein angegriffenes AS bestimmten Verkehr ganz oder teilweise zum eigenen Netz umzuleiten. Die Konsequenzen können, abhängig von den Zielen des Angreifers, vielfältig sein, wie bspw. Verlust der Verfügbarkeit aus Zensurgründen oder Verlust der Vertraulichkeit der übermittelten Daten [2]. Als Reaktion auf solche Fälle, die in den vergangenen Jahren mehrfach aufgetreten sind und verstärkt als Problem wahrgenommen werden, wurden verschiedene Gegenmaßnahmen vorgeschlagen. Diese Verfahren bieten Möglichkeiten, Hijacking-Vorfälle zu erkennen oder zu verhindern.

Die vorliegende Arbeit gibt eine Einführung in die Hintergründe von IP Prefix Hijacking, die Schwachstellen von BGP sowie den Ablauf und die Folgen von größeren bekannten Angriffen. Darauf folgend werden Systeme zur Erkennung und Verhinderung vorgestellt und verglichen. Abschnitt 2 erläutert die technischen Grundlagen von Internet-Routing, Autonomen Systemen sowie dem Border Gateway Protocol. Auch werden die Voraussetzungen für IP Prefix Hijacking dargestellt und eine Klassifikation vorgenommen. Abschnitt 3 gibt einen Einblick in ausgewählte, größere Hijacking-Vorfälle. Danach werden in Abschnitt 4 Präventiv- und Gegenmaßnahmen vorgestellt und evaluiert. Abschnitt 5 gibt einen Überblick über verwandte Arbeiten. Abschließend folgen Zusammenfassung und Fazit der Arbeit.

2. GRUNDLAGEN

Das Internet wird durch aktuell mehr als 40.000 eigenständige, miteinander verbundene Netze gebildet [3]. Diese werden als Autonome Systeme (AS) bezeichnet. Sie befinden sich unter der Kontrolle jeweils genau einer Organisation und werden durch eine eindeutige Nummer identifiziert. Jedes AS veröffentlicht außerdem die ihm zugewiesenen IP-Adressbereiche, die sogenannten Präfixe, und teilt sie seinen benachbarten Routern mit.

Bei der Übertragung eines Datenpakets von einem Quell-Host in einem AS zu einem Ziel-Host in einem anderen AS passiert dieses mehrere Zwischenstationen („Hops“), im Durchschnitt 16 [4]. Um sicherzustellen, dass ein Paket sein Ziel erreicht, kommen Routing-Protokolle zum Einsatz. Hierbei sind Interior Gateway-Protokolle für das Routing innerhalb eines AS („Intra-Domain Routing“) und Exterior Gateway-Protokolle für das Routing zwischen den AS zuständig („Inter-Domain Routing“). Auch wird versucht, die Anzahl an Hops zu minimieren, um das Routing schnell und effizient durchzuführen.

Das Border Gateway Protocol (BGP) hat sich als Stan-

dard für das Inter-Domain Routing etabliert. Es dient dem Austausch von Routen-Informationen zwischen Routern verschiedener Netze. Als Pfad-Vektor-Protokoll speichert es den gesamten Pfad von einem Router zu jedem Ziel-AS. Hierfür teilt jedes AS die ihm zugewiesenen Präfixe seinen Nachbar-routern in BGP-Update-Nachrichten mit. Da die Nachbar-router sowohl die ihrem AS zugewiesenen Präfixe wie auch die ihnen bekannten AS und zugehörige Präfixe wiederum ihren Nachbarn mitteilen, konvergieren die BGP-Tabellen in allen Routern gegen den selben Informationsstand. Vor der Weitergabe der eigenen bekannten Routen hängt sich das jeweilige AS vorne an jeden Eintrag an. Somit wird sichergestellt, dass jeder Router die Pfade zu allen Zielen kennt. Dies soll an dem in Abbildung 1 dargestellten Beispielnetz verdeutlicht werden:

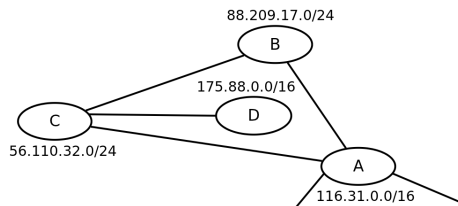


Abbildung 1: Beispiel-Netz

Die Routing-Tabelle für Router A gestaltet sich hierbei wie in Tabelle 1:

Tabelle 1: Routing-Tabelle von Router A

| Präfix | Pfad |
|----------------|-----------|
| 116.31.0.0/16 | A |
| 88.209.17.0/24 | A → B |
| 56.110.32.0/24 | A → C |
| 175.88.0.0/16 | A → C → D |
| ... | ... |

Die Tabelle wird in dieser Form an alle Nachbar-Router von A weitergegeben, die sich an alle Pfade vorne anfügen und die Tabelle danach ebenfalls weiterverbreiten. Auf diese Weise werden weltweit jedem BGP-Router die Pfade zu den Autonomen Systemen hinter den Routern B, C und D bekannt. Die Metrik, welche die Effizienz des Routings sicherstellt, wählt den Weg der geringsten Hop-Anzahl. Aus diesem Grund ist die Route von A nach D nicht A → B → C → D, selbst wenn diese Pfadinformation A früher erreicht hat als die von Router C.

Treffen mehrere Präfixe auf eine Ziel-Adresse zu, gilt die Longest Prefix Match-Regel. Diese besagt, dass das spezifischste Präfix als korrektes Ziel-System angenommen werden soll [5].

2.1 Schwachstellen des BGP

Viele ältere Netzwerkprotokolle wurden ohne Berücksichtigung von Sicherheitsaspekten entwickelt, wie z.B. DNS, SMTP oder auch IPv4. Mechanismen, die die Authentizität, Integrität und Vertraulichkeit von übermittelten Informationen sicherstellen, sind nicht integriert. Aus diesem Grund wurden die Protokolle weiterentwickelt oder erweitert. Für das Domain Name System wurde DNSSEC entwickelt, welches eine hierarchische Public Key-Infrastruktur (PKI) für

DNS-Einträge etabliert. IPv6 wurde um IPsec, Internet Protocol Security, erweitert. S/MIME und PGP sind Verfahren, die die Nutzdaten von SMTP verschlüsseln. Anwendungsprotokolle, die selber keine Verschlüsselung bieten, können auf TLS aufsetzen und damit Integrität und Vertraulichkeit, optional auch Authentizität, gewährleisten.

Die genannten Defizite im Protokolldesign treffen auf das Border Gateway Protocol ebenfalls zu. Insbesondere kann ein AS ohne Weiteres jedes Präfix als zu sich gehörend proklamieren und sich somit als „Origin AS“, also als das für das Präfix autoritative AS, ausgeben. Eine Authentizitätsüberprüfung findet nicht statt. Beanspruchen mehrere AS, das Origin AS eines Präfixes zu sein, spricht man von einem Multiple Origin AS-Konflikt, abgekürzt MOAS-Konflikt.

Weitere Probleme entstehen durch die mangelnde Sicherheit auf Transportebene. BGP setzt auf TCP auf und ist somit anfällig für übliche Angriffe auf TCP. Insbesondere können mit RST-Paketen gezielt Verbindungen zwischen BGP-Routern abgebrochen und permanent gestört werden. Somit kann ein Angreifer Einfluss auf die Verbreitung von Routen-Informationen und wahrgenommene Erreichbarkeit von Autonomen Systemen nehmen [1].

2.2 IP Prefix Hijacking und Interception

Da weder eine Überprüfung der Validität, noch des Ursprungs der Routen-Informationen vorgenommen wird, kann jeder BGP-Router beliebige Routen-Informationen veröffentlichen. Diese werden von seinen Nachbarn als Route gewählt, wenn die gelernten Pfade bevorzugbar erscheinen, und wiederum an ihre Nachbarn weiterverbreitet. Auf diese Weise kann ein Angreifer große Teile der globalen Routing-Tabelle manipulieren. Insbesondere ist er dazu in der Lage, beliebige IP-Adress-Präfixe selber zu annoncieren. Abbildung 2 verdeutlicht dies.

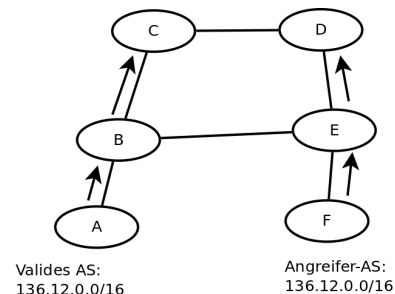


Abbildung 2: IP Prefix Hijacking

Im vorliegenden Beispiel übernehmen die Router der Autonomen Systeme D und E die falschen BGP-Updates des Angreifers F, da die Anzahl an Hops für sie minimal ist. Für B und C ist die Route nach A, dem legitimen Besitzer des Präfix 136.12.0.0/16, kürzer als nach F. Sie übernehmen daher die falschen Pfad-Informationen, die sie über D und E erhalten, nicht. Als Konsequenz werden alle Router, die die gefälschten Routen-Informationen übernehmen, sämtlichen an das annoncierte Präfix adressierten Verkehr zum Netz F des Angreifers leiten. An das Präfix 136.12.0.0/16 adressierte Pakete aus den Netzen D und E erreichen somit nicht den legitimen Empfänger in Netz A, sondern werden zu Netz F geschickt.

2.2.1 Typologie von Hijacking-Angriffen

Hijacking-Angriffe können auf verschiedene Weisen durchgeführt werden. Es ergeben sich hauptsächlich drei Vorgehensweisen [6]:

Um **vollständiges Prefix-Hijacking** handelt es sich, wenn wie im Beispiel in Sektion 2.2 ein Angreifer sich als legitimer Besitzer des gesamten angegriffenen Präfix ausgibt. Die Verbreitung der falschen Routen-Information wird durch die Metrik der kürzesten Pfade limitiert. Es wird nur ein begrenzter Teil des Internets durch die falschen Pfad-Informationen manipuliert.

Subprefix-Hijacking nutzt die Longest Prefix Match-Regel aus. Gibt sich ein Angreifer als Besitzer kleinerer Teilnetze des anzugreifenden Adressraumes aus und annonciert diese über das BGP, werden andere Router diese bevorzugen. Ein Großteil aller BGP-Router wird somit diese Route wählen und weitergeben.

Bei **Interception** verwendet ein Angreifer einzelne andere Autonome Systeme, deren Routen-Informationen nicht manipuliert wurden. Sie dienen ihm dazu, den abgefangenen Verkehr zurück an das Hijacking-Opfer zu leiten. Sie sind somit der Rückkanal eines BGP-basierten Man-in-the-Middle-Angriffs.

Dass BGP auf TCP aufsetzt, kann ein Angreifer zu seinem Vorteil nutzen. Beispielsweise kann er die Verbreitung seiner falschen BGP-Announcements beschleunigen, indem er BGP-Verbindungen, über die legitime Pfad-Informationen übertragen werden, via RST-Nachrichten kappt. Ferner kann er den Datenverkehr über bestimmte Routen zwingen, indem er die BGP-Verbindungen zu vermeidender AS auf die selbe Weise stört.

2.2.2 Folgen

Abhängig von der Methodik und den Motiven eines Angreifers haben Hijacking-Angriffe verschiedene Konsequenzen. Folgende Schutzziele können durch Hijacking-Angriffe betroffen sein:

- **Verfügbarkeit:** Werden nur die Routing-Tabellen illegitim ohne weitergehendes Wirken des Angreifers manipuliert, ist die Konnektivität des Opfer-AS beeinträchtigt. Da Antworten auf Pakete, die aus dem Opfer-Netz verschickt wurden, nicht wieder zurück in das betroffene AS geleitet werden, ist das Netz nicht mehr erreichbar. Dies gilt zumindest für Kommunikation mit allen Netzen, die die falschen Pfad-Informationen des Angreifers übernommen haben, oder auf deren Pfad sich manipulierte Netze befinden. Erreicht der Verkehr sein Ziel nicht und wird vom angreifenden System verworfen, spricht man von *Blackholing*. Wird solch ein Angriff bewusst durchgeführt, kann die Motivation bspw. ein Denial of Service-Angriff oder Zensur sein.
- **Authentizität:** Übernimmt ein Angreifer die Adressbereiche fremder Organisationen, nehmen dritte Parteien und Systeme an, er sei die angegriffene Organisation. Versendet er beispielsweise aus dem Adressbereich einer Firma Spam, kann dies den Ruf der Firma schädigen. Sie wird ggfs. auch noch nach Ende des Angriffs länger auf Spam-Blacklists enthalten sein. Ferner können fremde Dienste vorgetäuscht werden. Dies geschieht unerkennbar für Nutzer. Sie erhalten bei Auflö-

sung von Domain-Namen durch das DNS zwar die korrekte IP-Adresse zurück, ihre Anfrage wird aber zum System des Angreifers geleitet.

- **Vertraulichkeit und Integrität:** Bei allen genannten Angriffsformen ist ein Angreifer in der Lage, den an das angegriffene Netz adressierten Verkehr abzufangen und zu lesen. Somit kann ein Angreifer potentiell vertrauliche oder geschäftskritische Daten erhalten. Auch kann er Systeme in seinem Netz als legitime Systeme im angegriffenen Netz ausgeben, um bidirektionale Kommunikation zu ermöglichen. Authentifikationsdaten, wie bspw. bei Challenge-Response-Verfahren preisgegeben, können auf diese Weise abgefangen werden. Ebenso können Spoofing- und Phishing-Angriffe durchgeführt werden. Alternativ kann er als Man in the Middle bei Interception-Angriffen die Integrität übertragener Daten verletzen, indem er Pakete manipuliert.

3. BEKANNTE VORFÄLLE

Die Sicherheitsdefizite des BGP und ihre Implikationen sind in der Theorie bereits sehr lange bekannt [7]. Auch in der Praxis wurden sie bereits mehrfach ausgenutzt. Vorfälle unterschiedlichen Ausmaßes und unterschiedlicher Motivation wurden von diversen Parteien verursacht. Im Folgenden werden die bedeutendsten vorgestellt. Eine umfassendere Liste ist in [2] zu finden.

3.1 AT&T WorldNet

Dezember 1999. Ein anderer Internet Service Provider (ISP) veröffentlicht versehentlich falsche Routen-Informationen zu Einwahlservern des ISP AT&T WorldNet. Es kommt zu Blackholing, wodurch 1,8 Millionen Kunden einen Tag lang keinerlei Internetzugriff haben. [2]

3.2 Spammer – Northrop Grumman

14. Mai, 2003. Ein großer Teil des ungenutzten Adressbereichs der Rüstungsfirma Northrop Grumman wird von Spammern via falscher BGP-Updates übernommen. Im Vorfeld hatten die Spammer im DNS die Domain, die als Kontaktadresse für diesen Adressbereich angegeben war, unbemerkt auf sich neu registriert. Von dieser Domain aus schickten sie nun eMails an ihren Provider und gaben sich als legitime Besitzer des Adressbereichs aus. Der Provider versendete in der Folge BGP-Announcements, die den entsprechenden Adressbereich von Northrop Grumman dem Netz der Spammer zuwies. Es dauerte zwei Monate, bis Northrop Grumman den Adressbereich zurückerlangen konnte. In der selben Zeit wurden enorme Mengen an Spam versendet, sodass der Adressbereich auf nahezu allen Spam-Blacklisten landete. [8]

3.3 Malaysia – Yahoo

Mai 2004. Der malaysische ISP DataOne annonciert die verwendeten Präfixe eines Rechenzentrums von Yahoo, woraufhin dieses aus weiten Teilen des Internets nicht erreichbar ist. [8]

3.4 Türkei – Internet

24. Dezember, 2004. Der türkische ISP TTNNet veröffentlicht via BGP Routen für das gesamte Internet zu seinem Netz, höchstwahrscheinlich wegen einem Konfigurationsfehler. Telecom Italia hatte eine BGP-Session mit einem BGP-Router

von TTNNet, aber kein Limit für die Anzahl zu übernehmender Routen (**MAXIMUM_PREFIX**-Attribut). Die Telecom Italia übernahm somit sämtliche falschen Routen-Informationen, die TTNNet veröffentlichte. Weitere Provider sowohl in Europa wie auch in den USA, die wiederum BGP-Sessions mit Telecom Italia hatten, verbreiteten die falschen Routen-Informationen weiter. Die großen nordamerikanischen ISPs Verizon, Sprint und Hurricane hatten hieran einen sehr hohen Anteil. In der Konsequenz waren ihre Kunden für die Dauer des Falls nicht erreichbar. Betroffen waren sowohl Privat- wie auch Firmenkunden, die über die genannten Provider angebunden sind. Hierzu gehören u.a. Microsoft, Amazon und Yahoo. Der Vorfall dauerte insgesamt einen ganzen Tag, währenddessen der Großteil des Internets nicht oder nur teilweise erreichbar war. [9, 10]

3.5 DNS Root-Server L

November 2007 bis Mai 2008. Bis November 2007 befand sich der Root-Server L des DNS in einem Präfix, welches nicht zum Adressraum der ICANN gehörte. Am 1. November 2007 zog die ICANN die Adresse des Root-Servers in ihren eigenen Adressbereich um. In den darauf folgenden Monaten erschienen mehrere falsche L-Root-Server. Die Betreiber dieser Server veröffentlichten inkorrekte Routen-Informationen, die die alte Adresse des DNS-Root-Servers L ihren Netzen zuwiesen. Mangelnde Filterung ihrer Provider hatte zur Folge, dass sich diese falschen Routen-Informationen weiterverbreiteten. Da diese falschen Server nach außen hin keine erkennbare bössartige oder fehlerhafte Funktionalität aufwiesen, ist die Motivation der Betreiber unklar. [11, 12] Kontrolle über DNS-Server ermöglicht es Angreifern, die Auflösung von beliebigen Adressen zu steuern und so Nutzer auf falsche Server umzuleiten, um bspw. Authentifikationsdaten abzufangen. Auch können DNS-Anfragen mitgeschnitten werden, um das Verhalten der anfragenden Clients zu überwachen. Außerdem bedienen sich bereits im Einsatz befindliche Zensursysteme manipulierter DNS-Einträge [13].

3.6 Pakistan – YouTube

24. Februar 2008. Pakistan Telecom blockiert auf Basis von BGP YouTube, um die Webseite auf nationaler Ebene zu zensieren. Allerdings werden die falschen Routen unbeabsichtigt weltweit weiterverbreitet, weshalb YouTube nahezu global für zwei Stunden nicht erreichbar ist. Bei diesem Vorfall handelte es sich um Subprefix-Hijacking, da YouTube 208.65.152.0/22 annouciert, während das angreifende System 208.65.153.0/24 proklamiert. Wegen der Longest Prefix Match-Regel wird die Route nach Pakistan global bevorzugt. Als Gegenmaßnahme bewirbt YouTube nun erst 208.65.153.0/24 und darauf folgend 208.65.153.128/25 sowie 208.65.153.0/25. Auf Grund der Longest Prefix-Regel werden nun wieder die korrekten Routen global bevorzugt. [14]

3.7 DEFCON Proof of Concept

Auf der DEFCON 16 wurde 2008 ein Proof of Concept für BGP-basierte Interception geführt. Dies ist zwar kein reell aufgetretener Vorfall, es handelt sich allerdings um eine praktische Demonstration von Prefix Hijacking zum Durchführen einer BGP-basierten Man-in-the-Middle-Attacke. Hierbei kann ein Angreifer möglichst viele BGP-Router dazu bringen, das Netz des Angreifers für das Netz des Angegriffenen zu halten – mit Ausnahme genau einer Route. Diese

Route hält sich der Angreifer gezielt als Rückkanal offen. Somit leiten alle betroffenen Systeme ihren Verkehr für das angegriffene System zum Angreifer, und dieser den Verkehr transparent über den Rückkanal zum Opfer des Angriffs.

Um dies zu ermöglichen, nutzt der Angreifer eine Eigenschaft des BGP zur Schleifenverhinderung. Findet ein AS im Pfad, der in einem BGP-Update angegeben ist, seine eigene AS-Nummer, verwirft es dieses Update. Zur Ermittlung des Rückkanals führt der Angreifer ein Traceroute in Richtung seines Ziels durch. Er übernimmt alle AS auf der Route nun in sein gefälschtes BGP-Update, welches den Pfad zum zu kapernenden Adressbereich angibt. Zur Schleifenverhinderung werden die angegebenen AS dieses BGP-Update ignorieren. Somit besteht für die AS entlang des Pfades vom Angreifer zum Angegriffenen weiterhin die legitime Route. Die BGP-Router aller anderen AS schicken den an das Opfer-Netz adressierten Verkehr nun zum Angreifer. Dieser Angriff wurde auf der DEFCON auch praktisch demonstriert. [15]

3.8 China – USA

Am 8. April 2010 annouciert ein kleiner chinesischer ISP, IDC China Telecommunication, Routen zu mehr als 8.000 US-amerikanischen Präfixen, 1.100 australischen und 230 von France Telecom, sowie weitere. Diese Routen werden ungefiltert von der staatlichen China Telecom übernommen und propagiert. Die Deutsche Telekom, AT&T, Level3 und andere Provider übernehmen diese Routen ebenfalls und verbreiten sie wiederum weiter. Als Folge dessen enden für 20 Minuten 15% aller Routen weltweit in China. Betroffen sind hiervon sowohl privatwirtschaftliche Unternehmen wie auch viele militärische und Regierungs-Institutionen der USA. Jedoch tritt, im Gegensatz zu bisherigen Vorfällen, kein Blackholing auf, da der Verkehr auf nicht beeinträchtigten Routen zurück in die USA geleitet wird. Der Vorfall wird durch die U.S.-China Economic and Security Review Commission des Kongresses der USA als Hijacking bezeichnet [16].

4. GEGENMAßNAHMEN

Obwohl die Sicherheitsdefizite des BGP bereits seit 1998 bekannt sind und seitdem mehrere Vorfälle und Angriffe auf die globale Routing-Infrastruktur auftraten, sind die Schwachstellen bis heute nicht korrigiert worden. Dies liegt hauptsächlich darin begründet, dass eine Transition auf eine kryptographisch abgesicherte Version des BGP, ähnlich DNSSEC, kompliziert und im Betrieb außerordentlich ressourcenintensiv ist. Aus diesem Grund wurden behelfsmäßig Systeme entwickelt, die dazu dienen, Angriffe frühzeitig zu erkennen und manuell Gegenmaßnahmen ergreifen zu können. Im Folgenden werden mit PHAS und iSPY solche reaktiven Erkennungssysteme vorgestellt, wie auch Ansätze zur präventiven Korrektur der Sicherheitsdefizite des BGP.

4.1 Erkennungssysteme

4.1.1 PHAS

Um Hijacking-Vorgänge global und in Echtzeit zu erkennen, bezieht das Prefix Hijacking Alert System PHAS BGP-Datenströme von mehreren Dutzend bis Hundert BGP-Routern. Diese Datenströme werden durch BGP-Monitoring-Systeme wie Route Views, BGPmon oder die Routing Information Services von RIPE aggregiert. PHAS nimmt also eine möglichst globale Perspektive ein. Es wird stellvertretend für verschiedene vorherige Erkennungssysteme behandelt, da

seine Funktionsweise weitestgehend ähnlich ist. PHAS unterhält für jedes Präfix, welches via BGP propagiert wird, eine Menge an Autonomen Systemen. Diese Menge beinhaltet genau solche Autonome Systeme, welche dieses Präfix initial annonciieren, also die Origin AS. Die Menge wird als Vereinigung aller gemeldeter Origin AS von allen Routern gebildet, die Daten an den Monitoring-Dienst wie Route Views, BGPmon oder RIPEs RIS liefern. Diese Vereinigungsmenge aller weltweit observierten Origin AS wird *Origin Set*, formell O_{SET} , genannt. Für jedes annoncierte Präfix überwacht PHAS dieses Origin Set auf Änderungen. Dies soll durch folgendes Beispiel verdeutlicht werden.

Das Präfix 123.98.76.0/24 gehöre dem AS 5678. Zu einem beliebigen Zeitpunkt t sei dieses AS 5678 das einzige AS, welches das Präfix 123.98.76.0/24 annonciert. Somit beobachten alle BGP-Router, die als Datenquellen für BGP-Beobachtungsdienste dienen, einzig das AS 5678 als Origin AS für das genannte Präfix. Also gilt zum Zeitpunkt t : $O_{SET} = \{5678\}$. Führe nun zu einem beliebigen späteren Zeitpunkt $t' > t$ das AS 6666 einen Hijacking-Angriff durch, indem es das selbe Präfix annonciert. Sobald der erste BGP-Router des Monitoring-Systems Kenntnis dieser neuen Route erhält und sie übernimmt, da sie vorteilhaft erscheint, ändert sich O_{SET} . Das neue O'_{SET} beinhaltet nun auch das AS 6666: $O'_{SET}(123.98.76.0/24) = \{5678, 6666\}$. Unter der Annahme, dass die Route überall als günstiger betrachtet wird, wird nach der Konvergenz der globalen Routing-Tabelle zum Zeitpunkt t'' das legitime Origin AS 5678 aus dem O'_{SET} entfernt, da kein BGP-Router es mehr in seiner Routing-Tabelle hält. Somit ist zum Zeitpunkt t'' das Origin Set $O''_{SET}(123.98.76.0/24) = \{6666\}$. Wird der Angriff bemerkt und Gegenmaßnahmen ergriffen, ändert sich das Origin Set über $\{6666, 5678\}$ wieder zurück zu $\{5678\}$.

All diese Änderungen werden im von PHAS vorgehaltenen Origin Set für das Präfix 123.98.76.0/24 verfolgt. Es kann somit Konflikte feststellen, bei denen mehrere Origin AS für ein Präfix auftauchen. Damit Besitzer von Präfixen bei Vorfällen benachrichtigt werden, können sie sich bei PHAS für ihr Präfix registrieren. PHAS versendet bei Änderungen des Origin Set Mitteilungen per eMail an für das betroffene Präfix registrierte Nutzer. Da bei Hijacking-Vorfällen die Verfügbarkeit des betroffenen Netzes schnell sinkt, wird empfohlen, möglichst viele verschiedene eMail-Adressen bei verschiedenen Anbietern anzugeben. Dies soll sicherstellen, dass zumindest über eine Route noch eMails empfangen werden können. Nachteilhaft an PHAS ist, dass es trotz seiner möglichst globalen Sicht keine MOAS-Konflikte erkennen kann, die in ihrer Verbreitung begrenzt sind. Dies kann durch einen Angreifer gezielt herbeigeführt werden. [17]

4.1.2 iSPY

Statt wie PHAS eine möglichst allumfassende, externe Sicht zu erhalten, verwendet iSPY die interne Perspektive eines AS nach außen. Hijacking-Angriffe führen meist in kürzester Zeit zu Unerreichbarkeit einer signifikanten Anzahl von Zielen. iSPY nutzt diesen bei MOAS-Konflikten auftretenden Effekt des Konnektivitätsverlusts aus, um Vorfälle schnell zu erkennen. Hierfür verwendet iSPY eine Heuristik, die die individuelle Unerreichbarkeitssignatur eines AS überwacht und von Konnektivitätsverlust aus anderen Ursachen unterscheidet. Die Erreichbarkeit des eigenen Netzes wird durch periodisch durchgeführte Tests wie z.B. Pings – ICMP-basierte wie auch andere – ermittelt.

Um die Erreichbarkeit eines AS von innen heraus zu erfassen, führt iSPY für jedes Transit-AS eine Datenstruktur, die den Pfad zum jeweiligen Transit-AS darstellt. Transit-AS sind solche AS, welche für die Weiterleitung von Netzwerkverkehr zuständig sind. Außerdem haben Transit-AS nur einen kleinen Anteil an der Gesamtheit aller AS, weshalb die zu testende Anzahl an AS stark reduziert wird. Als Knotenpunkte des Routings werden Transit-AS jedoch notwendigerweise von BGP-Änderungen betroffen. Daher haben Hijacking-Angriffe auch zwingend auf Transit-AS Auswirkungen.

Der Pfad zu jedem Transit-AS wird via Traceroute ermittelt und als Datenstruktur T festgehalten. Ausgehend von einer Momentaufnahme T_{old} , welche vollständige Konnektivität abbildet, überprüft iSPY für jedes AS, ob der neu ermittelte Pfad zu jedem AS T_{new} teilweise oder vollständig nicht ermittelbar ist. Unermittelbarkeit eines oder mehrerer Hops im Pfad ist ein Indikator für Unerreichbarkeit des durch iSPY überwachten AS. Alle unerreichbaren Hops in allen T_{new} werden in einer Vereinigungsmenge Ω gespeichert. Diese ist die aktuelle Unerreichbarkeitssignatur. Da BGP-Updates immer eine größere Verbreitung erreichen, ist Ω bei Hijacking-Vorgängen notwendigerweise immer groß. Bei anderen Ursachen, die nicht per BGP weiterverbreitet werden, wie z.B. der Ausfall eines Links oder einzelner Router, ist Ω wesentlich kleiner. Die Unerreichbarkeitssignatur ist bei MOAS-Konflikten um mehrere Ordnungen größer als bei anderen Ursachen. Dies wurde auch durch Simulationen verifiziert, sodass durch iSPY eine Kardinalität von 10 für Ω als Schwellwert für Alarm verwendet wird.

Bei Datenbeständen vergangener Vorfälle und Simulationen erreicht iSPY laut seinen Entwicklern eine Fehlerrate erster Art kleiner als 0,45%. Fehler zweiter Art, also fälschliche Erkennung inexisterter Angriffe, treten mit einer Wahrscheinlichkeit von unter 0,17% auf. iSPYs Verlässlichkeit hängt weder von der Sicht einer beschränkten Anzahl an Kontroll-Routern eines einzigen, potentiell ausfallgefährdeten Systems ab, noch ist die Benachrichtigung des Angegriffenen bei Vorfällen beeinträchtigt. Vollständiges Prefix-Hijacking sowie Subprefix-Hijacking werden verlässlich erkannt. Allerdings wird ein korrekt durchgeführter Interception-Angriff nicht notwendigerweise festgestellt. [4]

4.2 Kryptographische Erweiterungen

4.2.1 S-BGP

S-BGP (Secure BGP) ist einer von vielen Vorschlägen, das Border Gateway Protocol kryptographisch abzusichern und Authentizität, Autorisierung und Integrität zu gewährleisten. Ähnlich DNSSEC etabliert es eine Public Key-Infrastruktur entlang der Adressvergebearchie. Die IANA ist die Wurzel der Vertrauenshierarchie und stellt die Zertifikate der regionalen Registries aus, welche wiederum den großen ISPs Zertifikate ausstellen. Diese zertifizieren ihre Kundenetze usw. Die gesamte PKI besteht aus zwei Teil-PKIs. Eine der beiden ist für die Absicherung der Vergabe von Adressbereichen zuständig, die andere für die Vergabe von AS-Nummern. Durch Attestierungen erlaubt eine Partei einer anderen, Adressen zu proklamieren oder Routen zu veröffentlichen. Die Attestierungen werden von der autoritativen Partei signiert und beinhalten die berechnete Partei sowie die Gegenstände der Berechtigung. Auf diese Weise wird sichergestellt, dass ein Router, der einen Adressbereich bewirbt, auch zum Origin AS gehört. Äquivalent gilt dies auch für BGP-Updates, diese müssen ebenfalls attestiert werden.

Da jeder Router die BGP-Nachrichten signiert, findet eine Verschlüsselung mit Signaturen statt, die alle Einträge einzeln verifiziert. Durch diese PKI werden Adressbereiche und AS-Nummern an Organisationen gebunden, und Organisationen an Router [1, 18]. Im trivialen Fall berechtigt sich also ein AS dafür, den ihm zugewiesenen Adressbereich zu proklamieren. Sicherheit auf Netzwerk- und Transportebene wird durch IPsec geleistet, welches von S-BGP vorgeschrieben wird. Es schützt die Vertraulichkeit und Integrität der BGP-Daten während der Übertragung. Die Schlüsseldistribution für IPsec wird durch die PKI von S-BGP übernommen. Die wichtigsten Schwachstellen des BGP, also Validität von Routen-Informationen sowie Berechtigungen für Annoncierung, werden durch S-BGP behoben.

Der Nachteil des Systems liegt in den Hürden der Inbetriebnahme und dem ressourcenintensiven Einsatz. Beim Aufbau einer BGP-Sitzung zwischen zwei Routern muss die gesamte Routing-Tabelle ausgetauscht werden. Für diese gesamte Routing-Tabelle müssen alle Einträge kryptographisch auf Legitimität geprüft werden. Im Jahr 2000 war das Netz, gemessen an der Anzahl Autonomer Systeme, um den Faktor 8 kleiner als zum jetzigen Zeitpunkt. Dabei waren pro Router, mit dem eine Sitzung etabliert wurde, 220.000 Überprüfungen notwendig. Aus diesem Grund wird von den Protokollentwicklern das Nachrüsten von nichtflüchtigem Massenspeicher für jeden Router vorgeschlagen, um verifizierte Daten zu cachen. [18]

4.2.2 *Secure Origin BGP*

Secure Origin BGP (soBGP) [1] basiert ebenfalls auf einer PKI. Diese ist allerdings nicht hierarchisch organisiert, sondern dezentral als Web of Trust. Jedem BGP-Router werden drei Zertifikate übergeben: (1) Ein Zertifikat bindet einen öffentlichen Schlüssel an den zertifikatsinhabenden Router. (2) Ein weiteres Zertifikat beinhaltet die Netzwerkconfiguration und Topologie-Information über Netzwerkneighbarn. (3) Ein drittes Zertifikat dient der Attestierung von Origin AS für ihre Adressbereiche und Erlaubnis für Routen-Propagation. Das Topologie-Zertifikat wird außerdem weiterverbreitet, sodass alle Router die Topologie kennen. Wie S-BGP setzt auch soBGP auf IPsec auf.

soBGP ist weniger vollständig als S-BGP, dafür aber mit insgesamt geringeren Einsatzhürden verbunden. Die größte Gefahr, das Annoncieren von Präfixen durch AS, die nicht das legitime Origin AS sind, wird unterbunden. Dennoch bestehen einige Mängel: Gefälschte Pfad-Informationen, die Topologie-konform sind, können nicht erkannt werden. Sie sind ggfs. ineffizient, aber werden nicht als bösartig erkannt. Außerdem sind die verwendeten Zertifikate nicht konform zu bereits existierenden Standards. Ferner konvergiert die Topologie-Information weltweit nicht schnell, und ist ohne Neuausstellung von Zertifikaten nicht änderbar, bspw. wenn Präfixe anderen Organisationen zugeordnet werden. Dies sorgt für eine geringere Flexibilität. Besonders wegen der Knappheit an IPv4-Adressen scheint es wahrscheinlich, dass in Zukunft mehr Adressen die Organisation wechseln.

4.2.3 *Pretty Secure BGP*

Pretty Secure BGP (psBGP) bildet den Mittelweg zwischen soBGP und S-BGP. Für die Authentisierung von AS wird eine zentralisierte PKI der Tiefe 1 verwendet. Sie bindet AS-Nummern an öffentliche Schlüssel, sodass Angreifer sich nicht für andere AS ausgeben können. Der Unterschied zu

S-BGP besteht darin, dass alle ASN direkt von der IANA zertifiziert werden, um den enormen Aufwand der PKI von S-BGP zu minimieren. Um IP-Adressen an AS zu binden, kommt eine dezentrale PKI als Web of Trust zum Einsatz. Hierbei gibt jedes AS Listen von Präfixen an, die es als zu ebenfalls angegebenen AS-Nummern zugeordnet erklärt. Eine solche Liste legt jedes AS für sich an wie auch für seine Peers, also Nachbar-AS mit beidseitigem kostenlosem Transit. So bestätigen sich mehrere AS gegenseitig, dass sie die korrekten Origin AS für ihre Präfixe sind. Integrität wird ebenfalls durch IPsec sichergestellt. Validität der Pfad-Informationen wird identisch zu S-BGP implementiert, indem eine verschachtelte Signierung aller Einträge einer BGP-Route vorgenommen wird. [19]

4.2.4 *RPKI*

Den Ansätzen S-BGP, soBGP und psBGP ist gemein, dass sie Attestierungen verwenden, um die Berechtigung zur Annoncierung von Adressbereichen und Weiterverbreitung von Routen zu gewähren. RPKI, die Resource Public Key-Infrastruktur, verwendet hingegen Zertifikate. Hierbei handelt es sich um X.509-Zertifikate, die auch in vielen anderen Gebieten Anwendung finden. Im Fall von RPKI beinhalten diese Zertifikate ein Feld, in welchem Netzwerkressourcen wie IP-Adressen und AS-Nummern angegeben sind. Identisch zu S-BGP verläuft die Hierarchie der RPKI entlang der Adressvergabekette. In dieser Kette stellen hierarchisch höher angesiedelte Institutionen die Certificate Authorities der ihnen untergeordneten Institutionen dar und stellen ihnen Zertifikate aus. Eine weitere Form von Objekten, sogenannte Route Origin Authorizations, werden von autoritativen Organisationen signiert. Sie bestätigen dem Besitzer der ROA, dass er berechtigt ist, bestimmte Präfixe zu annoncieren. Ein solches ROA beinhaltet ein oder mehrere Präfixe und genau eine AS-Nummer, die berechtigt ist, diese Präfixe via BGP zu bewerben. Die ROAs werden in der RPKI veröffentlicht. Ebenfalls enthalten sie ein Maximum Prefix Length-Feld, welches die maximale Länge eines annoncierbaren Präfix angibt. Es verhindert Subprefix-Hijacking durch Ausnutzen der Longest Prefix Match-Regel. [20]

Auf diese Weise wird sichergestellt, dass kein AS Präfixe für sich beansprucht, die ihm nicht gehören, und diese Informationen in die globale Routing-Tabelle gibt. Dies verhindert die gefährlichsten Angriffe auf das BGP und garantiert die Validität von Origin AS. Allerdings wird nicht die Validität der gesamten Route sichergestellt. So kann ein Angreifer zwar nicht sein AS als das Ziel-AS für ein bestimmtes Präfix ausgeben, aber immerhin Verkehr über sich leiten. RPKI bietet hiergegen keine Sicherheit. Allerdings ist die RPKI, auch wenn sie eigenständig operieren kann, nur ein Teil von BGPsec. Bei BGPsec handelt es sich um eine Überarbeitung des BGP, welche alle in Sektion 3 genannten Probleme adressiert. BGPsec wird von der IETF-Arbeitsgruppe für Secure Inter-Domain Routing (SIDR) entwickelt. Auch schreibt BGPsec gesicherten Transport über Protokolle wie TLS oder TCP/MD5 vor, um den Problemen von TCP zu begegnen. Es kann auch über SSH getunnelt werden [21]. RPKI ist das erste für den Einsatz fertige Ergebnis der SIDR-Arbeitsgruppe. Auf lange Sicht soll BGPsec möglichst weitgehend umgesetzt werden, um so die bekannten Angriffsvektoren des BGP zu schließen. [20]

Der größte Vorteil von BGPsec bzw. RPKI liegt dort, wo bisherige Vorschläge scheitern. Der RPKI-Standard sieht Ca-

ches vor, welche Zertifikate und ROAs vorhalten und die kryptographische Validierung vornehmen [21]. Somit werden Speicher- und rechenintensive Operationen, welche signifikanten Overhead produzieren und eine Erweiterung der Hardware von BGP-Routern fordern würden, an dedizierte Instanzen ausgelagert. Auch wird redundanter Verkehr und Datenhaltung reduziert, da ein Cache Dienste für mehrere BGP-Router erbringen kann. Dies macht den Einsatz von RPKI weitaus einfacher und weniger ressourcenintensiv als bisherige Ansätze zur kryptographischen Absicherung des BGP. RIPE hat sich für eine Weiterentwicklung und Förderung von RPKI ausgesprochen [22]. Die American Registry for Internet Numbers (ARIN) hat in Koordination mit anderen Registries den Testbetrieb der RPKI begonnen [23]. Der aktuelle Entwicklungsstand von RPKI ist in den RFCs mit Draft-Status 6480 bis 6493 festgehalten [24].

5. VERWANDTE ARBEITEN

Die vorliegende Arbeit gibt einen Einblick in die technischen Hintergründe von BGP Hijacking und Interception. Hierfür werden die Schwachstellen des BGP aufgezeigt. Die praktische Vorgehensweise bei BGP-basierten Angriffen wird durch Vorfälle aus der Praxis beleuchtet. Reaktive Erkennungssysteme zur frühen Identifikation von Vorfällen aus verschiedenen Perspektiven wurden ebenso vorgestellt wie präventive Erweiterungen des BGP zur Beseitigung der Ursachen. Als weiterführende Arbeiten seien folgende empfohlen:

- [1] ist eine umfassende Sicherheitsanalyse des BGP. Es beleuchtet detailliert sowohl die Schwachstellen des BGP wie auch die in dieser Arbeit behandelten Lösungsansätze S-BGP, psBGP und soBGP. Da RPKI zum Zeitpunkt der Veröffentlichung der Arbeit noch nicht weit entwickelt war, erhält es keine eingehende Behandlung. Diese Arbeit ist sehr umfassend und erörtert nahezu alle relevanten Aspekte der BGP-Sicherheit.
- Bei [25] handelt es sich um eine Bachelorarbeit zur Entwicklung einer RPKI-Validierungs-Bibliothek. Insbesondere geht sie detailliert auf BGPsec und RPKI ein.
- [2] setzt einen starken Fokus auf BGP-Angriffe. Die Theorie von Angriffen auf die globale Routing-Infrastruktur wird sehr ausführlich erläutert. Auch wird ein Überblick über eine Vielzahl vergangener BGP-Vorfälle gegeben.
- In [26], einer Präsentation für die North American Network Operators Group, wird der in Abschnitt 3.4 vorgestellte Vorfall ausführlich beleuchtet. Vor allem findet eine graphische Analyse der Propagierung der falschen Routen und eine zeitliche Aufbereitung des Verlaufs des Vorfalls statt. Diese Aufbereitung vermittelt einen guten Eindruck des Ablaufs von Hijacking-Angriffen.

6. ZUSAMMENFASSUNG

Trotz langjähriger Bekanntheit der Schwachstellen im einzigen eingesetzten Protokoll für Inter-Domain Routing, dem BGP, ist dieses noch in fast ursprünglicher Form im Einsatz. Das BGP sieht keine Mechanismen vor, die Angaben über Routen oder Präfix-Ursprünge validieren können. Daher kann jedes Netz beliebige Präfixe für sich beanspruchen.

Vorfälle durch Fehlkonfiguration und -verhalten von BGP-Routern traten in der Vergangenheit mit großem Effekt auf. Große Teile des Internets waren zeitweise, von mehreren Stunden bis hin zu mehr als einem Tag, vollkommen un erreichbar. Hiervon waren große Firmen wie Microsoft oder Amazon ebenso wie viele Millionen bis hin zu Milliarden Nutzer betroffen. Auch gezielte Angriffe können bis dato sehr einfach und erfolgreich durchgeführt werden. Zum Versand von Spam-Mails, zur Zensur oder auch zu vielen anderen Zwecken können Adressbereiche anderer Organisationen übernommen werden, sodass in großen Teilen der Welt Blackholing auftritt. Bei korrekter Umsetzung können sogar Man-in-the-Middle-Angriffe gegen ganze Netze durchgeführt werden.

Mit PHAS und iSPY wurden zwei Systeme zur Erkennung von Hijacking-Vorfällen vorgeschlagen. Sie unterscheiden sich insbesondere in der Perspektive, die zur Erkennung eingenommen wird. Wird ein Vorfall beobachtet, benachrichtigen PHAS und iSPY den Präfix-Besitzer, der manuell Gegenmaßnahmen ergreifen kann. Nachteilig an PHAS ist, dass u.U. die Benachrichtigungen das angegriffene Präfix nicht erreichen. Wegen der Redundanz durch die weitverbreitete Nutzung mobiler Datennetze ist dies heutzutage jedoch nicht mehr von großer Relevanz. Beiden Systemen ist gemein, dass sie nur die Symptome eines unsicher entwickelten Protokolls bekämpfen.

Validität von Routen und Präfix-AS-Zuordnungen kann nur durch kryptographische Absicherung des BGP erreicht werden. Mit S-BGP existiert bereits lange ein Ansatz, der alle Schwachstellen des BGP adressiert. Allerdings ist S-BGP sehr ressourcenintensiv. Insbesondere fordert es Hardware-Erweiterungen mit nichtflüchtigem Speicher an allen BGP-Routern, um PKI-Objekte vorzuhalten. Auch der Overhead durch kryptographische Operationen ist signifikant. Aus diesen Gründen ist S-BGP sowohl in der Inbetriebnahme wie auch im laufenden Einsatz nicht praktikabel und mit sehr großen Hürden verbunden. psBGP und soBGP haben zwar geringere Einsatzhürden, bieten aber keinen vollständigen Schutz gegen die in Sektion 2 genannten Schwachstellen. Aus diesen Gründen konnte sich bis heute keines dieser drei Systeme durchsetzen.

RPKI, als eigenständig einsetzbarer Teil von BGPsec, behebt die Nachteile von S-BGP. Insbesondere wird die Datenhaltung der PKI-Objekte auf externe Caches ausgelagert, die mehrere BGP-Router bedienen können. Gleichzeitig ist der durch BGPsec gebotene Schutz ebenso vollständig wie der von S-BGP und schließt alle bekannten Schwachstellen. RPKI eliminiert im eigenständigen Betrieb immerhin den größten Angriffsvektor, indem es die Validität von Origin AS forciert. Hijacking von fremden Adressbereichen ist somit nicht mehr möglich. RIPE hat seine Unterstützung von RPKI erklärt und ARIN unterhält eine öffentliche RPKI-Testumgebung. Ferner haben alle relevanten Netzerküstungshersteller das RPKI-Protokoll bereits zu großen Teilen implementiert [27].

7. FAZIT

Mit BGPsec und insbesondere RPKI scheint eine Lösung gefunden zu sein, die breite Unterstützung durch Standardisierungskomitees, Registries wie auch durch Hersteller erfährt. Die Einsatzhürden sind wesentlich geringer als bei vorherigen Lösungen, der Schutz jedoch bei Fertigstellung von BGPsec vollständig. Die von RPKI gebotene Sicherheit ge-

gen Fälschung von Präfix-Besitz und somit Hijacking beseitigt das größte Problem des BGP. RPKI wird daher bereits als eigenständige Lösung Einsatz finden.

Komplementär zu RPKI kann, um Manipulation der via BGP verbreiteten Pfade zu erkennen, auf PHAS und iSPY gesetzt werden. Diese können auch parallel betrieben werden. Sie sind mit keinen nennenswerten Einsatzhürden verbunden, weshalb ihre Inanspruchnahme keine Nachteile mit sich bringt. In der Übergangsphase bis zum Einsatz von RPKI sind sie eine gute Möglichkeit, auf Hijacking-Vorfälle reagieren zu können. Aber auch darüber hinaus können sie Vorfälle erkennen, die nicht durch RPKI abgedeckt werden, z.B. solche, die durch Entwendung von Zertifikaten herbeigeführt werden.

8. LITERATUR

- [1] K. Butler, T. R. Farley, P. McDaniel, and J. Rexford, „A Survey of BGP Security Issues and Solutions”, in *Proceedings of the IEEE*, vol. 98, pp. 100–122, Januar 2010.
- [2] K. T. Latt, Y. Ohara, S. Uda, and Y. Shinoda, „Analysis of IP Prefix Hijacking and Traffic Interception”, *International Journal of Computer Science and Network Security*, vol. 10, pp. 22–31, Juli 2010.
- [3] T. Bates, P. Smith, and G. Huston, „CIDR Report”, März 2012. <http://www.cidr-report.org/as2.0/>.
- [4] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, „iSPY: Detecting IP Prefix Hijacking on My Own”, in *ACM SIGCOMM*, 2008.
- [5] S. M. Günther, *Skript: Grundlagen Rechnernetze und Verteilte Systeme*. Lehrstuhl für Netzarchitekturen und Netzdienste, Fakultät für Informatik, Technische Universität München. 2011.
- [6] X. Hu and Z. M. Mao, „Accurate Real-time Identification of IP Prefix Hijacking”, 2006.
- [7] S. Murphy, „BGP Security Analysis”, August 1998.
- [8] L. Benkis, „Practical BGP Security: Architecture, Techniques and Tools”, September 2005.
- [9] T. Underwood, „Internet-wide catastrophe – last year”, 24. Dezember 2005. <http://www.renesys.com/blog/2005/12/internetwide-nearcatastrophela.shtml>.
- [10] A. C. Popescu, B. J. Premore, and T. Underwood, „The Anatomy of a Leak: AS9121”, 15. Mai 2005.
- [11] M. A. Brown, A. Popescu, and E. Zmijewski, „Who’s Manning the L root?”, in *NANOG Meeting 43*, renesys, Juni 2008.
- [12] E. Zmijewski, „Identity Theft Hits the Root Name Servers”, 19. Mai 2008. http://www.renesys.com/blog/2008/05/identity_theft_hits_the_root_n_1.shtml.
- [13] G. Lowe, P. Winters, and M. L. Marcus, „The Great DNS Wall of China”, 21. Dezember 2007.
- [14] RIPE Network Coordination Centre, „YouTube Hijacking: A RIPE NCC RIS case study”, März 2008.
- [15] A. Pilosov and T. Kapela, „Stealing The Internet – An Internet-Scale Man In The Middle Attack”, DEFCON 16, 10. August 2008. www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-pilosov-kapela.pdf.
- [16] U.S.-China Economic and Security Review Commission, „Report to Congress”, U.S. Congress, November 2010.
- [17] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, „PHAS: A Prefix Hijack Alert System”, 2006.
- [18] S. Kent, C. Lynn, and K. Seo, „Secure Border Gateway Protocol (S-BGP)”, *IEEE Journal on Selected Areas in Communications*, vol. 18, April 2000.
- [19] T. Wan, E. Kranakis, and P. C. van Oorschot, „Pretty Secure BGP (psBGP)”, 5. November 2004.
- [20] G. Huston and R. Bush, „Securing BGP and SIDR”, *IETF Journal*, vol. 7, Juli 2011. <http://isoc.org/wp/ietfjournal/?p=2438>.
- [21] R. Bush and R. Austein, „The RPKI/Router Protocol (Draft)”, 3. Februar 2012. <https://datatracker.ietf.org/doc/draft-ietf-sidr-rpki-rtr/>.
- [22] M. Neylon, „RIPE Members Vote To Continue RPKI Work”, 3. November 2011. http://www.circleid.com/posts/20111103_ripe_members_vote_to_continue_rpki_work/.
- [23] ARIN, „RPKI Pilot”, 2011. <https://rpki-pilot.arin.net/>.
- [24] SIDR Work Group, „Secure Inter-Domain Routing (sidr) Work Group Documents.” <https://datatracker.ietf.org/wg/sidr/>.
- [25] F. Holler, „Konzeption und Entwicklung einer Client-seitigen RPKI-RTR Library zur Validierung der Präfix-Zugehörigkeit von autonomen Systemen in BGP-Routen”, Bachelorarbeit, Hochschule für Angewandte Wissenschaften Hamburg, 9. November 2011.
- [26] A. C. Popescu, B. J. Premore and T. Underwood (Renesys Corporation), „The Anatomy of a Leak: AS9121”, 15. Mai 2005. <http://www.renesys.com/tech/presentations/pdf/renesys-nanog34.pdf>.
- [27] R. Bush, R. Austein, K. Patel, H. Gredler, and M. Waehlich, „RPKI Router Implementation Report”, 8. Januar 2012.

Traceroute Anomalies

Martin Erich Jobst

Supervisor: Dipl.-Inf. Johann Schlamp

Seminar Future Internet SS2012

Chair for Network Architectures and Services

Department for Computer Science, Technische Universität München

Email: martin.jobst@tum.de

ABSTRACT

Traceroute is – after ping – one of the most widely used network diagnostic tools, due to its simplicity and yet very wide range of applications. Possible applications for traceroute range from simple error diagnosis to large scans, which reveal the underlying network topology. However, since traceroute was not built with modern network technologies in mind, it faces many difficulties. These difficulties usually manifest themselves in strange or false results, so-called *anomalies*. This drastically affects traceroute’s abilities for network diagnosis and analyzation, especially in large-scale networks. The correct use of traceroute and interpretation of its output has therefore become more and more important. Projects trying to map the topology of the Internet are also greatly affected by traceroute anomalies, as they usually have to solely rely on traceroute and similar scans.

This paper gives a systematic overview of the most frequent traceroute anomalies. The main symptoms of each anomaly are examined based on example scenarios and corresponding output. Additionally, the consequences each anomaly has on the diagnosis of network failures, congestion and mapping efforts is analyzed. This also includes typical wrong conclusions drawn from anomalous traceroute results. Finally, several existing and promising future countermeasures against the respective anomalies are presented and analyzed.

Keywords

Traceroute, Anomalies, Load Balancing, Paris Traceroute, Traceroute Extensions

1. INTRODUCTION

Developed by Van Jacobson in 1988 [4], traceroute has become one of the most important tools for diagnosing network problems. It is used to measure the path a packet takes from the local host to a specified destination. Additionally, for each hop on the path, the *round-trip times* or *RTTs* are recorded.

Since large-scale networks, like the Internet, are usually operated by many different administrative entities, complete and up-to-date information about the network topology and state is usually difficult to obtain. In many cases, measurements taken with traceroute are the only way, to obtain such information. The traceroute user base therefore ranges from end-users in small home LANs to operators of large backbone networks. The conclusions taken from traceroute output are often the only way to effectively diagnose net-

work problems, like link failure and congestion issues, and to analyze traffic flow. As even information about the Internet’s global network topology is relatively scarce, some projects have emerged which try to map the topology based on several different scans [8]. These projects also heavily rely on the accuracy of the results taken from traceroute and similar scans to thousands of destinations.

However, the classic traceroute was not built with modern network management technologies in mind. Since it is mostly oblivious to such new developments, it often generates false or strange results, so-called *anomalies*. These anomalies make diagnosing network problems with traceroute much more difficult, if not downright impossible.

Several measurements [1, 2, 10] have in fact shown that, against common belief, traceroute anomalies are actually occurring quite frequently. Hence, traceroute anomalies are a very big obstacle for network administrators, as well as the various efforts to create a complete and accurate map of the Internet. To effectively use traceroute and correctly interpret its output has become more and more of a skill today, as can be seen in additional efforts taken to instruct network operators on these topics, e.g. [9].

In section 2 general background information for this paper is discussed. Section 3 contains an overview of the respective anomalies, with a description of their effects and common causes, as well as example scenarios and corresponding output. The impact on network analysis and diagnosis of each anomaly is also analyzed. In section 4 several existing solutions to mitigate problems related to traceroute anomalies are examined, as well as different existing and future extensions for traceroute. Finally, section 5 summarizes the results of the discussed anomalies and solutions.

2. BACKGROUND

This section presents some background information which is important for the understanding of this paper. It gives a general overview about traceroute, as well as different load balancing principles and routing techniques which affect traceroute.

2.1 Traceroute Basics

The classic traceroute works by sending out ICMP echo requests, so-called *probes*, with a fixed TTL. The TTL value usually starts at one and is incremented on each probe. Each hop then decrements the TTL by one, when forwarding the

packet, and if the TTL reaches zero, it sends back an ICMP TTL exceeded error. This way, traceroute gets an error message from each hop between itself and the destination, containing the IP address of each hop. By subtracting the time when the error is received by the time the probe was sent, traceroute is also able to compute the RTT for each hop. Finally, when the probe reaches the destination, traceroute gets an ICMP echo reply and stops. The basic traceroute message flow is shown in figure 1.

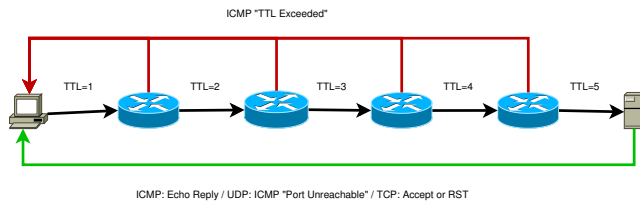


Figure 1: A typical traceroute message flow

Modern traceroute variants also include support for UDP and TCP, as well as IPv6. When using UDP or TCP, the only difference to ICMP is that the packet received from the destination is usually either an ICMP port unreachable or TCP RST packet, respectively. Typical exceptions of this are if the packet is blocked by a firewall or if the port is in use, in which case no error is returned.

2.2 Load Balancing

Load balancing in general is the distribution of packets among several different links or paths. Load balancing mechanisms are usually distinguished in three categories, explained below.

2.2.1 Per-flow Load Balancing

Per-flow load balancing tries to distribute packets according to their so-called *flow*. A flow is usually identified by the 5-tuple of the corresponding packets, i.e. IP addresses, protocol and ports. This is done, so that packets belonging to the same connection are delivered in order to the destination, as best as possible.

2.2.2 Per-packet Load Balancing

Per-packet load balancing distributes each packet individually among the links available. Normally, the packets are distributed randomly or in a round-robin fashion. This has the advantage of requiring less effort inside the router, but on the other hand often introduces huge jitter to connections, especially if the different routes aren't equal in length. Per-packet load balancing usually presents the most problems to traceroute in general, because of its random nature.

2.2.3 Per-destination Load Balancing

Per-destination load balancing distributes packets based on their destination. It is mostly identical to classic routing and normally has little to no impact on the network. Traceroute usually remains completely unaffected by per-destination load balancing.

2.3 MPLS

Multiprotocol Label Switching or *MPLS*, described in RFC 3031 [7], is used to effectively route packets in large-scale

networks, e.g. the Internet. Normally, each router has to make its own routing decisions based on the information contained in the IP header. Since IP addresses are spread quite thin in the Internet, this often requires routers to hold very large routing tables. Additionally, since only few fields in the IP header, i.e. the source and destination address, as well as the TTL, are actually used for routing, it introduces a large unnecessary overhead.

MPLS uses its own header, which encapsulates the original packet. With this, only the first router has to examine the IP header and assigns a *Forwarding Equivalence Class* or *FEC* to the packet in the new header. This designates destinations which are considered equivalent for routing decisions. Since most destinations can actually be grouped together into large blocks, the corresponding tables can be very small. Subsequent routers are then able to base their routing decisions on the much shorter and easier to handle FEC in the MPLS header. Since the TTL values can be copied back and forth between the IP and MPLS header, MPLS routers are also able to honor TTL values set in the original packet. Additionally, RFC 4950 [3] enables the generation and use of ICMP packets in an MPLS context. Hence, MPLS routers may offer basic support for traceroute.

3. TRACEROUTE ANOMALIES

The following is a description of the most frequent traceroute anomalies and their characteristic symptoms. For each anomaly an example message flow is shown, as well as the corresponding output. Additionally, the impact on network diagnosis and analysis is examined.

3.1 Missing Hops

This is the most basic anomaly, where one or more hops are missing from the traceroute output. It usually occurs when a router is protected by a firewall or otherwise configured not to generate ICMP TTL exceeded errors. An example message flow for this anomaly is shown in figure 2, along with the corresponding output. The three asterisks highlighted below, meaning no reply was received for the respective probe, are the key signs for this anomaly.

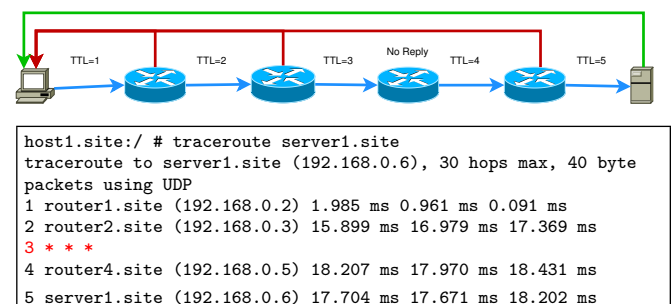


Figure 2: Missing hops example

This anomaly is very easy to notice and of little impact in real life. However, when the network problem is situated exactly on the hop which is not responding, it may actually be quite annoying.

Another reason for missing hops in the traceroute output are MPLS routers which don't honor the TTL value set in the IP

header. Thus, one or more MPLS hops are simply missing in the resulting output. In figure 3 an example scenario for missing MPLS hops is shown. The line where the MPLS routers should appear is highlighted in the output.

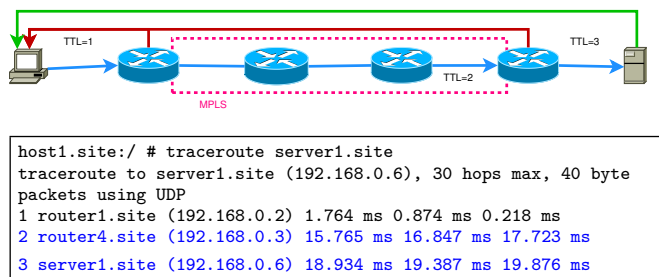


Figure 3: Missing MPLS hops example

This anomaly is very hard, if not impossible, to notice and sometimes very annoying, especially if a MPLS related problem is to be diagnosed.

3.2 Missing Destination

Another, also quite trivial, anomaly is when the destination is missing from the traceroute result. In this case traceroute simply continues with the scan, until it reaches the maximum probe TTL value or if it is interrupted by another constraint. An example would be to stop after a certain number of unsuccessful tries. A special side effect of this anomaly is, that there may be an arbitrary number of hops missing at the end of the output. The usual case for a missing destination is a destination which is protected by a firewall. Figure 4 shows an example of this anomaly. The output again contains the typical three asterisks for the unsuccessful probes and then continues on until the maximum TTL is reached.

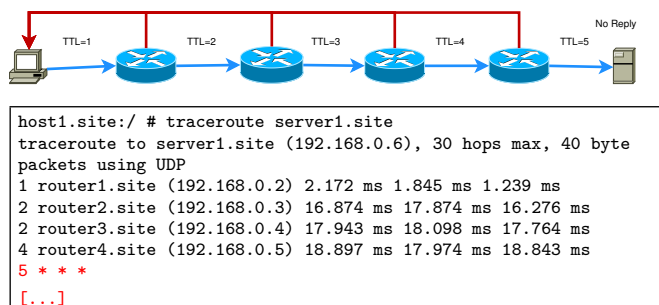


Figure 4: Missing destination example

Again, this anomaly is easily noticeable and merely annoying, for the most part. As it causes scans to take unnecessary long, this anomaly may become a problem, though, especially in cases where the complete topology of a network is to be scanned.

3.3 False Round-Trip Times

This is the case, when the round-trip times reported by traceroute are false. There are usually two reasons for this, either asymmetric packet paths or MPLS routing.

When the respective paths to and from the destination are asymmetric, i.e. the packets are routed on different paths

to and from the target, the round-trip times may not reflect the actual time it takes for a packet to reach the destination. The resulting round trip subsequently show misleading values. The actual path may in fact be much shorter or longer than the round-trip time indicates, depending on the situation. In figure 5 such a scenario is shown, with the corresponding times highlighted in the output. If the return path would jump from the longer path to the shorter, the RTTs measured by traceroute would even become shorter, i.e. the output would show a negative increase in the TTL for the last link.

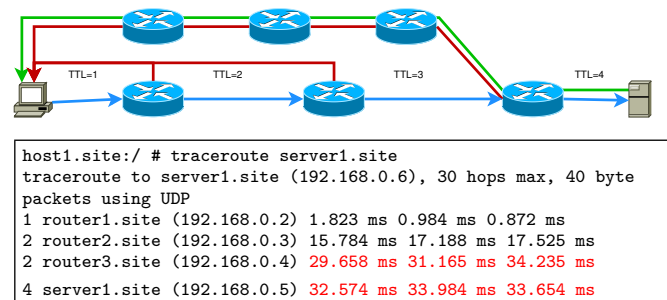


Figure 5: Asymmetric path example

This anomaly is especially problematic, since it may lead to wrong conclusions related to congestion. A sudden and overly large increase in the RTT is usually a very accurate sign for congestion on that link or hop. In this case, however, it may simply be a result of returning packets taking a different route. As this is not visible in the output, it may lead to the wrong conclusion that a link or hop is congested.

A similar result occurs on MPLS links, where the response packet has to travel to the end of the MPLS path, until it is returned to the sender of the probe. Since pure MPLS routers only know about the next hop of a packet, they can't send ICMP errors back right away. Instead, they have to use the path where the original packet would have gone. The result of this is, that all packets are travelling to the last MPLS hop first. Therefore the round-trip times shown in traceroute for the hops in the MPLS path all reflect roughly the round-trip time for the last MPLS router. An example for this anomaly is shown in figure 6. The characteristic signs for this anomaly are the almost equivalent round-trip times for multiple hops in the traceroute output, highlighted below.

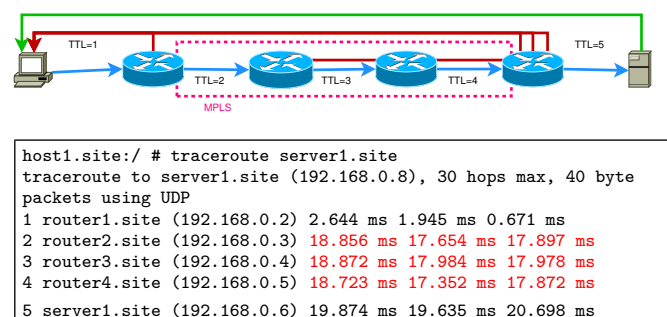


Figure 6: MPLS path example

This can also lead to the wrong conclusion, that a link or hop is congested, for the same reasons as above. In case there actually is congestion on the MPLS-routed path, this anomaly additionally obscures the link or hop which is congested. Since the RTTs reflect the time it takes to reach the last MPLS router, it may be any router in the MPLS path that is congested. However, the output would suggest, that it is the first router or link where the congestion issue is located, if any.

3.4 Missing Links

This anomaly means that the traceroute output is missing links, which are present in the actual topology. The usual reason for this is load balancing, in this case, when all packets are routed on a single path. Figure 7 shows an example of this anomaly. The other link should appear at the two highlighted lines in the output.

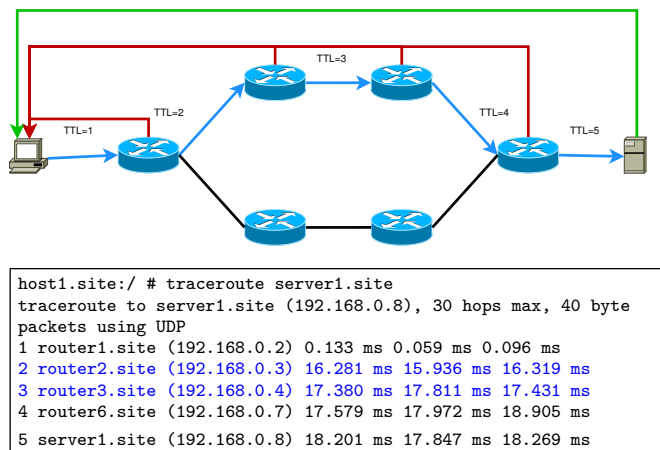


Figure 7: Missing links example

This anomaly is quite problematic, in the sense that it obscures the actual topology. This is a concern, if the network topology itself is to be scanned, as well as if an error is to be diagnosed on the missing link. In the latter case, an error wouldn't show up on the traceroute output, even when it may actually have a great impact on the network.

3.5 False Links

In this case, traceroute implies a false link between hops. It usually occurs in load-balanced links, when some packets are routed via one path and some are routed on another path. An example of this can be seen in figure 8. The false link shows up at the two lines highlighted in the output.

This anomaly is actually a huge problem modern networks, especially since it is not obvious to users without knowledge of the actual topology. Hence, it may lead people to wrong conclusions about the network or a problem to be solved.

3.6 Loops and Circles

This is one of the more complex anomalies, where some hops are missing and other hops are shown multiple times, i.e. the packets seem to travel in loops or circles. The most common case for loops is when load balancing is used for paths of unequal length. Another example may be MPLS links, if

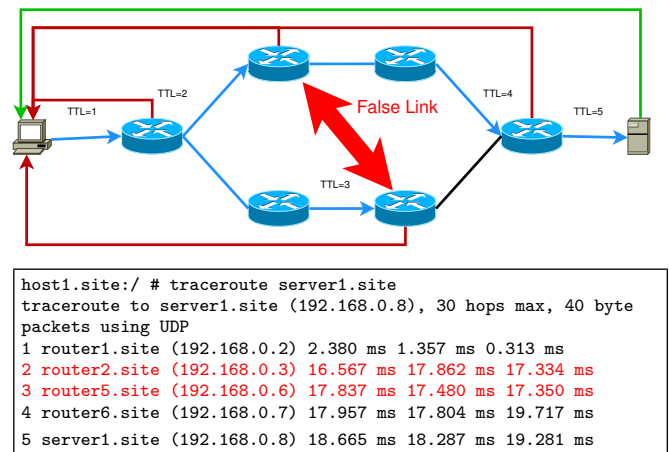


Figure 8: False links example

the address of the last MPLS router is used for ICMP errors, e.g. when intermediate routers lack an IP address. A rarer example is, when packets with a TTL of zero are forwarded to the next hop, e.g. by a faulty router. Cycles usually occur only on load-balanced links, where the difference in length is greater than one. An example message flow is depicted in figure 9. The two lines highlighted below show the hop which is probed twice. However, the corresponding output may also be justified, in case there is an actual forwarding loop or cycle.

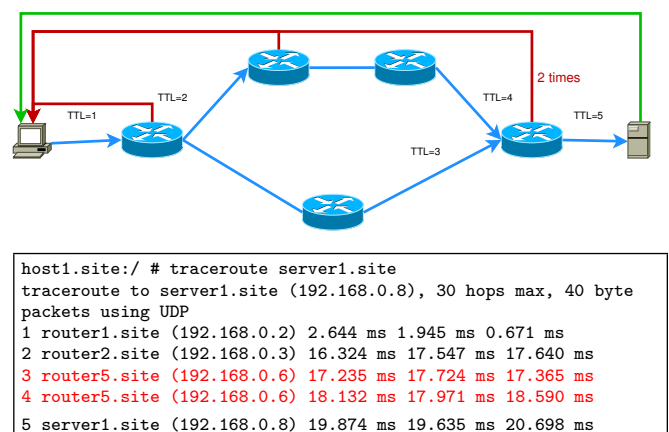


Figure 9: Loops example

This anomaly is normally quite obvious, but still a serious problem. Since some links are missing, the actual topology is yet again obscured. To an unsuspecting user, it may even seem, that packets are actually moving in loops or circles. This is especially the case, if the missing destination anomaly above occurs in conjunction with this anomaly. In that case, it may seem like a valid network problem, when it is only an unfortunate combination of different anomalies.

3.7 Diamonds

Diamonds belong to the most complex anomalies, where some additional links are shown, while others are missing. This anomaly only occurs, when sending out multiple probes for one hop. It is usually caused by load balancing, when

some probes are forwarded on one path and some on other paths. This leads to a complete chaos in the traceroute output, as seen in figure 10. In this case, instead of the textual output, the resulting links which would be inferred by traceroute are shown in figure 11, for brevity.

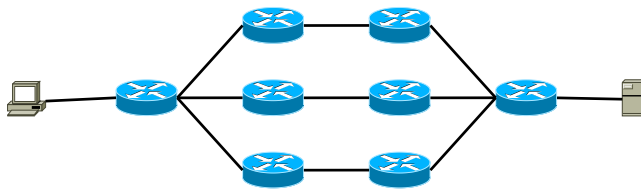


Figure 10: Diamond example

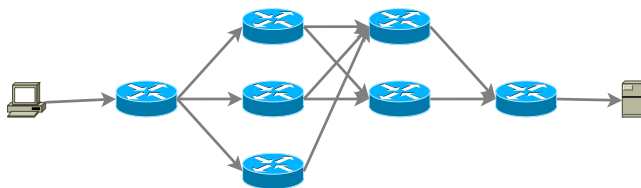


Figure 11: Diamond results

This is yet another case, when load-balanced links cause the traceroute output to be completely useless, even if the correct topology may be known. It can be seen as a combination of the anomalies regarding missing and false links, as well as loops and circles.

4. SOLUTIONS

The following are several solutions for the various anomalies presented above. These solutions can, of course, only limit the impact of said anomalies most of the time. A summary to the anomalies and their respective possible solutions is shown in table 1 further below.

4.1 Paris Traceroute

Paris traceroute was developed to correct most of the deficiencies found in classic traceroute, especially in regard to load-balanced networks. The distinguishing feature of Paris traceroute is, that it tries to actively influence routing decisions in per-flow load-balanced links. It does this by carefully setting header fields in the sent probe packets, which are taken into account by per-flow load balancing [1]. The respective header fields are depicted in tables 2, 3, 4 and 5. The fields used for per-flow load balancing and thus set by Paris traceroute are underlined. The fields used by traceroute to match replies to sent probes are double underlined. A special case is the identifier field in the UDP header, which is specifically modified to produce the desired checksum.

| | | | | |
|----------------------------|-----------------|-----|-----------------|-----------------|
| Version | IHL | TOS | Total Length | |
| <u>Identification</u> | | | Flags | Fragment Offset |
| TTL | <u>Protocol</u> | | Header Checksum | |
| Source Address | | | | |
| <u>Destination Address</u> | | | | |
| Options and Padding | | | | |

Table 2: IP Header fields used by Paris traceroute

| | |
|--------------------|-------------------------|
| <u>Source Port</u> | <u>Destination Port</u> |
| <u>Length</u> | <u>Checksum</u> |

Table 3: UDP Header fields used by Paris traceroute

| | | |
|-------------------|------------------------|-----------------|
| Type | Code | <u>Checksum</u> |
| <u>Identifier</u> | <u>Sequence Number</u> | |

Table 4: ICMP Header fields used by Paris traceroute

| | |
|------------------------|-------------------------|
| <u>Source Port</u> | <u>Destination Port</u> |
| <u>Sequence Number</u> | |
| ... | |

Table 5: TCP Header fields used by Paris traceroute

By keeping the necessary fields constant, Paris traceroute is able to scan a single path. To scan all paths, the fields are intentionally varied and several scans are conducted to, hopefully, traverse all possible links. Thus, it is able to accurately scan single paths, as well as all load-balanced paths to a destination in case of per-flow load balancing. Per-packet load balancing may only be detected by current traceroute versions, due to the randomness of the packet's distribution. Future versions are supposed to include statistical algorithms to accurately distinguish per-packet load-balanced links, too [10]. Paris traceroute additionally includes support for limited control over the return path by influencing the flow information of returned ICMP error packets [2].

4.2 Traceroute Extensions

The following is a list of important traceroute extensions related to the anomalies examined above. Most of them can be found in all modern traceroute variants by now.

4.2.1 UDP and TCP probes

Modern variants of traceroute also support sending of UDP or TCP probes, instead of ICMP echo requests, as described before. Since most routers and firewalls block ICMP echo requests, most modern traceroute implementations in fact use UDP by default. Another advantage of UDP probes is, that they don't require root privileges for sending probes on Linux systems. TCP probes are normally only used in very special cases, usually either to circumvent very restrictive firewalls or to traverse NAT gateways. The main reason against TCP is that it tries to create a connection which subsequently introduces state into the network. Additionally, an application listening on TCP is more likely than for UDP. In fact, to more easily traverse firewalls, most implementations use TCP port 80 as default. To clear up pending connections an additional TCP RST packet is then required. All in all, by using either UDP or TCP instead of ICMP echo requests, missing hops or missing destination anomalies may be somewhat mitigated.

4.2.2 AS-number lookup

This feature makes it possible to automatically query AS-numbers from databases, e.g. the RIPE database, for IP

| Anomaly | Solutions | Comments |
|------------------------|---|---|
| Missing Hops | none | usually impossible to solve from the user's end |
| Missing Destination | UDP/TCP probes | some hosts also block UDP/TCP probes |
| False Round-Trip Times | (Reverse traceroute), MPLS Label-decoding (if caused by MPLS) | helps for a more accurate interpretation of the results |
| Missing Links | Paris Traceroute | only partially helps for per-packet load balancing |
| False Links | Paris Traceroute | only partially helps for per-packet load balancing |
| Loops and Cycles | Paris Traceroute, MPLS Label-decoding (if caused by MPLS) | only partially helps for per-packet load balancing |
| Diamonds | Paris Traceroute | only partially helps for per-packet load balancing |

Table 1: Summary of solutions to the respective traceroute anomalies

addresses encountered by traceroute. It is especially useful to identify network operators, as well as to detect network boundaries. This information may subsequently be used to contact administrators, in case of network failure. There is also a modern algorithm, which combines BGP information with information from several databases to produce even more accurate results [6].

4.2.3 Path-MTU discovery

Path-MTU discovery in traceroute enables users to identify the MTU until each hop. This can ease the identification of “MTU-bottlenecks”, i.e. links where the MTU suddenly drops. It may also help to identify the ideal default MTU to set for outgoing packets, in case automatic detection yields unsatisfactory results.

4.2.4 MPLS-label decoding

This is used to decode MPLS labels, i.e. FECs, returned in extended ICMP error packets, as defined in RFC 4950 [3]. It makes diagnosing MPLS related problems much easier and additionally allows for a more accurate interpretation of the traceroute output. This is especially useful if MPLS-related anomalies, like the one causing false round-trip times or loops resulting from MPLS routing, are suspected.

4.2.5 Reverse traceroute

Reverse traceroute techniques are used to track the path a packet takes from a remote source to the local host. By inspecting the packet return path, additional network problems may be diagnosed and the interpretation of existing output may be eased. This is especially of interest concerning the anomalies related to asymmetric paths.

There is a proposal which would actually achieve this without interaction from the target system [5]. However, it requires the use of the *record-route* or *RR* IP option, which records the hops traversed by the packet. This is done to record the return path of each packet. The RR option was invented to be an alternative to traceroute, but since it requires interaction by the respective routers, it is often not supported and was abandoned. It is also only able to record 8 hops in both directions, which is why the proposal requires multiple hosts, so-called “vantage points”, between the target and the local host. The proposal is therefore not feasible for users without the necessary resources. Finally, it is necessary to spoof the source IP address in sent probes to

redirect the responses to said hosts, which is prevented by most routers.

5. CONCLUSION

Some of the described anomalies have little to no impact on network analysis and diagnosis, while others pose a huge problem. Paris traceroute solves or at least limits the impact of traceroute anomalies in load balancing contexts. Several other traceroute extensions also contribute to counteracting several problems for traceroute found in modern networks. There is also quite some research on this topic, to further improve the situation. Especially reverse traceroute is a promising candidate to solve at least some of the remaining anomalies.

6. REFERENCES

- [1] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding traceroute anomalies with Paris traceroute. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pages 153–158, New York, NY, USA, 2006. ACM.
- [2] B. Augustin, T. Friedman, and R. Teixeira. Measuring load-balanced paths in the internet. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 149–160, New York, NY, USA, 2007. ACM.
- [3] R. Bonica, D. Gan, D. Tappan, and C. Pignataro. ICMP Extensions for Multiprotocol Label Switching. RFC 4950 (Proposed Standard), August 2007.
- [4] V. Jacobson. Original traceroute announcement, 1988.
- [5] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. Van Wesep, T. Anderson, and A. Krishnamurthy. Reverse traceroute. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, NSDI'10, pages 15–15, Berkeley, CA, USA, 2010. USENIX Association.
- [6] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz. Towards an accurate AS-level traceroute tool. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pages 365–378, New York, NY, USA, 2003. ACM.
- [7] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard), January 2001.
- [8] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP topologies with rocketfuel. *IEEE/ACM Trans. Netw.*, 12(1):2–16, Feb. 2004.
- [9] R. Steenberg. A Practical Guide to (Correctly) Troubleshooting with Traceroute. NANOG 47, 2009.
- [10] F. Viger, B. Augustin, X. Cuvellier, C. Magnien, M. Latapy, T. Friedman, and R. Teixeira. Detection, understanding, and prevention of traceroute measurement artifacts. *Comput. Netw.*, 52(5):998–1018, Apr. 2008.



Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nd/3.0/>

Beyond WEP - WLAN Security Revisited

Rolf Sotzek

Betreuer: Holger Kinkel

Seminar Future Internet SS2012

Lehrstuhl Netzarchitekturen und Netzdienste

Fakultät für Informatik, Technische Universität München

Email: sotzekr@in.tum.de

KURZFASSUNG

Computer werden in der heutigen Zeit immer wichtiger und sind aus unserem Alltag nicht mehr wegzudenken, seien es Smartphones oder Laptops. Die Vernetzung von Computern zu Netzwerken war eine bahnbrechende Entwicklung, die zum Erfolg der Computer beigetragen hat. Neue Erkenntnisse in der Funk-Technologie ermöglichten nicht nur kabelgebundene, sondern auch drahtlose Netzwerke (WLANs). Jedoch mussten auch neue Sicherheitsmechanismen für WLANs entworfen werden. Denn bei diesen sind Angreifer nicht mehr auf den physikalischen Zugriff auf das Kabel angewiesen. Wired Equivalent Privacy (WEP) war ein erster Standard, bei dem sehr schnell gravierende Sicherheitsmängel gefunden wurden. Wi-Fi Protected Access (WPA) ist eine überarbeitete Version von WEP. Es besitzt zwar weniger Mängel als WEP, jedoch wird noch immer ein nicht sicheres Kryptographie-Verfahren verwendet. Wi-Fi Protected Access 2 (WPA 2) ist die Weiterentwicklung von WPA und gilt aus heutiger Sicht bei einer richtigen Konfiguration als sicher. Aber in letzter Zeit werden ständig neue Schwachstellen bei WLANs gefunden, mit denen die Sicherheitsmechanismen von WPA 2 umgangen werden können.

Schlüsselworte

WPA, WPA 2, WPS, WPS Sicherheit, Hole 196

1. EINLEITUNG

Das 1997 vorgestellte WEP sollte in WLANs, die nach IEEE 802.11 Standard spezifiziert sind, die Geheimhaltung der Daten sicherstellen. Mögliche Dritte, die versuchen, die Geheimhaltung zu umgehen, werden als Angreifer bezeichnet. Jedoch wurden schon 2 Jahre nach der Entwicklung Sicherheitslücken im verwendeten Kryptographie-Verfahren, dem RC4 Algorithmus, entdeckt. Die vier Ziele der WEP Sicherheitsmechanismen sind die Vertraulichkeit, die Zugriffskontrolle, die Nachrichtenauthentisierung und -integrität. Die Vertraulichkeit ist gebrochen, sobald Angreifer Nachrichten mitlesen können. Der Cyclic Redundancy Check (CRC) ist für die Überprüfung der Nachrichtenintegrität ungeeignet. Durch Einschleusen neuer Nachrichten kann die Zugriffskontrolle umgangen werden. WEP sollte unter keinen Umständen mehr eingesetzt werden, da keines der Sicherheitsziele erreicht werden konnte. S. Fluhrer, I. Mantin und A. Shamir stellten 2001 einen Angriff auf den RC4 Algorithmus vor [1]. Mittlerweile kann WEP in weniger als 60 Sekunden geknackt werden [2].

Bei der Entwicklung von WPA im Jahr 1999 wurde spezi-

ell darauf geachtet, dass die Angriffe, die bei WEP möglich waren, nicht auf WPA anwendbar sind. Dafür wurde das Temporal Key Integrity Protocol (TKIP) entwickelt. Eine zentrale Neuerung war, dass jedes versendete Paket mit einem neuen Schlüssel chiffriert wird. Des Weiteren wurde der CRC Algorithmus durch einen verschlüsselten Message Integrity Check (MIC) (auch oft Michael genannt) abgelöst. Leider verwendet TKIP aber immer noch den unsicheren RC4-Algorithmus.

Im folgendem Kapitel 2 wird die Funktionsweise von WPA 2 erklärt. Dies beinhaltet die Betriebsarten von WPA 2, das Schlüsselmanagement und die Erläuterung der Sicherheitsprotokolle. Eine Auswahl von möglichen Angriffen auf WPA 2 wird in Abschnitt 3 vorgestellt. Hierbei wird zuerst ein Angriff auf den Advanced Encryption Standard (AES) erläutert. Des Weiteren werden Angriffe, die auf der Unbedarftheit des Nutzer beruhen, beschrieben. Es werden zudem protokollbasierte Angriffe dargestellt. Auf die verwandten Arbeiten wird in Paragraph 4 hingewiesen. Zum Abschluss der Arbeit wird in Kapitel 5 ein Resümee gezogen.

2. WPA 2

Der Begriff „WPA 2“ bezeichnet die Implementierung des IEEE 802.11i Standards. Durch den IEEE 802.11i Standard [11] gab es gravierende Neuerungen, wie zum Beispiel die Trennung von Authentifizierung der Clients und der Geheimhaltung und Integrität der Daten. Es musste also eine robuste und skalierbare Netzwerkarchitektur entworfen werden. Diese neue Architektur für WLANs wird unter dem Oberbegriff Robust Security Network (RSN) zusammengefasst. RSN benutzt den IEEE 802.1X Standard für die Authentifizierung, eine sichere Verteilung der Schlüssel und neue Integritäts- und Verschlüsselungsverfahren. Kommt bei der Authentifizierung oder dem Verbindungsaufbau ein Four-Way Handshake zum Einsatz, so spricht man auch von einer Robust Security Network Association (RSNA).

Der Aufbau einer sicheren Verbindung nach IEEE 802.11i [11] erfolgt in vier Schritten:

- Aushandlung der verwendeten Sicherheitsmechanismen
- Authentifizierung
- Ableitung und Verteilung der Schlüssel
- Verschlüsselung und Integritätsschutz der Datenkommunikation

2.1 Aushandlung der verwendeten Sicherheitsmechanismen

Damit ein Client sich an einem Netzwerk anmelden kann, muss er erst einmal wissen, welche Netzwerke sich in seiner Umgebung befinden. Dies wird erreicht, indem der Client auf jedem Kanal einen Probe Request sendet. Alle Access Points in Reichweite antworten dem Client mit einer Probe Response, die ein RSN Information Element (IE) enthält. Durch das RSN IE wird dem Client mitgeteilt, welches Authentifizierungsverfahren und Sicherheitsprotokoll der Access Point benutzt und ob eine Wiederaufnahme der vorherigen Verbindung möglich ist. [16]

2.2 Authentifizierung

Bei der Authentifizierung während des RSNA-Handshakes gibt es verschiedene Arten. Eine erste Unterscheidung bieten die zwei verschiedenen Betriebsarten von WPA und WPA 2:

WPA-Personal wird auch WPA-PSK genannt und wurde für private Haushalte und kleine Firmen entworfen, die keinen Authentifizierungsserver besitzen. Jeder Client authentifiziert sich durch einen geheimen Text, den Pre-Shared Key (PSK). Dieser PSK ist für alle Clients im WLAN gleich.

WPA-Enterprise wurde für Unternehmen entworfen und dabei wird ein Authentifizierungsserver benötigt. Durch die komplexere Struktur wird mehr Sicherheit gewährleistet. Für die Authentifizierung empfiehlt der IEEE 802.1X Standard das Extensible Authentication Protocol (EAP).

2.3 Ableitung und Verteilung der Schlüssel

Die RSN Netzwerkarchitektur besitzt im Gegensatz zu WEP ein Schlüsselmanagement. Bei WEP wird ein einziger Schlüssel für die Verschlüsselung und die Authentifizierung verwendet. Im Gegensatz dazu gibt es bei WPA 2 verschiedene Schlüssel, die in einer Schlüsselhierarchie angeordnet sind. In der Kryptographie muss sich der Benutzer ausweisen, zum Beispiel durch die Kenntnis von einem Geheimnis, woraufhin er einen neuen Schlüssel bekommt, mit dem er auf die vom ihm gewünschten Dienste zugreifen kann. Diese neuen Schlüssel sind meistens nur temporäre Schlüssel, die nach der Nutzung des Dienstes ungültig werden. Das vereinbarte Geheimnis wird ebenfalls durch einen Schlüssel repräsentiert. Dieser wird aufgrund seiner Wichtigkeit auch Master Key genannt. [3]

Zum einen soll jeder Client des Netzwerkes mit dem Access Point in einer einzelnen, sicheren Verbindung kommunizieren. Zum anderen soll der Access Point aber auch Daten an alle Clients gleichzeitig schicken können. Aus diesen beiden unterschiedlichen Anwendungsfällen ergeben sich die zwei Schlüsselarten des Schlüsselmanagements. Für jede Adressierungsart werden somit auch verschiedene Schlüssel gebraucht. Bei der Kommunikation von einem Access Point mit einem Client kommt ein paarweiser Schlüssel zum Einsatz. Dieser paarweise Schlüssel ist ausschließlich dem Access Point und dem Client bekannt. Zur Kommunikation vom Access Point zu allen Clients wird ein Gruppenschlüssel festgelegt, der allen Clients und dem Access Point bekannt ist.

Dieser Gruppenschlüssel schützt somit Broadcast und Multicast Daten. [3]

2.3.1 Paarweise Schlüssel

In der Schlüsselhierarchie steht beim paarweisen Schlüssel der 256 Bit lange Pairwise Master Key (PMK) an der Spitze. Der gleiche PMK muss sowohl dem Access Point als auch dem Client bekannt sein. Hierbei spielt das verwendete Authentifizierungsverfahren von WPA 2 eine wichtige Rolle. Bei WPA-Personal wird aus dem PSK der PMK berechnet. Bei WPA-Enterprise wird der IEEE 802.1X Standard verwendet, um auf beiden Seiten den PMK zu erzeugen. Jeder Client und der Access Point besitzen ihren eigenen PMK, da bei der Berechnung des PMK Zufallszahlen benutzt werden. In der Hierarchie unter dem PMK ist der Pairwise Transient Key (PTK). Der PTK wird mittels eines Pseudo-Random Number Generators (PRNG) aus dem PMK erzeugt. Dieser PTK zerfällt wieder in kleinere, temporäre Schlüssel. Diese kleineren Schlüssel haben das Ziel, die Schlüssel- und die Datenübertragung zu schützen. Die Schlüsselhierarchie des paarweisen Schlüssels ist in Abbildung 1 verdeutlicht.

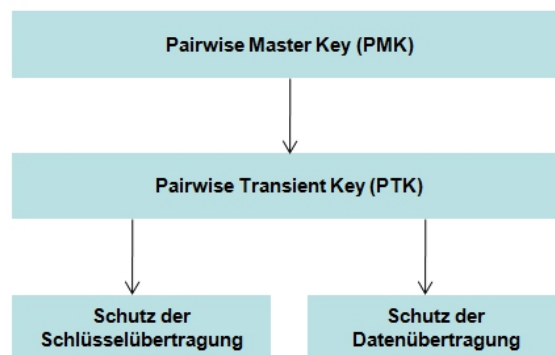


Abbildung 1: Ableitung des PTK

Damit sowohl der Access Point als auch der Client den gleichen PTK erzeugen, wird ein Four-Way Handshake benutzt. Dabei erzeugen beide Kommunikationspartner jeweils eine Pseudozufallszahl und tauschen diese aus. Aus dem PMK, den beiden Pseudozufallszahlen und den MAC-Adressen wird durch dieselbe Pseudo-Random Function (PRF) der gleiche PTK gewonnen. Bei dem gesamten Four-Way Handshake wird das EAP over Lan (EAPoL) Protokoll verwendet. Das EAPoL Protokoll regelt die Verwendung von EAP in Netzwerken und somit auch in WLANs. Nach dem Abschluss des Four-Way Handshakes besitzen beide Kommunikationspartner die nötigen temporären Schlüssel und aktivieren die Verschlüsselung. Die Verwendung des PTK zur Datenverschlüsselung ist in Abbildung 2 veranschaulicht. [3]

2.3.2 Gruppenschlüssel

Genau wie beim PMK ist der Group Master Key (GMK) an der ersten Stelle in der Hierarchie der Gruppenschlüssel. Die Handhabung mit dem GMK ist viel einfacher als bei den paarweisen Schlüsseln, da an der Erstellung nur der Access Point beteiligt ist. Für den GMK wählt der Access Point einfach eine Zufallszahl. In der Hierarchie folgt unter dem GMK der Group Transient Key (GTK). Der GTK

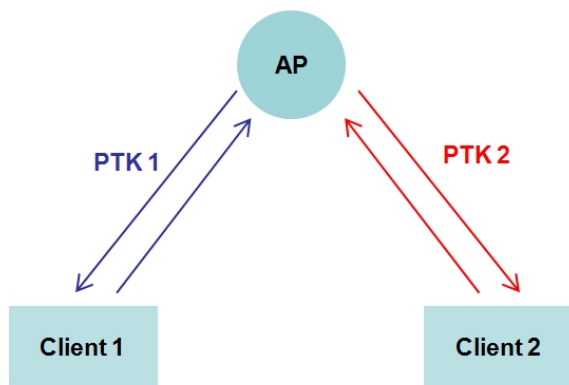


Abbildung 2: Verwendung des PTK

zerfällt wieder in temporäre Schlüssel, mit denen Multicast-Datenübertragungen geschützt werden können. Der GTK wird aus dem GMK, der MAC-Adresse des Access Points und einem Zufallswert unter Verwendung einer PRF erzeugt. Die Hierarchie des Gruppenschlüssels ist in der Abbildung 3 dargestellt.

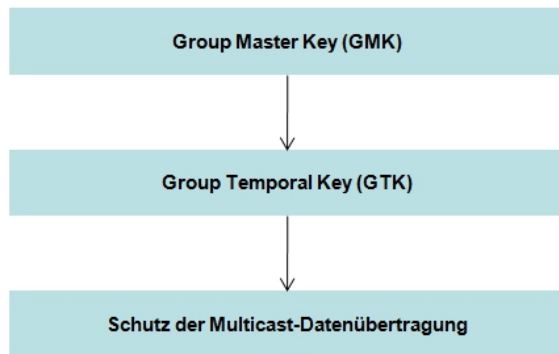


Abbildung 3: Ableitung des PTK

Die Verteilung des GTK an alle Clients ist einfacher, da die bestehende Absicherung durch den PTK benutzt wird. Die Verwendung des GTK zur Datenverschlüsselung von Multicasts ist in Abbildung 4 veranschaulicht. [3]

Bei WPA 2 wurde auch ein Schlüsselwechsel hinzugefügt. Verliert der GTK eines Netzwerks seine Gültigkeit, wird vom Access Point an alle Clients ein neuer GTK verteilt. Der GTK muss zum Beispiel erneuert werden, sobald ein Client das Netzwerk verlässt, damit der Client nicht mehr die Multicast und Broadcast Daten des Access Points empfangen kann. Der Group Key Handshake, der in IEEE 802.11 [11] definiert ist, kümmert sich um die Aktualisierung des GTK. Er ist einem Two-Way Handshake sehr ähnlich:

1. Der Access Point teilt jedem Client des Netzwerks den neuen GTK mit. Der neue GTK ist dabei durch die Schlüsselübertragung des PTK verschlüsselt.
2. Jeder Client bestätigt den GTK und antwortet dem Access Point.

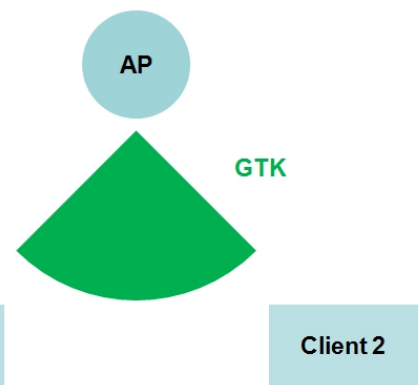


Abbildung 4: Verwendung des GTK

2.4 Verschlüsselung und Integritätsschutz der Datenkommunikation

WPA 2 unterstützt aus Kompatibilitätsgründen sowohl das Temporal Key Integrity Protocol (TKIP) als auch das Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP). Die Unterstützung von CCMP ist bei WPA 2 fest vorgeschrieben. Die beiden Sicherheitsprotokolle für WPA 2 unterscheiden sich stark:

TKIP benutzt den RC4-Algorithmus mit einer Schlüssellänge von 128 Bit. Zur Sicherung der Nachrichtenintegrität wird der Michael Algorithmus verwendet.

CCMP verwendet den Advanced Encryption Standard (AES) mit einer Block- und Schlüssellänge von 128 Bit. Die Nachrichtenintegrität wird durch einen Cipher Block Chaining - Message Authentication Code (CBC-MAC) gesichert.

2.4.1 TKIP

Die zentralen Verbesserungen von WEP zu TKIP sind [3]:

- In die Berechnung des MIC wird ein geheimer Schlüssel, welcher ein Teilschlüssel des PTK ist, miteinbezogen. Der Michael Algorithmus hat den Vorteil, dass die Prüfsumme von der gesamten Nachricht berechnet wird. Diese beinhaltet somit auch die Ziel- und Quelladresse.
- Das bereits beschriebene Schlüsselmanagement wird eingeführt.
- Jedes Paket wird mit einem neuen Schlüssel chiffriert.
- Sollten MIC-Fehler auftreten, werden Gegenmaßnahmen eingeleitet.
- Der von WEP bekannte Initialisierungsvektor (IV) wird durch einen erweiterten IV ersetzt.

2.4.2 CCMP

Bei CCMP wird AES in Counter Modus zur Verschlüsselung von Daten benutzt. Ein weitere Betriebsart von AES ist der Counter with CBC-MAC Modus (CCM Modus). Dieser wird

für die Berechnung des CBC-MAC gebraucht. Der berechnete CBC-MAC, quasi eine verbesserte Prüfsumme, bietet sowohl Authentifizierung als auch Integrität. Für die weitere Arbeit hat das CCMP eine geringe Bedeutung, deswegen sei der Vollständigkeit halber auf J. Edney und W. A. Arbaugh [3] verwiesen.

3. ANGRIFFE AUF WPA 2

Sucht man heute danach, wie man ein WPA 2 Netzwerk knacken kann, wird man meistens zu Aircrack-ng [9] weitergeleitet. Bei WPA 2 kann mit Aircrack-ng nur die WPA-PSK Betriebsart gebrochen werden. Das Knacken des Schlüssels von WPA-Enterprise ist so gut wie unmöglich, weil ein Schlüssel mit der maximalen Länge verwendet wird. Bei WEP können statistische Lösungsverfahren die Rechenzeit verkürzen. Im Gegensatz dazu muss bei WPA 2 der komplette Schlüsselraum ausprobiert werden. Der Angriff kann nicht beschleunigt werden, da die Schlüssel bei WPA 2 dynamisch sind. Die einzige Information, die man über den PSK erhält, ist in dem Four-Way Handshake enthalten. Der Angreifer muss also die Anmeldung eines Clients an dem Access Point mitschneiden. Ungeduldige Angreifer täuschen einem mit dem Netzwerk verbundenen Client einen Verbindungsabbruch vor, woraufhin sich der Client wieder mit dem Access Point verbindet und somit auch ein neuer Four-Way Handshake stattfindet. [10]

In den Unterkapiteln werden verschiedene Angriffe auf WPA 2 erläutert. Zuerst werden Angriffe auf den AES vorgestellt, gefolgt von Angriffen, die aufgrund der Unvorsichtigkeit von Nutzern ermöglicht werden. Als Letztes werden Protokolle oder Auszüge eines Protokolls gezeigt, die die Sicherheit eines WPA 2 WLANs gefährden können.

3.1 Angriffe aus AES

2011 stellten A. Bogdanov, D. Khovratovich und C. Rechberger den ersten Angriff auf den vollen AES-Algorithmus vor. Mit Hilfe von vollständigen bipartiten Graphen wird eine viermal höhere Geschwindigkeit als mit der Brute-Force-Methode erreicht. In der Kryptographie wird eine Verschlüsselung als geknackt bezeichnet, wenn es ein Verfahren gibt, dass schneller als Brute-Force ist. Statt 2^{128} werden $2^{126,1}$ Operationen benötigt, um einen AES-128 Schlüssel zu brechen. Zwar ist der Rechenaufwand immer noch viel zu groß, aber dennoch gelang ein wichtiger theoretischer Durchbruch. [5]

3.2 Unbedarftheit der Nutzer

Ein großes Problem bei der Sicherheit von WLANs ist, dass sich viele Nutzer der Risiken einer falschen Konfiguration des Access Points nicht bewusst sind. Mögliche Anwender wissen meist nur wenig über die verwendeten kryptographischen Verfahren oder haben keine Ahnung, was ein 256 Bit Schlüssel ist. Bei einem 256 Bit Schlüssel beträgt der Schlüsselraum 2^{256} . Allgemein kann man sagen, dass wenn man die Anzahl der Bits des Schlüssels erhöht, man also einen längeren Schlüssel nimmt, auch die Anzahl der möglichen Schlüssel zunimmt. Angreifer brauchen somit länger, um alle möglichen Schlüssel auszuprobieren. Jedoch sollte man im Umgang mit Schlüsseln nicht vergessen, dass die Sicherheit eines WLANs durch die kryptographische Verschlüsselung erreicht wird und nicht durch den Schlüssel.

Bei WPA-Personal kann der PSK mit 64 Hexadezimalzahlen oder über ein Passwort von 8 bis 63 ASCII Zeichen eingegeben werden. Der PBKDF 2 (ein kryptographisches Verfahren zum Ableiten von Schlüsseln) berechnet aus dem PSK, der SSID und der Länge der SSID mit Hilfe von Hashing den PMK. Ist der PSK n ASCII Zeichen lang, so liefert der PBKDF 2 eine sicherheitsrelevante Schlüssellänge von $2,5 * n + 12$ Bit. [11]

Viele private Haushalte verwenden einfache Wörter als PSK für ihre WLANs. Dies stellt ein großes Problem dar, weil diese mittels Wörterbuch-Attacken schnell erraten werden können. Leider sind diese einfachen PSKs meistens auch noch sehr kurz und somit auch gut durch Brute-Force-Angriffe brechbar. Maßgeblich für die Sicherheit eines WLANs ist immer die Wahl eines guten PSK. Der PSK sollte mindestens 13 Buchstaben lang sein, Zahlen und bestenfalls Sonderzeichen enthalten. A. Ku empfiehlt nach seinen Tests [18] eine Mindestpasswortlänge von acht Zeichen mit mindestens einem Sonderzeichen, einer Zahl und einem Groß- und Kleinbuchstaben. Allerdings scheint ein PSK mit 13 Groß- und Kleinbuchstaben auch sicher zu sein.

In der heutigen Zeit des Internet-Zeitalters gibt es aber auch ein weiteres Problem bei der Sicherheit von Schlüsseln. Weil die Produktion von Computern immer billiger wird, kaufen große Firmen ganze Rechenzentren, in denen sie tausende von Prozessoren zu einem Super-Computer vernetzen. Meistens brauchen die Firmen nicht die ganze Kapazität des Super-Computers und verkaufen somit Rechenzeit an Kunden. Die EC2 von Amazon ist ein Beispiel dafür [17]. Auf solchen Super-Computern geht das Knacken eines Schlüssels sehr viel schneller. Das Knacken eines Schlüssels ist somit eher eine Frage des Geldes anstatt der Zeit.

3.2.1 Unbemerkte Windows-Ad-hoc-Netzwerke

2006 entdeckte S. Nomad ein abnormes Verhalten von Windows-Ad-hoc-Netzwerken [4]. Ein Beispielszenario dieses Verhaltens ist:

- Zuhause besitzt Alice einen Access Point mit der SSID „Netgear“. Sie hat ihren Laptop so eingerichtet, dass er sich automatisch mit diesem Netzwerk verbindet.
- Weil Alice viel reist und ihren Laptop zum Arbeiten nutzt, arbeitet sie auch unterwegs. Das kann beispielsweise an Flughäfen sein.
- Zufällig hat der neben Alice sitzende Bob einen Laptop, der ein Ad-hoc-Netzwerk mit dem Namen „Netgear“ hat.
- Der Laptop von Alice sucht nach dem Netzwerk mit der SSID „Netgear“ und verbindet sich mit Bobs Ad-hoc-Netzwerk.
- Beim nächsten Booten von Alices Laptop ohne angeschlossenes Netzkabel und ohne ein Netzwerk mit der SSID „Netgear“ in Reichweite, erstellt der Laptop von Alice ein Ad-hoc-Netzwerk mit dem Namen „Netgear“.

Angreifer können sich somit gezielt mit den betroffenen Computern verbinden, wodurch weitergehende Attacken ermög-

licht werden. Dieses Szenario basiert auf einem Konfigurationsfehler und er kann sich virusartig von PC zu PC verbreiten. Microsoft hat diesen Fehler mittlerweile behoben. Der beste Schutz vor diesem Fehler ist, die WLAN Karte im Laptop abzuschalten, wenn man kein WLAN benötigt. Die meisten Laptops besitzen sogar eine Taste oder Tastenkombination, mit der man die WLAN Karte abschalten kann. Auch sollte man aus diesem Fehler lernen, dass man sein Betriebssystem und alle Treiber immer auf dem aktuellsten Stand halten sollte.

3.2.2 Karma Attacke

Karma [12] greift die automatische Anmeldung an schon bekannten Netzwerken an.

Generell läuft die Suche nach WLANs unter Windows XP SP 2 wie folgt ab [13]:

- Der Client erstellt eine Liste mit allen verfügbaren Netzwerken in Reichweite, indem ein Broadcast Probe Request auf jedem Kanal gesendet wird.
- Die Access Points in der Umgebung antworten mit ihren Probe Response.
- Der Client besitzt eine Preferred Networks List (PNL). Diese ist eine geordnete Liste von Netzwerken, zu denen sich der Client in der Vergangenheit verbunden hat.
- Sollte ein Netzwerk aus der PNL in Reichweite sein, verbindet sich der Client mit diesem.
- Wenn keines der gefundenen Netzwerke in der PNL ist, werden spezifische Probe Requests für jedes Netzwerk aus der PNL gesendet. Dies ermöglicht, sich mit versteckten Netzwerken zu verbinden.
- Falls immer noch keine Verbindung hergestellt werden konnte und es ein Ad-hoc-Netzwerk in der PNL gibt, wird dieses Netzwerk erstellt und der PC wird der erste Client.
- Ist die Option „Verbindung zu unbekannten Netzwerken herstellen“ gewählt, wird in der Reihenfolge, in der die Netzwerke entdeckt wurden, versucht, sich mit jedem erreichbaren Netzwerk zu verbinden. Diese Option ist standardmäßig ausgeschaltet.
- Sollte immer noch keine Verbindung bestehen, wird die SSID der Netzwerkkarte auf eine Zeichenkette aus 32 zufälligen char-Werten festgesetzt und für eine Minute gewartet, bis der Algorithmus von neuem gestartet wird.

Angriffe auf den Algorithmus:

Angreifer können die SSID aller Netzwerke in der PNL herausfinden, indem sie die Probe Requests des Clients mithören. Angreifer können gefälschte Verbindungsabbrüche an den Client senden, wodurch der Client den Algorithmus neu startet. Dies bewirkt, dass der Client die spezifischen Probe Requests für jedes Netzwerk in der PNL sendet. Angreifer können mit dem Wissen der PNL einen gefälschten Access Point mit der SSID eines Eintrages aus der PNL erstellen.

Karma macht sich dieses Verhalten zu Nutze und erstellt diese gefälschten Access Points automatisch. Sucht ein Client ein Netzwerk mit der SSID „FritzBox“, wird ihm durch Karma ein unverschlüsseltes Netzwerk mit dem Namen „Fritz-Box“ angeboten. Entscheidend ist nun, ob der Client ein verschlüsseltes Netzwerk erwartet oder nicht. Ist in der PNL das Netzwerk mit einer Verschlüsselung gespeichert, wird die Verbindung zu dem von Karma erstellten Netzwerk fehlgeschlagen, da dieses unverschlüsselt ist. Sollte das Netzwerk in der PNL nicht verschlüsselt sein, verbindet sich der Client mit dem Karma Netzwerk. Über weitere gefälschte Services, wie zum Beispiel DHCP und HTTP, kann man relativ einfach einen Man-in-the-middle-Angriff ausführen. [12]

Bei Mac Systemen funktioniert die automatische Anmeldung bei Netzwerken anders als unter Windows XP SP 2. Jedoch bleiben fast die gleichen Probleme erhalten. So geben auch Mac System ihre PNL preis. [13]

Nutzer sollten bei der Verwendung von WLANs also immer alle unverschlüsselten Netzwerke aus der PNL löschen. Des Weiteren sollten die Nutzer vorsichtig im Umgang mit WLANs werden und in unsicheren Umgebungen, wie zum Beispiel öffentlichen Plätzen, nicht einfach ein offenes WLAN benutzen.

3.3 Protokollbasierte Angriffe

Entwickler von neuen Protokollen sind nicht perfekt. Dadurch kommt es zu Design- und Implementierungsfehlern. Diese Fehler sind tief im Protokoll verankert und dadurch später schwer zu kompensieren und aufzufinden.

3.3.1 WPS

Wi-Fi Protected Setup (WPS), früher Wi-Fi Simple Config genannt, ist eine Zertifizierung von der Wi-Fi Alliance [7], die beim Hinzufügen von Geräten zu schon bestehenden Netzwerken hilft. Anfang 2007 wurde dieser Standard von der Wi-Fi Alliance entwickelt, mit dem Ziel, dass auch Leuten ohne umfangreiche Computerkenntnisse eine einfache Möglichkeit geboten wird, neue Clients unkompliziert in ein vorhandenes Netzwerk hinzuzufügen. Unerfahrene Anwender sollten vor der Eingabe von langen PSKs verschont bleiben. Auch könnten Anwender von der Anzahl der verschiedenen Kryptographieverfahren, wie WEP, WPA und WPA 2 verwirrt sein. Mittlerweile bieten fast alle großen Hersteller von Access Points (Cisco, Netgear, D-Link, usw.) Geräte mit WPS-Zertifizierung an. Heutzutage gibt es schon mehr als 200 Produkte, die eine WPS-Zertifizierung von der Wi-Fi Alliance besitzen. Die Wi-Fi Alliance vergibt die WPS-Zertifizierung auf Grundlage der Wi-Fi Simple Configuration Specification (WSC). [6]

Der Standard beinhaltet drei verschiedene Methoden, wie Anwender neue Geräte in ein Netzwerk integrieren können [8]:

Push Button Configuration. Bei diesem Verfahren können neue Geräte durch das Drücken von Knöpfen angemeldet werden. Der Access Point besitzt einen physikalischen Knopf. Die anzumeldende Gegenstelle kann entweder einen physikalischen oder einen Software implementierten Knopf aufweisen. Um neue Geräte am Access Point anzumelden,

muss zuerst der Knopf am Access Point gedrückt werden. Hiernach beginnt ein zweiminütiger Countdown, während dem sich neue Geräte anmelden können. Die Verbindung wird durch das Drücken des Knopfs an dem anzumeldenden Gerät hergestellt.

Pin. Bei der Pin-Variante muss beim Hinzufügen eines Gerätes ein Pin eingegeben werden. Der Pin kann aus einer fest voreingestellten Zahlenkombination bestehen. Jedoch sind auch Implementierungen mit einem dynamischen Pin möglich. Mit Hilfe eines Monitors kann man sich den generierten, dynamischen Pin anzeigen lassen. Beim fest voreingestellten Pin befindet sich zum Beispiel ein Aufkleber auf der Unterseite des Access Points mit dem Pin. Will sich zum Beispiel ein neuer Client zum Netzwerk verbinden, erscheint bei ihm ein Fenster. In diesem muss der Pin eingegeben werden. Ist der Pin richtig, wird der Client anschließend mit dem Access Point verbunden. Sollte der Pin falsch sein, tritt ein Fehler auf.

Near Field Communication (NFC). Die NFC ermöglicht die drahtlose Übertragung von Daten über eine Strecke von einigen Zentimetern. Sie baut auf der Technologie von RFID-Chips auf. Hierbei wird die Konfiguration des WLANs im RFID-Chip gespeichert. Durch ein entsprechendes Lesegerät kann die Konfiguration durch den Client ausgelesen werden.

Die letzte Methode wird auch oft als Out-of-band Methode bezeichnet, da bei ihr die Informationen nicht über den WLAN Kanal übertragen werden. Die Wi-Fi Alliance vergibt die WPS-Zertifizierung für Access Points, wenn sie die Push Button Methode und die Pin-Eingabe unterstützen. Die drahtlosen Clients müssen nur die Pin-Methode ermöglichen.

Wenn man die verschiedenen Methoden nach ihrer Sicherheit unterscheidet, ergibt sich folgende Übersicht:

Push Button Configuration. Durch das Drücken des Knopfes am Access Point kann sich jeder beliebige Client innerhalb eines Zeitfensters von zwei Minuten anmelden. Angreifer könnten sich in dieser Zeit mit ihren Computern anmelden. Sie müssen nur wissen, dass der Button gedrückt wurde.

Pin. 2011 entdeckte S. Viehböck eine Schwachstelle bei der Pin Methode, wenn der Access Point einen fest voreingestellten Pin besitzt [6]. Die genaue Erläuterung, wie man den Pin wiederherstellen kann, folgt nach dieser Auflistung.

Near Field Communication (NFC). Angreifer müssen bis auf einige Zentimeter an den Access Point herankommen, damit sie den RFID-Chip lesen können.

Im nächsten Abschnitt wird gezeigt, wie man die fest voreingestellte Pin aufgrund eines Design-Fehlers des WPS Protokolls knacken kann.

Ein vereinfachter Ablauf des WPS Protokolls bei der Pin Eingabe sieht wie folgt aus [6]:

- Verbindungsaufbau zwischen Access Point und Client.
- EAP Initiation unter Verwendung von EAPoL zwischen Access Point und Client.
- Diffie-Hellman Schlüsselaustausch zum Aufbau einer sicheren Verbindung.
- Der Client sendet den ersten Teil des beim ihm eingegebenen Pins.
- Der Access Point bestätigt den ersten Teil.
- Der Client sendet den zweiten Teil des beim ihm eingegebenen Pins.
- Der Access Point bestätigt den zweiten Teil.
- Der Access Point sendet seine Konfiguration (SSID, Verschlüsselungsart, WLAN Schlüssel, usw.) an den Client.
- Der Client baut eine Verbindung zum Access Point mit den übermittelten Information auf.

Sollte bei diesem Ablauf ein Fehler auftreten, antwortet der Access Point sofort mit einer Fehlermeldung. Durch die sofortige Antwort des Access Points ist es einem Angreifer möglich, die Richtigkeit der Teile des Pins abzuleiten [6]:

- Erhält der Angreifer nach dem Senden des ersten Teils des eingegebenen Pins eine Fehlermeldung, weiß er, dass der erste Teil des eingegebenen Pins falsch ist.
- Ebenso verhält es sich mit dem zweiten Teil des Pins.

Wenn man den voreingestellten Pin genau betrachtet, stellt man fest, dass der achtstellige Pin in zwei Teile zerlegt werden kann. Es werden zuerst alle möglichen Kombinationen der ersten vier Stellen ausprobiert. Bei vier Stellen entspricht dies lediglich 10^4 Kombinationen. Der zweite Teil des Pins kann noch schneller erraten werden, da die achte Stelle des Pins eine Prüfsumme (Checksum) ist und somit aus den restlichen sieben Stellen berechnet werden kann. Die Aufteilung des Pins in zwei Teile ist in Abbildung 5 verdeutlicht. Aus dem zweiten Teil des Pins folgen also nur 10^3 Kombinationen, welche durchprobiert werden müssen. Insgesamt müssen nur 11000 Kombinationen ausprobiert werden bis der richtige Pin gefunden ist. S. Viehböck brauchte durchschnittlich 1,3 Sekunden pro Pin Versuch, das Ausprobieren aller Pins dauert somit circa vier Stunden. [6]

| | | | | | | | |
|------------------|---|---|---|--|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1. Teil des Pins | | | | 2. Teil des Pins <small>Checksum</small> | | | |

Abbildung 5: Aufteilung des WPS Pins in zwei Teile

Aufgrund dessen, dass WPS die Konfiguration des Access Points dem Client mitteilt, funktioniert dieser Angriff unabhängig von der verwendeten Verschlüsselungsart des

WLANs. Das reine Ausprobieren vieler Pins funktioniert nicht bei allen Access Points, weil manche Hersteller nach einer gewissen Anzahl von Pin-Eingabe-Versuchen die WPS Funktion automatisch für eine kurze Zeit deaktivieren. Der einzige Schutz vor dem Herausfinden des WPS Pins ist, die WPS Funktion im Access Point auszuschalten. [6]

3.3.2 Hole 196

Hole 196 ist eine andere Art eines Angriffs auf WLANs als die bisher vorgestellten. Das Besondere, was Hole 196 von den meisten Angriffen unterscheidet, ist, dass man Mitglied des Netzwerks sein muss. Es wird also nicht der Schlüssel des Netzwerks oder die AES Verschlüsselung angegriffen, sondern ein fehlendes Management ausgenutzt. Studien zeigen, dass die Zahl der internen Angriffe auf Netzwerke zunehmen und diese sehr hohe finanzielle Schäden verursachen können.

Die Lücke, die Hole 196 ermöglicht, wurde von AirTight Networks [14] auf Seite 196 des IEEE 802.11 Standard (2007) [11] gefunden, daher auch der Name „Hole 196“. Bei Hole 196 wird das fehlende Management des GTK ausgenutzt. Bei Datenpaketen, die mit dem PTK verschlüsselt sind, kann ein Client erkennen, ob die MAC-Adresse gefälscht oder der Inhalt verändert wurde. Jedoch hat der GTK keine dieser Eigenschaften. Normalerweise sollte nur der Access Point Daten mit dem GTK verschlüsseln und die Clients die Daten mit dem GTK entschlüsseln. Aber nichts im IEEE 802.11 Standard verhindert, dass ein Client gefälschte, mit dem GTK verschlüsselte Pakete verschickt. Dadurch ergeben sich drei mögliche Angriffe: Denial of Service, Einschleusen von Schadcode in andere Clients und ARP Poisoning. Das Besondere bei dieser Art des ARP Poisoning ist, dass dieses verschlüsselt erfolgt und über Funk übertragen wird. Beim bisherigen ARP Poisoning wurden die gefälschten ARP Pakete über den Access Point weitergeleitet und die Attacke konnte somit erkannt werden. Im Gegensatz dazu werden bei Hole 196 die gefälschten ARP Pakete direkt vom Client zum Client geschickt. Ein weiteres Problem ist, dass die gefälschten ARP Pakete in der Luft existieren und verschlüsselt sind. Kabelgebundene Sicherheitssoftwarelösungen können diese Art der Attacke also nicht erkennen. Zudem erkennen existierende Access Points kein abnormes Verhalten. Die einzige Möglichkeit, sich vor dem Angriff zu schützen, wäre bei der Verteilung des GTK jedem Client einen eigenen GTK zuzuweisen. [15]

In Abbildung 6 wird die Man-in-the-middle-Angriffsvariante von Hole 196 veranschaulicht. Zuerst (Schritt 1) sendet der Angreifer einem Client ein gefälschtes ARP Paket, das mit dem GTK verschlüsselt ist. Daraufhin sendet der Client die Daten wie üblich an den Access Point, wobei diese mit dessen PTK verschlüsselt sind (Schritt 2). Jedoch ist die Ziel-MAC-Adresse die des Angreifers. Der Access Point sendet die Daten mit dem PTK des Angreifers verschlüsselt an den Angreifer, da dies die Ziel-MAC-Adresse ist (Schritt 3). Als Letztes leitet der Angreifer die gesamten Daten wieder zum Access Point, damit dem Client keine Beeinträchtigung auffällt (Schritt 4).

4. VERWANDTE ARBEITEN

Lehembre gibt in seiner Ausarbeitung [16] eine gute Einführung zum Thema WLANs und deren Sicherheit. Die vielen Grafiken veranschaulichen die Abläufe sehr gut und bilden

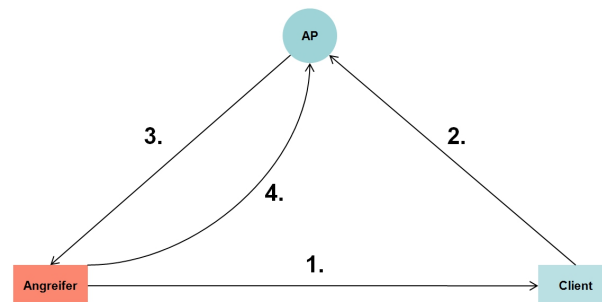


Abbildung 6: Man-in-the-middle-Angriffsvariante von Hole 196

ein Nachschlagewerk. Jedoch werden manchmal umfangreiche Details nicht genau erklärt.

A. Müller, H. Kinkelin, S. K. Ghai und G. Carle stellen in ihrer Arbeit [19] ein System vor, mit dem X.509 Zertifikate für neue Geräte eines Netzwerks ausgestellt werden können. Durch diese Zertifikate kann der Zugriff auf das Netzwerk geregelt werden.

Diese Ausarbeitung stellt die Funktionsweise von WPA 2 in einer vereinfachten Form dar. Des Weiteren werden verschiedene Angriffe auf WPA 2 erläutert. Das Ziel dieser Arbeit ist, dass man sich der Risiken und einhergehende Gefahren von WLANs bewusst wird und weiß, wie man sich gegen einige davon schützen kann.

5. ZUSAMMENFASSUNG

Aus heutiger Sicht gilt WPA 2 als sicher, wenn der Access Point richtig konfiguriert ist. Es sollte im privaten Umfeld ein möglichst langer und komplexer PSK gewählt werden, damit dieser nicht erraten werden kann. Als Verschlüsselungsart sollte WPA 2 mit CCMP gewählt werden, weil diese heutzutage als sicher gilt. Des Weiteren sollten der Access Point als auch alle Clients auf einem aktuellen Stand gehalten werden. Nicht verwendete Protokolle, wie WPS, sollte man am besten ausschalten oder nur kurz, wenn man sie benötigt, einschalten. Auch sollte bei Laptops das WLAN vollkommen ausgeschaltet sein, wenn es nicht gebraucht wird. Es bleibt abzuwarten, ob neue Schwachstellen gefunden werden.

6. LITERATUR

- [1] S. Fluhrer, I. Mantin und A. Shamir: *Weaknesses in the Key Scheduling Algorithm of RC4*, 2001
- [2] E. Tews, R.-P. Weinmann und A. Pyshkin: *Breaking 104 bit WEP in less than 60 seconds*, In WISA'07 Proceedings of the 8th international conference on Information security applications, pages 188-202, Springer, 2007
- [3] J. Edney und W. A. Arbaugh: *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*, Boston, MA, Addison-Wesley, 2004
- [4] *Microsoft Windows Silent Adhoc Network Advertisement*, <http://www.nmrc.org/pub/advise/20060114.txt>, S. Nomad, Nomad Mobile Research Centre, 2006
- [5] A. Bogdanov, D. Khovratovich und C. Rechberger: *Biclique Cryptanalysis of the Full AES*, 2011

- [6] S. Viehböck: *Brute forcing Wi-Fi Protected Setup*, 2011
- [7] *Wi-Fi Alliance*, <http://www.wi-fi.org>
- [8] *Wi-Fi Alliance FAQ*,
<http://www.wi-fi.org/knowledge-center/faq>
- [9] *Aircrack-ng*, <http://www.aircrack-ng.org/>
- [10] *Aircrack-ng Cracking WPA*, http://www.aircrack-ng.org/doku.php?id=cracking_wpa
- [11] IEEE 802.11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, 2007
- [12] *Karma*, <http://www.theta44.org/karma/index.html>
- [13] D. A. D. Zovi und S. A. Macaulay: *Attacking Automatic Wireless Network Selection*, In Proceedings from the Sixth Annual IEEE SMC, pages 365-372, 2005
- [14] *AirTight Networks*,
<http://www.airtightnetworks.com>
- [15] *AirTight Networks - WPA2 Hole196 Vulnerability*,
<http://www.airtightnetworks.com/WPA2-Hole196>
- [16] G. Lehembre: *Wi-Fi security - WEP, WPA and WPA2*, 2005
- [17] *Amazon Elastic Compute Cloud (Amazon EC2)*,
<http://aws.amazon.com/de/ec2/>
- [18] *Tom's Hardware*, A. Ku,
<http://www.tomshardware.de/WLAN-Einbruch-Hacken-Brute-Force,testberichte-240879.html>
- [19] A. Müller, H. Kinkelin, S. K. Ghai und G. Carle: *An Assisted Device Registration and Service Access System for future Home Networks*, 2009

Device Fingerprinting mit dem Web-Browser

Thomas Pieronczyk
Betreuer: Ralph Holz
Seminar Future Internet SS2012
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: thomas.pieronczyk@tum.de

KURZFASSUNG

Nahezu jeder Computer besitzt heutzutage einen Internetanschluss. Web-Browser sind dabei die Hauptschnittstelle zum World Wide Web. Neben all ihren Vorzügen stellen sie jedoch auch ein ernstzunehmendes Sicherheitsrisiko für die Privatsphäre dar. Der Grund dafür ist die Preisgabe von Informationen wie z. B. installierte Plugins, System-Schriftarten, das genutzte Betriebssystem, oder der genutzte Browser. Diese Informationen können genutzt werden um Nutzer im Internet zu identifizieren.

In dieser Arbeit wird die Identifizierung von Nutzern im Internet anhand der von ihnen hinterlassenen digitalen Spuren (sog. Privacy Footprints) erörtert. Es wird aufgezeigt wie digitale Spuren im Internet ausgelesen und gespeichert werden und welche Gefahren dadurch für die Privatsphäre entstehen können. Es werden Methoden vorgestellt, mit denen man aus den gesammelten Spuren digitale Fingerabdrücke (sog. Device-Fingerprints) erzeugen kann. Im Anschluss wird das Thema Device Fingerprinting mit Hilfe von Web-Browsern genauer betrachtet. Basis für diese Betrachtung ist die Publikation „How unique is your web browser“ von Peter Eckersley [6].

Es wird aufgezeigt, wie Internetnutzer über frei zugängliche Informationen relativ eindeutig identifiziert und verfolgt werden können und welche Möglichkeiten existieren, diese Identifizierung zu erschweren.

Peter Eckersley zeigt, dass man einerseits den Web-Browser gut verwenden kann um Internet-Nutzer zu identifizieren, andererseits aber auch, dass die bisher gesammelten Versuchsdaten nicht ausreichend, bzw. nicht geeignet sind, um eine globale Identifizierbarkeit von Internetnutzern bestätigen zu können.

Schlüsselworte

Device Fingerprinting, Web-Browser, Digitale Spuren im Internet, Datenschutz, Privacy

1. EINLEITUNG

In der heutigen Gesellschaft stellen Informationen aller Art ein starkes Machtinstrument zur Verfügung. In den Medien wird regelmäßig über Datenschutzverletzungen und das Erbeuten von Informationen berichtet. Diese Informationen werden gesammelt und beispielsweise für personalisierte Werbung verwendet.

Ein immer größer werdender Teil der alltäglichen Aktivitäten wie z. B. Einkäufe wird heute über das Internet abgewickelt. Damit einhergehend erhöht sich auch der Anteil der preisgegebenen Informationen [10, S. 1]. Obwohl kritische

Informationen wie Bank- und Kreditinformationen in der Regel ausreichend vor unbefugten Zugriffen geschützt werden, können vermeintlich unwichtige Informationen von jedem Webserver mühelos ausgelesen und gespeichert werden. Mit IP-Adressen, Cookies oder digitalen Fingerabdrücken ist es möglich, unbedenkliche Informationsschnipsel zu wertvollen Informationen zu kombinieren. Diese Fingerabdrücke werden von Peter Eckersley in der Publikation „How unique is your web browser“ [6] behandelt und sind Hauptgegenstand dieser Arbeit.

Digitale Fingerabdrücke werden aus Informationen des Nutzers generiert, welche unbemerkt und ohne explizite Zustimmung beim Besuch von Webseiten ausgelesen werden können. Mit Hilfe der Identifizierung durch Fingerabdrücke können Internetnutzer verfolgt, ihre Daten gesammelt und diese dann an Drittanbieter weitergereicht werden.

Es werden Möglichkeiten vorgestellt, digitale Fingerabdrücke zu generieren und aufgezeigt, wie eindeutig sich Nutzer im Internet identifizieren lassen, wie beständig digitale Fingerabdrücke sind und welche Gegenmaßnahmen man ergreifen kann um seinen persönlichen digitalen Fingerabdruck zu minimieren.

2. SPUREN IN DIGITALEN MEDIEN

Jeder Aufruf einer Webseite hinterlässt Spuren. Diese Informationen verletzen im Allgemeinen nicht die Privatsphäre des Nutzers. Sie können jedoch über längere Zeit hinweg gesammelt und gespeichert werden und anschließend dafür genutzt werden die Anonymität des Nutzers aufzuheben und die Privatsphäre des Nutzers zu gefährden. In den folgenden Abschnitten wird die Bedeutung von Anonymität erörtert, die Begriffe „Privacy Footprints“ und „Device Fingerprinting“ definiert und auf die Gefahren beider Themen eingegangen.

2.1 Was bedeutet Anonymität im Internet?

Sich anonym im Internet zu bewegen bedeutet, dass man keine Informationen hinterlässt, mit denen man identifiziert und verfolgt werden kann.

Doch warum ist Anonymität im Internet so wichtig? Auf den ersten Blick suggeriert das Verlangen nach Anonymität die Absicht, illegalen Aktivitäten nachzugehen. Anonymität im Internet ist jedoch für jeden Internetnutzer essentiell. Anonymität im Internet schützt die Privatsphäre einer Person. Diese sichert das Recht auf freie Entfaltung der Persönlichkeit, welches eines der Grundrechte im deutschen Grundgesetz beschreibt [13, Art 2. Abs. 1]. Des Weiteren schützt sie das vom Bundesverfassungsgericht erwähnte, im

Grundgesetz jedoch nicht erwähnte „Recht auf informationelle Selbstbestimmung“ [2].

Sie verhindert, dass private Informationen jeglicher Art an Dritte weitergegeben werden, dass Interessen und Nutzungsverhalten aufgezeichnet werden und verhindert damit Konsequenzen wie finanziellen oder sogar physischen Schaden, Rufschädigung oder Diskriminierung [4].

Die im Internet verwendeten Technologien ermöglichen es auf unterschiedliche Art und Weise, Nutzer über ihre Informationen zu verfolgen und damit die Anonymität im Internet auszuhebeln. Die folgenden Abschnitte behandeln diese Problematik.

2.1.1 Privacy Footprints

Privacy Footprints kann man sich als Fußspuren oder Fahrten im Internet vorstellen. Sie beschreiben, zu welchem Ausmaß scheinbar unabhängige Webseiten über sogenannte Aggregatorknoten gemeinsamen Zugriff auf nutzerbezogene Informationen erhalten. Ein größerer Footprint bedeutet, dass mehr Informationen auf verschiedenen Aggregatorknoten gespeichert sind. Der Austausch von Informationen wird meist über Cookies bewerkstelligt. Folgende Schritte zusammen mit der Abbildung 1 erläutern anhand eines Beispiels die Funktionsweise von Aggregatorknoten [10, S. 1]:

1. Der Nutzer besucht Seite A und schickt damit Daten X an Server A.
2. Server A speichert Daten X im Aggregatorknoten.
3. Parallel zu Schritt 1. und 2. speichert Server A einen Cookie mit der Adresse des Aggregatorknotens auf dem Computer des Nutzers.
4. Der Nutzer besucht anschließend Seite B und schickt damit Daten Y auf Server B.
5. Server B liest den zuvor gespeicherten Cookie vom Computer des Nutzers und kennt dadurch den Aggregatorknoten.
6. Server B vergleicht Daten Y mit Daten X des Aggregatorknotens und kann den Nutzer reidentifizieren.
7. Server B kann durch die Reidentifikation beispielsweise nutzerbezogene Werbung anzeigen.

Zwei einflussreiche und bekannte Vertreter dieser Aggregatorknoten sind **doubleclick.net** und **google-analytics.com**. In Veröffentlichung [10] wurde in einem Experiment festgestellt, dass unter 1075 unabhängigen Webseiten bei **doubleclick.net** insgesamt 201 (19%) Web-Seiten und bei **google-analytics.com** insgesamt 78 Web-Seiten (7%) über Aggregatorknoten miteinander verbunden waren.

2.1.2 Device Fingerprinting

Unter Device Fingerprinting versteht man die Generierung einer Identifikation (eines Fingerabdrucks) eines Betriebssystems (Software) oder einer Klasse von Geräten (Hardware), ohne die Kooperation der zu identifizierenden Geräte, bzw. Software [17, S. 1]. Dies ist mit unterschiedlichsten Geräten möglich. Es ist z. B. bereits länger bekannt, dass man

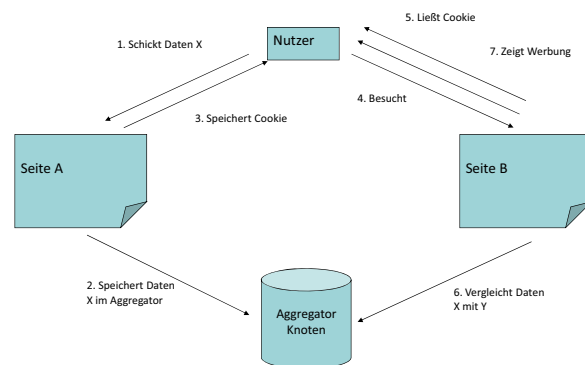


Abbildung 1: Ablauf beim Erzeugen von Privacy Footprints

Schreibmaschinen an den unterschiedlichen Ausfransungen der gedruckten Buchstaben unterscheiden kann [12]. Das Sensorrauschen in Bildern ermöglicht ebenfalls eine Identifizierung von Digitalkameras [8]. Neben der Identifizierung über Hardware können auch Fingerabdrücke mit Hilfe von Software generiert werden [6, S. 1]. Beispiele hierfür sind:

- CSS Font Detector [14]
- CSS History Hack [1]
- Fingerprinting mit dem Browser [6]

Mit den oben aufgeführten Hilfsmitteln lassen sich digitale Fingerabdrücke anfertigen, mit deren Hilfe man Internetnutzer identifizieren und verfolgen kann.

2.2 Gefahren von Privacy Footprints

Die größte Gefahr, die von Privacy Footprints ausgeht, ist die Reidentifikation eines Nutzers [10, S. 1]. Mit ihr können private, vermeintlich anonymisierte Daten mit harmlosen, jedoch personalisierten Daten kombiniert werden und damit persönliche Informationen offengelegt werden [15]. Folgendes Beispiel soll den Ablauf einer Reidentifikation veranschaulichen: Der erste Datensatz ist eine medizinische Akte, die lediglich das Geburtsdatum, das Geschlecht, eine geographische Ortsangabe und medizinische Befunde enthält (anonymer Datensatz). Dieser wird durch Seite A (siehe Grafik 1) repräsentiert und im Aggregatorknoten gespeichert. Der zweite Datensatz enthält Informationen über den Fahrzeughalter eines Kraftfahrzeugs (personalisierter Datensatz). Dieser wird durch Seite B repräsentiert und ebenfalls im Aggregatorknoten hinterlegt. Die Kombination beider Datensätze im Aggregatorknoten führt dazu, dass man die medizinische Akte einer Person namentlich zuordnen kann. Solche Offenlegungen von Informationen können zu finanziellen Schäden oder Rufschädigung der Person führen [10, S. 1].

2.3 Realisierungen von Device Fingerprinting

In den folgenden Abschnitten gehen wir auf die in 2.1.2 vorgestellten Fingerprinting-Beispiele etwas genauer ein.

2.3.1 CSS Font Detector

Der CSS Font Detector ist JavaScript-Code, mit dem man – mit Hilfe von Cascading Style Sheets (CSS) – verfügbare Schriftarten in einem Browser identifizieren kann. Der eigentliche Zweck des Detektors ist es, besseres Design von Web-Seiten durch das Setzen von passenden Schriftarten zu ermöglichen. Der Algorithmus nutzt die Gegebenheit aus, dass jedes Zeichen eine unterschiedliche Pixelgröße in unterschiedlichen Schriftarten besitzt (siehe Abbildung 2).

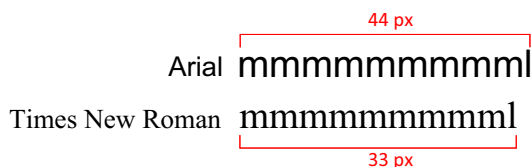


Abbildung 2: Pixellängen für unterschiedliche Schriftarten

Die Funktionsweise ist folgende: Es werden 3 Strings in Monospace, Sans-serif und Sans erzeugt und die Pixelbreite dieser Strings notiert. Anschließend wird ein String mit der zu testenden Schriftart als primäre Schriftart und einer der generischen Schriftarten als Fall-Back Schriftart erzeugt. Die Pixelbreite wird mit den generischen Schriftarten verglichen. Ist die Pixelbreite unterschiedlich, existiert die Schriftart, ist sie gleich, existiert die Schriftart nicht [14].

2.3.2 CSS History Hack

Der CSS History Hack erkennt, wie beim CSS Font Detector mit Hilfe von CSS, ob ein Nutzer bekannte Webseiten besucht hat oder nicht. Dabei besucht der Nutzer eine präparierte Webseite, auf der durch eine festgelegte Menge von Webseiten iteriert wird. Bei jeder Webseite wird überprüft, ob der Nutzer die Seite besucht hat, oder nicht. Der Algorithmus funktioniert folgendermaßen:

1. Es wird eine CSS Regel festgelegt, die einen besuchten Link auf eine festgelegte Farbe setzt. Beispiel:

```
<style>  
    a:visited { color: red; }  
</style>
```

2. Im nächsten Schritt werden mit JavaScript HTML-Link-Elemente aus einer selbst definierten Menge von Webseiten (www.google.com, www.facebook.com, etc.) erzeugt.
3. Anschließend iteriert man durch die generierten HTML Elemente und vergleicht die Farbe mit der in Schritt 1. festgelegten Farbe für besuchte Links. Entspricht die Farbe des Links der CCS Regel, hat der Nutzer die Web-Seite bereits besucht.

Die zu untersuchenden Elemente müssen nicht sichtbar in die Seite eingebaut werden und werden nach Abschluss des Tests wieder entfernt. Aus diesem Grund kann man den Test auch vom Nutzer unbemerkt durchführen. [1].

3. DEVICE FINGERPRINTING MIT DEM BROWSER

Der CSS Font Detector und der CSS History Hack sind zwei Möglichkeiten, Informationen über einen Nutzer im Internet zu erhalten. Kombiniert man diese mit weiteren Informationen, kann man ziemlich präzise digitale Fingerabdrücke eines Internetnutzers anfertigen. Web-Browser geben viele dieser Informationen ohne Kenntnis oder Zustimmung des Nutzers preis. In den folgenden Abschnitten erörtern wir anhand der Arbeit von Peter Eckersley, wie man mit Hilfe des Web-Browsers digitale Fingerabdrücke erzeugen kann, wie eindeutig diese Fingerabdrücke einen Nutzer identifizieren können, wie stabil diese Fingerabdrücke gegen Veränderungen sind und wie man sich gegen die Verfolgung über digitale Fingerabdrücke schützen kann.

3.1 Methodik

In den folgenden Abschnitten werden wir die Funktionsweise des Browser-Fingerprint-Algorithmus, die mathematischen Grundlagen, die hinter diesem Algorithmus stecken und die Aufbereitung der Datensätze analysieren.

3.1.1 Browser-Fingerprint-Algorithmus

Grundlage der Publikation von Peter Eckersley [6] ist ein selbst konstruierter Algorithmus für die Generierung von digitalen Fingerabdrücken, mit deren Hilfe über die Webseite <http://panopticlick.eff.org> bisher insgesamt 2.102.470 (Stand 25.03.2012) digitale Fingerabdrücke gesammelt wurden.

Beim Besuch der Seite werden zuerst anonymisiert die Konfiguration samt Versionsinformationen von Betriebssystem, Browser und Plugins gespeichert. Anschließend werden die Informationen mit den Datensätzen in der Datenbank verglichen. Im Ergebnis wird dem Nutzer angezeigt, wie eindeutig seine Browserkonfiguration innerhalb des Testdatensatzes ist.

3.1.2 Mathematische Grundlagen

Die mathematischen Grundlagen für die Ermittlung der Eindeutigkeit eines digitalen Fingerabdrucks sind Informationsgehalt und Entropie aus dem Bereich der Informationstheorie nach Claude E. Shannon. Die Identifizierbarkeit einer Browserkonfiguration hängt dabei direkt von der Auftrittswahrscheinlichkeit $P(m)$ einer Messvariablen innerhalb einer endlichen Menge von Messvariablen $m \in M$ ab. Je höher die Wahrscheinlichkeit des Auftretens einer Messvariable, desto geringer ist der Informationsgehalt, desto schwieriger lässt sie sich identifizieren. Umgekehrt sind Messvariablen mit geringer Auftrittswahrscheinlichkeit und dem daraus resultierenden hohen Informationsgehalt sehr leicht zu identifizieren. Dies ist durch die Verwendung von $-\log_x(y)$ in den Formeln für Informationsgehalt und Entropie begründet. Informell kann man sagen: je mehr Browser-Konfigurationen eine Messvariable mit hohen Auftrittswahrscheinlichkeiten enthalten, desto schwieriger sind sie zu identifizieren. In den folgenden Abschnitten werden Informationsgehalt und Entropie noch etwas genauer betrachtet.

Formell betrachtet, ist der Informationsgehalt eines Versuchs die Menge an Information, die benötigt wird, um zu wissen, dass ein bestimmtes Ereignis x einer Zufallsvariablen X eingetreten ist [9, S. 23]. In Eckersleys Arbeit gibt die Informationsmenge Auskunft über die Identität eines einzelnen

Web-Browsers $x \in X$. Formel 1 beschreibt den Informationsgehalt für eine Messvariable $P(f_n)$ und für einen Web-Browser $\{F(x) = f_n\}$:

$$I(F(x) = f_n) = -\log_2(P(f_n)) \quad (1)$$

Die Basis für die Informationsmenge ist 2, da die einzelnen Experimente jeweils nur den Ausgang „wahr“ oder „falsch“ haben können (Beispielfrage: Sind Cookies aktiviert?). Die Informationseinheit für Logarithmen zur Basis zwei ist **bit**. Die Entropie liefert den maximalen Informationsgehalt für alle x Ereignisse einer Zufallsvariablen X [9]. In Eckersleys Arbeit ist damit der Informationsgehalt einer Messvariable über alle Browser X gemeint. Formel 2 beschreibt die Entropie für eine Messvariable $P(f_n)$ und für die Menge aller untersuchten Browser:

$$H(F) = -\sum_{n=0}^N P(f_n) \log_2(P(f_n)) \quad (2)$$

Möchte man den Informationsgehalt von mehr als einer Messvariablen $s \in S$ berechnen, muss man unterscheiden, ob die Messvariablen von einander abhängig sind oder nicht. Im Falle der Unabhängigkeit der Messvariablen wird Formel 3 angewendet:

$$I_s(f_n, s) = -\log_2 P(f_n, s) \quad (3)$$

Sind die Messvariablen hingegen voneinander abhängig, wird Formel 4 angewendet:

$$I_{s+t}(f_{n,s}, f_{n,t}) = -\log_2(P(f_n, s|f_n, t)) \quad (4)$$

Ein Beispiel für statistisch abhängige Messvariablen wäre die Identifikation eines Flash Block Plugins mit $P(\text{Schriftart} = \text{„nicht gefunden“} \mid \text{„Flash“} \in \text{Plugins})$. Die Entropie für mehrere Messvariablen kann mit Formel 5 berechnet werden:

$$H_s(F_s) = -\sum_{n=0}^N P(f_{s,n}) \log_2(P(f_{s,n})) \quad (5)$$

Es ist nun bekannt, wie man den Informationsgehalt und die Entropie von Web-Browser-Konfigurationen berechnen kann. Als Nächstes muss geklärt werden, wie die ermittelten Kennzahlen zu interpretieren sind. Man muss herausfinden, wie viel bits an Entropie benötigt werden um einen einzelnen Nutzer (bzw. seinen Web-Browser) im Datensatz eindeutig identifizieren zu können. Dazu wird zuerst die Auftrittswahrscheinlichkeit $P(h)$ einer einzelnen Konfiguration innerhalb des Datensatzes benötigt. Im Datensatz von Panopticlick mit 2.102.470 Einträgen (Stand 25.03.2012) beträgt die Auftrittswahrscheinlichkeit eines Datensatzes: $P(h) = 1/2.102.470$. Der Informationsgehalt beträgt dann:

$$S = -\log_2(P(h)) = -\log_2(1/2.102.470) = 21.003654 \text{ bits} \quad (6)$$

Aufgerundet bedeutet dies, dass ein Nutzer mit einem Fingerabdruck mit 22 bits Informationsgehalt eindeutig im Datensatz von Panopticlick identifizierbar ist. Wendet man dieses Erkenntnis auf die Population der Erde mit ca. 7.039.632.000 Menschen (Stand 2012 [16]) aus, erhält man eine Auftrittswahrscheinlichkeit von $P(h) = 1/7.039.632.000$ und damit einen Informationsgehalt von:

$$S = -\log_2(P(h)) = -\log_2(1/7.039.632.000) = 32,71 \text{ bits} \quad (7)$$

Aufgerundet benötigt man also 33 bits an Entropie um eine Person auf der Erde eindeutig zu identifizieren [5].

3.1.3 Datensammlung und Vorbearbeitung

Die Datensätze, auf die sich die Publikation von Peter Eckersley stützt, wurden im Zeitraum 27.01.2010 – 15.02.2010 über die Webseite <http://panopticlick.eff.org> gesammelt. Es wurden folgende Daten erhoben:

- Ein Browser-Fingerprint
- Eine HTTP-Cookie ID, welche 3 Monate gespeichert wurde
- Ein HMAC-Wert der IP-Adresse
- Ein HMAC-Wert der IP-Adresse mit gelöschtem letzten Oktet (Subnetzadresse)

Die neben dem Fingerabdruck gespeicherte HMAC (Keyed-Hash Message Authentication Code) der IP-Adresse, die HMAC des Subnetzes und die Cookie ID dienen der Verfolgung von Besuchern, welche die Seite mehr als einmal besucht hatten. Die HMACs werden dazu verwendet die IP-Adresse und die Adresse des Subnetzes zu anonymisieren. Die HMAC wird dabei aus einer Nachricht (in dem Fall die IP-Adresse) und einem privaten Schlüssel nach einer festgelegten Hashfunktion berechnet. Das Hashing ohne Schlüssel würde bei der IPv4-Adress-Länge von 2^{32} Bit keine ausreichende Sicherheit bieten [7].

Der Datensatz wurde vor den Messungen auf Doppeleinträge untersucht und bereinigt. Bei Besuchern mit aktivierten Cookies konnten doppelte Datensätze leicht erkannt und entfernt werden. Bei Besuchern mit deaktivierten Cookies wurde angenommen, dass Datensätze mit gleicher IP-Adresse und identischem Fingerabdruck den selben Browser repräsentieren und wurden als Folge dessen auf einen Eintrag reduziert. Bei letzteren gab es jedoch auch eine Ausnahme. Es wurden im Laufe des Experiments identische (Fingerabdruck, IP) Tupel mit unterschiedlichen Cookies registriert. Dies bedeutet, dass ein Besucher mit der gleichen IP-Adresse wiederholt <http://panopticlick.eff.org> mit Cookie A, dann mit Cookie B und anschließend wieder mit Cookie A besucht hatte. Diese Charakteristik wurde bei 3.5% aller IP-Adressen festgestellt.

Zu Beginn der Datensammlung wurden außerdem noch einige Datensätze auf Grund von Fehlern im Algorithmus entfernt.

Aus anfänglich 1.043.426 Datensätzen wurden 470.161 Fingerabdrücke mit minimalen Doppeleinträgen für die Messungen extrahiert [6, S. 7–8].

3.1.4 Erhobene Daten

In Tabelle 1 sind alle Messvariablen samt Quelle aufgelistet, die mit dem Browser-Fingerprint-Algorithmus gesammelt wurden. Alle Informationen werden über den Browser und ohne Zutun des Nutzers den aufgerufenen Webseiten zur Verfügung gestellt.

Die Datensätze könnten um zusätzliche Informationen erweitert werden, wurden jedoch in dem Algorithmus aus folgenden Gründen nicht berücksichtigt [6, S. 7]:

Tabelle 1: Aufschlüsselung der erhobenen Daten [6, S. 5]

| Variable | Quelle |
|--|--|
| User Agent | Über HTTP übertragen, vom Server aufgezeichnet |
| HTTP Accept Header | Über HTTP übertragen, vom Server aufgezeichnet |
| Cookies aktiviert? | Aus HTTP abgeleitet, vom Server aufgezeichnet |
| Bildschirm Auflösung | JavaScript AJAX Post |
| Zeitzone | JavaScript AJAX Post |
| Browser Plugins, Plugin Versionen und MIME Typen | JavaScript AJAX Post |
| System Schriftarten | Flash oder Java Applet, mit JavaScript/AJAX ausgelesen |
| Supercookie Test | JavaScript AJAX Post |

1. Mangelnde Kenntnis oder Mangel an Zeit zur korrekten Implementation (Bsp.: Microsoft Active X)
2. Die Informationen wurden als nicht ausreichend stabil erachtet
3. Die Information kann nur durch explizite Zustimmung des Nutzers erlangt werden

3.2 Ergebnisse

In dem von Peter Eckersley erhobenen Datensatz sind 83,6% der Fingerabdrücke eindeutig identifizierbar, 8,2% teilen sich mit 2 bis 9 Fingerabdrücken die gleiche Konfiguration und 8,1% teilen sich mit 10 Fingerabdrücken die gleiche Konfiguration. Nur ein Bruchteil von unter 0,1% befindet sich in einer Konfigurationsmenge größer 10 (siehe Abbildung 3) [6, S. 9]. Diese Zahlen sind laut Eckersley etwas verfälscht, da er die Hauptnutzer von <http://panopticlick.eff.org> als technisch versiert und sicherheitsbewusst ansieht. Für realistischere Ergebnisse mit durchschnittlichen Nutzern würde die Eindeutigkeit der Fingerabdrücke geringer ausfallen [6, S. 7].

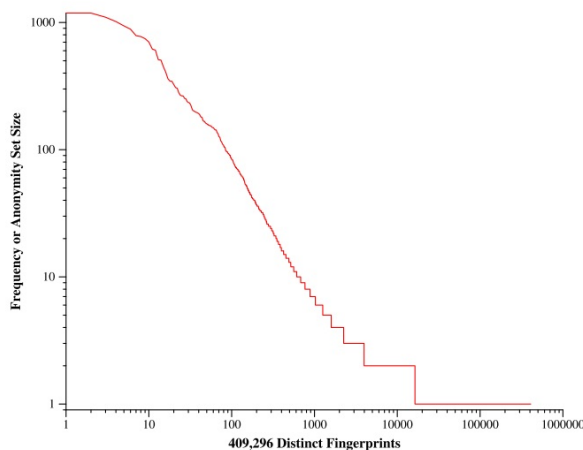


Abbildung 3: Verteilungsfunktion der ermittelten Fingerabdrücke [6, S. 8]

In der Verteilungsfunktion des Informationsgehalts für verschiedene Browser in Abbildung 4 ist leicht zu erkennen, dass der Großteil der Web-Browser in Bezug auf digitale Fingerabdrücke schlecht abschneidet. 90% der modernen Webbrowser sind eindeutig identifizierbar. Gut abgeschnitten haben Web-Browser, bei denen JavaScript deaktiviert ist oder

Web-Browser von mobilen Geräten wie iPhone oder Android. Web-Browser von mobilen Geräten sind durch limitierte Pluginfähigkeiten viel einheitlicher und deshalb schwer zu unterscheiden. Ihr Nachteil sind jedoch die mangelnde Kontrolle über angelegte Cookies, welcher die Verfolgung von Nutzern wiederum erleichtert [6, S. 9].

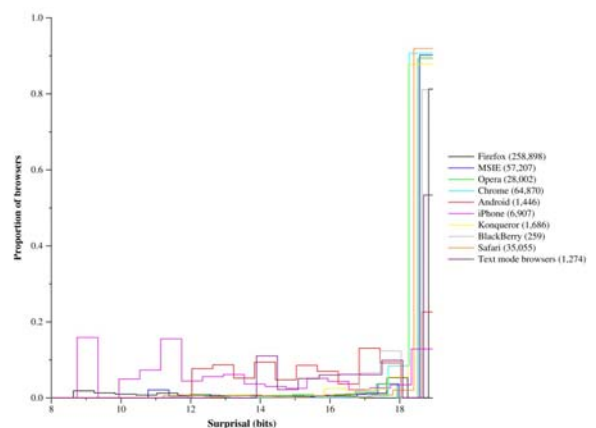


Abbildung 4: Informationsgehalt für verschiedene Web-Browser [6, S. 9]

Eckersley berechnete auch die Fingerabdrücke der Web-Browser für jede einzelne Messvariable (siehe Tabelle 2). Den höchsten Grad an Identifizierbarkeit zeigen Plugins und Schriftarten, gefolgt von User-Agent-Informationen (String mit Informationen über Betriebssystem, Browser, und deren Versionen), HTTP Accept Header (nennt dem Browser den Medientyp des HTTP-Response), Bildschirmauflösung, Zeitzone, Supercookies (Cookies für den Adobe Flash Player [3]) und Cookies.

3.2.1 Fingerabdruck-Charakteristiken

Neben den direkt ablesbaren Charakteristiken zum Identifizieren von Nutzern hat Peter Eckersley noch weitere, subtilere Charakteristiken entdeckt, mit denen sich Nutzer leicht identifizieren lassen. Ein Beispiel ist die JavaScript-Konfiguration eines Browsers. Bei deaktiviertem JavaScript werden Standardwerte für Video, Plugins, Schriftarten und Supercookies festgelegt. Die Präsenz dieser Werte zeigt, dass JavaScript deaktiviert ist.

Ein weiteres Beispiel sind Browser, die Flash in den Plugins auflisten, bei denen es jedoch nicht möglich ist, die Systemschriftarten auszulesen (System-Schriftarten werden im Browser-Fingerprint-Algorithmus mit Hilfe von Flash aus-

Tabelle 2: Durchschnittliche Entropie einzelner Variablen [6, S. 17]

| Variable | Durchschn. Eigeninformation |
|--|-----------------------------|
| User-Agent | 10,0 |
| HTTP-Accept-Header | 6,09 |
| Cookies aktiviert? | 0,353 |
| Bildschirm-Auflösung | 4,83 |
| Zeitzone | 3,04 |
| Browser Plugins, Plugin Versionen und MIME-Typen | 15,4 |
| System Schriftarten | 13,9 |
| Supercookie verfügbar | 2,12 |

gelesen). Diese Charakteristik ist ein eindeutiges Indiz für die Nutzung eines Flash Blockers und verstärkt die Eindeutigkeit von Fingerabdrücken.

Das letzte Beispiel handelt von User-Agent-Spoofing-Plugins, mit denen sich User-Agent-Informationen wie z. B. Betriebssystem oder Browser verfälschen lassen. In der Untersuchung wurden Datensätze entdeckt, welche sich als iPhone ausgaben, jedoch auch das Flash-Plugin installiert hatten (iOS unterstützt zur Zeit kein Flash). Die geringe Anzahl der Fingerabdrücke, die diese Charakteristiken aufzeigen, verstärkt die Identifizierbarkeit [6, S. 4].

3.2.2 Globale Extrapolation

Peter Eckersley hat in seiner Publikation auch die Frage behandelt, ob sich der über <http://panopticklick.eff.org> erhobene Datensatz auf globaler Ebene extrapolieren lässt.

Er geht auf die Aussage von Mayer [11] ein, dass sich mit einer Stichprobe auf Grund der Multinomialverteilung keine Aussagen über die globale Eindeutigkeit von Web-Browser-Fingerabdrücken machen lassen. Er unterstützt Mayers These, behauptet jedoch, dass sie im Bereich Privatsphäre etwas zu optimistisch ausgelegt sei. Eckersley geht davon aus, dass man mit einer geeigneten Stichprobe von Fingerabdrücken und Monte-Carlo-Simulation mindestens ein globales Mengenverhältnis von Eindeutigkeiten ermitteln könnte.

Dieser Ansatz ist jedoch für eine globale Extrapolation mit dem Panopticklick Datensatz nicht umsetzbar, weil er hauptsächlich aus Datensätzen von technisch versierten, sicherheitsbewussten Internetnutzern besteht. Diese würden ein verzerrtes Ergebnis globaler Fingerabdrücke wiedergeben. Der Datensatz müsste um neutrale Einträge ausgeweitet werden, um ein realistisches Ergebnis zu liefern [6, S. 10–11].

3.3 Stabilität von Fingerabdrücken

Die Mitverfolgung von wiederkehrenden Besuchern von <http://panopticklick.eff.org> (siehe 3.1.3) zeigt, dass sich digitale Fingerabdrücke schnell ändern können. Diese Veränderungen werden z. B. durch Updates des Browsers, Updates der Plugins, Deaktivieren von Cookies, Installation von neuen Fonts, oder durch das Anschließen eines externen Monitors (Veränderung der Auflösung) hervorgerufen. Veränderte Fingerabdrücke eines Nutzers werden mit Hilfe der gespeicherten HTTP-Cookie ID (siehe 3.1.3) und einem einfachen Algorithmus erkannt. Der Algorithmus erkannte Veränderungen bei 65% aller Fälle, lag bei 0,56% falsch und war durch zu große Umstellungen der Browser-Konfiguration bei

35% aller Fälle nicht in der Lage, eine Veränderung zu erkennen. Insgesamt 37,4% der wiederkehrenden Besucher mit aktivierten Cookies besaßen einen veränderten Fingerabdruck. Die Änderungsrate bei <http://panopticklick.eff.org> fällt laut Eckersley etwas höher aus als in der realen Welt, da das Experiment die Besucher zum Ändern ihrer Konfigurationen animiert. Dieses Experiment zeigt, dass Veränderungen der Konfiguration nicht zuverlässig vor Identifikation oder Verfolgung schützen [6, S. 11–13].

3.4 Mögliche Schutzmaßnahmen

in diesem Abschnitt werden Möglichkeiten aufgezeigt, wie man sich gegen das Erzeugen von digitalen Fingerabdrücken schützen kann. Es ist anzumerken, dass man immer einen Kompromiss aus Schutz und Bequemlichkeit / Nutzbarkeit eingehen muss. Je mehr Schutzfunktionalitäten beim Surfen im Internet verwendet werden, desto langsamer werden die Internetseiten aufgebaut und desto mehr Nutzerinteraktion ist für das Freischalten von Inhalten notwendig. Teilweise können Inhalte nicht korrekt bzw. teilweise oder überhaupt nicht dargestellt werden. Man kann dies sehr leicht feststellen, indem man z. B. <http://www.facebook.com/> mit deaktiviertem JavaScript aufruft.

Darüber hinaus ist noch anzumerken, dass nicht alle Schutzmaßnahmen auch zu dem gewünschten Ergebnis führen, nicht identifizierbar zu sein. Es existieren Produkte, die Informationen eines Nutzers im Internet verschleiern, welche das Identifizieren eines Nutzers durch ihre geringe Verbreitung sogar erleichtern. Kontraproduktive Verschleierungstechniken haben wir mit Flash Blocker und User Agent Spoofing bereits in Kapitel 3.2.1 beschrieben. Beispiele für Produkte mit kontraproduktiven Verschleierungstechniken sind der Privoxy Web Proxy (<http://www.privoxy.org>), welcher in Eckersleys Untersuchungen durchschnittlich 15.5 bits Entropie aufwies und der Privacy-Browser Browzar (<http://www.browzar.com/>), bei dem alle Datensätze eindeutig identifizierbar waren.

Es existieren jedoch auch Lösungen, die den digitalen Fingerabdruck im Internet tatsächlich minimieren. Zwei Werkzeuge, die das Erzeugen von Fingerabdrücken im Experiment erheblich erschwert haben waren einerseits das Anonymisierungsnetzwerk TOR¹ und das NoScript² Plugin für den Firefox Browser. Mit Hilfe des Tor Projektes kann effektiv das Hinterlassen von Spuren im Internet verringert werden. Das NoScript Plugin verhindert das Ausführen von JavaScript, Java und anderen Plugins auf besuchten Seiten. Der Nutzer muss bei jeder Webseite explizit die Ausführung von Skripten erlauben.

Neben den in Browsern verfügbaren Schutzmaßnahmen führt Eckersley noch Funktionalitäten in Browsern auf, die große Teile von Systeminformationen freilegen und damit das Erzeugen von Fingerabdrücken stark vereinfachen. Diese Funktionalitäten lassen sich zur Zeit nicht durch den Nutzer beeinflussen oder deaktivieren. Dazu gehören z. B. das Auflisten von Systemschriftarten (mit Ausnahme der kompletten Deaktivierung des Flash Plugins) oder Plugins samt Versionsinformationen. Die hohe Aussagekraft von Plugins ist

¹<https://www.torproject.org/>

²<https://addons.mozilla.org/de/firefox/addon/noscript/>

mit einer durchschnittlichen Entropie von 15,4 bits und von Schriftarten mit einer durchschnittlichen Entropie von 13,9 bits (siehe Tabelle 2) im Gegensatz zu den restlichen Charakteristika deutlich zu erkennen. Der eigentliche Nutzen der API Methoden zum Auflisten von installierten Plugins samt Versionshinweisen liegt in der Vereinfachung der Software-Entwicklung. Die Auflistungen sind deshalb lediglich Software-Entwicklern von Vorteil und sind für den Endnutzer von keinerlei Interesse. Laut Peter Eckersley sollten System-Schriftarten und Plugins nur über Zustimmung des Nutzers aufgelistet werden.

Ein weiteres Problem existiert mit den detaillierten Versionsangaben in User Agent Strings. Hier werden oft Mikroversionen angegeben (Java 1.6.0_17 statt Java 1.6). Diese detaillierten Versionsangaben führen zu hohen Entropien und damit zu besseren Fingerabdrücken. Die Mikroversionen werden von Softwareentwicklern für Fehleranalysen genutzt. Hier sollte der Schwerpunkt weg von der Annehmlichkeit der Entwickler in Richtung Schutz der Privatsphäre des Nutzers verlagert werden, indem man lediglich Hauptversionen im User Agent String anführt.

Eine weitere, die Entropie steigernde Gegebenheit ist, dass Plugin- und Schriftart-Auflistungen unsortiert zurückgegeben werden (Peter Eckersley hat in seinen Experimenten nicht den CSS Font Detector (Kapitel 2.3.1) verwendet, bei dem die Reihenfolge der zu testenden Schriftarten selbst definiert wird). Dadurch können Nutzer mit gleicher Konfiguration unterschieden und damit auch identifiziert werden. Eine einfache Sortierung der Auflistungen würde dieses Problem beseitigen [6, S. 13–15].

4. ZUSAMMENFASSUNG

Peter Eckersley zeigt exemplarisch, wie man Internet-Nutzer anhand der Informationen ihres Web-Browsers identifizieren kann. Der entwickelte Web-Browser-FingerprintAlgorithmus konnte einen Großteil der Web-Browser-Konfigurationen im Testdatensatz eindeutig identifizieren. Sehr gut ist auch, dass nicht nur die einzelnen Fingerabdrücke an sich, sondern auch das Reidentifizieren von veränderten Fingerabdrücken behandelt wird.

Die Wahl der Messvariablen im Algorithmus (siehe Tabelle 1) brachte ausreichend hohe Entropiewerte um Browser-Konfigurationen im Testdatensatz in über 83,6% der Fälle eindeutig zu identifizieren. Die Kennzahlen lassen sich jedoch aus den in Kapitel 3.2.2 beschriebenen Gründen nicht auf die globale Ebene extrapolieren. Eine Möglichkeit die Extrapolation der Daten zu ermöglichen, wäre die Messvariablen um weitere Informationen wie Active-X, Microsoft Silverlight, Adobe Flex, oder JavaFX Komponenten zu erweitern und damit die Entropiewerte nochmals zu erhöhen. Eine weitere Möglichkeit wäre die Sammlung von Fingerabdrücken von weniger voreingenommenen, technisch unversierteren Internet-Nutzern.

Eckersley zeigt neben der Problemstellung auch unterschiedliche und mehr oder weniger effektive Möglichkeiten auf, sich gegen das Erheben von digitalen Fingerabdrücken zu schützen. Er zeigt darüber hinaus auch auf, dass es in diesem Bereich noch einigen Spielraum für Verbesserungen gibt.

Abschließend ist zu sagen, dass digitalen Fingerabdrücken in Zukunft im Bereich Privatsphäre die gleiche Relevanz zugeschrieben werden muss, wie es bereits bei Cookies und IP-Adressen der Fall ist.

5. LITERATUR

- [1] What the internet knows about you. <http://whattheinternetknowsaboutyou.com>.
- [2] BVerfGE, Urteil des Ersten Senats, 65, 1. <http://sorminiserv.unibe.ch:8080/tools/ainfo.exe?Command=ShowPrintText&Name=bv065001>, Dezember 1983.
- [3] Website-Speichereinstellungen. http://www.macromedia.com/support/documentation/de/flashplayer/help/settings_manager07.html, März 2012.
- [4] J. Appelbaum. The Tor Project. <https://www.torproject.org/about/overview.html.en#whyweneedtor>.
- [5] P. Eckersley. A Primer on Information Theory and Privacy. <https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy>, Januar 2010.
- [6] P. Eckersley. How unique is your web browser? pages 1–18, LNCS 6205/2010, 2010. Proc. 10th Privacy Enhancing Technologies Symposium (PETS2010).
- [7] R. C. H. Krawczyk, M. Bellare. HMAC: Keyed-Hashing for Message Authentication. Februar 1997.
- [8] Jan Lukás, Jessica Fridrich, and Miroslav Goljan. Digital Camera Identification from Sensor Pattern Noise. *IEEE Transactions on Information Forensics and Security* 1, 2, 2006.
- [9] R. Johannesson. *Informationstheorie – Grundlagen der (Tele-)Kommunikation*. Addison-Wesley Publishing Company, 1992.
- [10] B. Krishnamurthy. Generating a Privacy Footprint on the Internet. *Proc. 6th ACM SIGCOMM Conf. on Internet Measurement (IMC'06)*, pages 1–6. ACM, Oktober 2006.
- [11] J. R. Mayer. Any person... a pamphleteer - internet anonymity in the age of web 2.0. *Undergraduate Senior Thesis, Princeton University*, 2009.
- [12] Ordway Hilton. The Complexities of Identifying the Modern Typewriter. *Journal of Forensic Sciences* 17, 2, 1972.
- [13] Parlamentarischer Rat. *Grundgesetz für die Bundesrepublik Deutschland*. Bonn, 1949. Stand: September 2010.
- [14] L. Patel. JavaScript/CSS Font Detector. <http://www.lalit.org/lab/javascript-css-font-detect/>, März 2007.
- [15] S. Schoen. What Information is Personally Identifiable? <https://www.eff.org/deeplinks/2009/09/what-information-personally-identifiable>, September 2009.
- [16] Stiftung Weltbevölkerung. Die Weltbevölkerungsuhr. <http://www.weltbevoelkerung.de/oberes-menue/publikationen-downloads/zu-unseren-themen/weltbevoelkerungsuhr.html>, 2012.
- [17] A. B. Tadayoshi Kohno and K. Claffy. Remote Physical Device Fingerprinting. *Dependable and Secure Computing, IEEE Transactions on*, Juni 2005.

TLS Solutions for Wireless Sensor Networks

Sebastian Wöhrl
Betreuer: Corinna Schmitt
Seminar Future Internet SS2012
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: sebastian.woehrl@mytum.de

ABSTRACT

Wireless Sensor Networks are an interesting research topic with many possible real world applications. The increasing number of sensor networks and their widespread deployment throughout the world makes them a more and more interesting target for attackers. With the Internet Protocol (IPv6) becoming the standard for communication among these networks it is possible to use established standards for this task. The current standard for secure communication in the internet is Transport Layer Security (TLS) which is used worldwide and easy to implement. This paper discusses several solutions for and enhancements of TLS to use it in IP-based Wireless Sensor Networks to leverage the power of TLS while still keeping in mind the limited resources of sensor networks.

Keywords

TLS, SSL, IPv6, Security, Wireless Sensor Network

1. INTRODUCTION

A Wireless Sensor Network (WSN) is a network of small autonomous sensor nodes (usually with an embedded processor and therefore with low computing power) which are communicating using wireless connections. The application area of such sensor networks is usually monitoring external conditions ranging from physical to environmental values[11]. A relatively new field of application is using the sensors to monitor medical readings of human patients[12].

With the internet and its primary protocol (IP) being the worldwide standard for data communication it is only logical to also build WSNs based on this standard. This became possible with the upcoming of IPv6 and its greatly enlarged address pool making it possible to create IP-based networks for sensor nodes with each sensor having its own IP-Address. This makes it possible to integrate these sensor networks into the internet and using established standards.

Due to the nodes communicating wirelessly special considerations have to be made for securing these links. With TLS/SSL being the standard protocol for encrypting and authenticating connections in IP networks it is only logical to also use it for IP-based WSNs. Of course with the sensor nodes being very limited in terms of energy and computing power and considering the special circumstances of the deployment of such networks just using plain TLS will not be very efficient.

This paper will describe and discuss several approaches to this topic and is organized as follows. Section 2 describes "original" TLS based on the official RFC. Sections 3 to 5 will discuss possible approaches for using TLS with more than two entities[1], for using identity-based cryptography with TLS[2] and finally Tiny-3-TLS[3]. Finally, the solutions are compared and summarized in Section 6.

2. TLS FOLLOWING RFC 5246

The current version of TLS, 1.2 (or SSL 3.1), is defined in RFC 5246¹ [4]. TLS consists of a series of protocols. The basic protocol is the so called "TLS Record Protocol" which in the ISO/OSI-Layer-Model is put directly above the Transport Layer (c.p. Figure 1).

| Handshake Protocol | Change Cipher Spec Protocol | Alert Protocol | Application Data Protocol |
|---------------------|-----------------------------|----------------|---------------------------|
| SSL Record Protocol | | | |
| TCP | | | |
| IPv4 / IPv6 | | | |

Figure 1: TLS in the ISO/OSI-Model [2]

Above the record protocol are the Handshake Protocol, the Change Cipher Spec Protocol (used to negotiate key and algorithm changes), the Alert Protocol (to signal problems) and the Application Data Protocol (used to transport the user data). The most interesting is the Handshake Protocol because it is used to establish the TLS session and to negotiate the session parameters like encryption keys and algorithms. As such it provides the most starting points to optimize the TLS protocol.

Following is a description of a normal TLS Handshake between a Client and a Server using X.509 Certificates as shown in Figure 2.

The connection starts with the client sending a ClientHello (1) containing a random number (*rnc*) and his supported algorithms (the Cipher Suites). The random number *rnc* is generated and transmitted to protect against replay attacks. The Server responds with a ServerHello (2) also containing

¹For a more lightweight reading you can refer to [6]

a random number (rns) and the Cipher Suite the server has chosen from the offered ones. The server will also send its certificate (3) and will optionally request a certificate from the client (4) before ending the Hello phase (5). The client checks the transmitted server certificate and transmits its own certificate (6).

With certificate checks done the client generates a new random number (called the pre-master-secret, **PMS**, 7) and sends it to the server (8) encrypted with the server's public key which is contained in the server certificate. Using this pre-master-secret both calculate the master-secret (10, **MS**) using a pseudo-random-function specified in the cipher suite. Earlier versions of TLS used the MD5 and SHA-1 hash functions.

Also the client proves it really is in control of the private key for which he has presented a certificate by sending a CertificateVerify message (9) which is a signature over all previously exchanged messages using the client's private key.

Once the master-secret is calculated both parties send a ChangeCipherSpec message (11, 13) and a Finished message (12, 14) and from there on encrypt all messages using this master secret. The Finished messages (11, 13) are already encrypted and contain a hash and a Message Authentication Code (MAC) over all previously exchanged messages. If the other party cannot decrypt the Finished message or the hash or MAC verification fails then the entire handshake is considered a failure and the connection closed.

Encryption of messages during data transport is usually done using AES. In Wireless Sensor Networks AES-128 is used most often as it provides the best trade-off between security and computational effort. As AES is already quite efficient most TLS optimizations for Wireless Sensor Networks focus on the TLS Handshake.

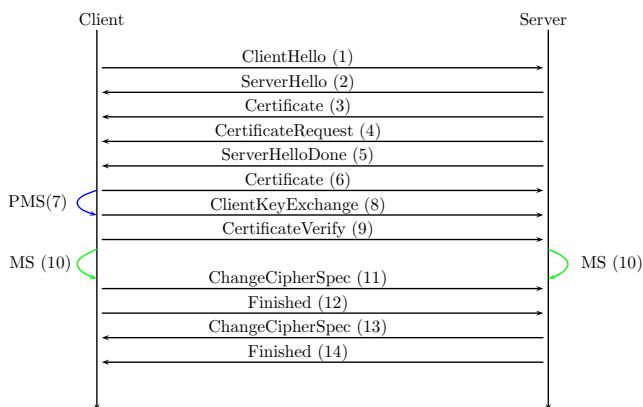


Figure 2: Time-schematic TLS Handshake[6]

3. TLS FOR MORE THAN TWO ENTITIES

Mohamad Badra describes in [1] a way to expand the original TLS protocol to use it for more than two entities and therefore to no longer be bound by the client/server-architecture of the original design.

3.1 Why more than two?

Most of the communication in the internet is done between two entities, either in a client/server-mode or as P2P (peer-to-peer). As client/server is the standard, TLS was designed for that use case. Wireless Sensor Networks are often built with several layers. The sensor nodes communicate with each other and over possibly several hops with a router node. These in turn communicate with a gateway node which connects the WSN to the internet and allows clients in the internet - e.g. a lab computer controlling the nodes and checking the sensor readings - to communicate with the sensor nodes (for details see [9] and [11]).

The gateway node can act as a caching proxy to reduce the number of messages to the sensor nodes if many clients want to pull the same readings or to log the readings for further study. But with an encrypted connection between the sensor and the client the proxy has no way to read the content of the messages. Therefore a way must be found to allow a secure connection between three entities. This can be generalized to N entities, one client, one server and $N-2$ intermediaries.

3.2 Expanding TLS

A naive approach at solving this problem would be to have each of the entities maintain $N-1$ separate connections and send each message to all the other nodes. It is quite clear that this is not the best approach. Mohamad Badra describes "an enhanced way to establish a TLS session between N entities. To simplify [he] describe[s] [his] solution with $N=3$ "[1]. The following section is based on his paper (Section IV).

The basic extension is that all intermediate entities are told the pre-master-secret so they can compute the master-secret and therefore decrypt all the messages sent. Following is a detailed description of a handshake for $N = 3$ entities, Client (C), Intermediary (E) and Server (S), also shown in Figure 3:

The client sends a ClientHello to the Intermediary (containing a random number rn_C and suggested cipher suites), the Intermediary selects one or more of the cipher suites and sends an own ClientHello to the Server containing the same random number rn_C . The Server generates its own random number rn_S and sends a ServerHello to the Intermediary. Then these two do a normal TLS Handshake Hello as described in section 2 (including sending and validating the server certificate). Once this is done the Intermediary passes on the ServerHello and the certificate from the Server but also includes his own certificate. The client verifies all certificates and sends his own client certificate to the Intermediary which relays it to the Server. After this the client computes a normal pre-master-secret but instead of just sending it to the server (encrypted with the server's public key) he also sends it encrypted to the Intermediary (whose public key he has from the transmitted intermediary certificate). As all involved parties now have the pre-master-secret they can each calculate the master secret and do a ChangeCipherSpec. Following that step client and server have a TLS-secured encrypted connection which all intermediaries can legally eavesdrop on.

The extension to the original TLS Handshake protocol are

several additional messages used to exchange information with the intermediaries, client and server still do a normal handshake.

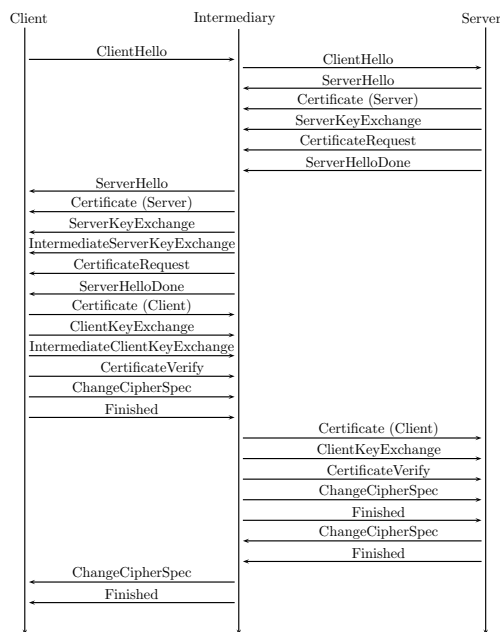


Figure 3: Time-schematic TLS Handshake for more than two entities [1]

3.3 Discussion

The described approach is superior to the mentioned naive approach in all respects. Doing the intermediary-handshake is cheaper than doing $N-1$ separate handshakes but still more expensive than just doing one client/server-handshake. Also during the session each message only needs to be encrypted with one key (the common master secret) regardless of the number of N . This makes it interesting for Wireless Sensor Networks which often need to communicate with several entities in the loop. But doing all these crypto operations for the establishment of the connection is not ideal for sensor nodes. In terms of energy and computing needs RSA (which is normally used in TLS with certificates) is quite expensive and therefore can be too costly for the embedded processors used in sensor nodes. This leads to the next section which describes solutions for cheapening TLS.

4. TLS WITH IDENTITY-BASED CRYPTOGRAPHY FOR IP-BASED WSNS

In [2] the authors describe two ways to do a TLS Handshake without using RSA and X.509 Certificates: One is done using Identity-based Cryptography and Elliptic Curve Diffie-Hellman, the other is done using Elliptic Curves and Bilinear Pairing.

4.1 Some Explanations

First some introductions and explanations of the techniques and algorithms used in these approaches.

4.1.1 Identity-Based Cryptography

Identity-based Cryptography (IBC) avoids using certificates which reduces the number of messages to be exchanged during a key exchange as certificates are usually quite large. Instead in IBC there is a Private Key Generator (PKG) which is the replacement for the Certification Authority (CA) used in Certificate-based cryptography. It generates a secret key for each node based on its unique identity (which for ip-based sensor nodes is usually the IPv6 address). The nodes must be preloaded with this secret key prior to their deployment. Due to this role the PKG is a trusted party and must not be compromised.

This even provides security against IP address spoofing. An attacker could try to use the address of a legit node. But since he does not have the private key corresponding to the IP address and - without compromising the PKG and getting its secure parameters - has no way of generating it.

4.1.2 Bilinear Pairing

In short bilinear Pairing allows two parties to agree on a shared key (e.g. as a session key) without exchanging any messages. In more mathematical terms:

"Let G_1 denote a cyclic additive group of some large prime order q and G_2 a cyclic multiplicative group of the same order. A pairing is a map $e : G_1 \times G_1 \rightarrow G_2$ and has an important property that is bilinearity; if $P, Q, R \in G_1$ and $a \in \mathbb{Z}_q^* [\dots]$ $e(aP, Q) = e(P, aQ) = e(P, Q)^a$." [2]

4.1.3 Elliptic Curves

Elliptic Curves (EC) [10] are an algebraic concept which are plane curves described by the equation

$$y^2 = x^3 + ax + b \quad (1)$$

and the points on that curve.

Suffice it to say that as long as the discrete logarithm problem is still expensive and difficult to solve, elliptic curve cryptography can be considered secure. Their main advantage over RSA is that the same level of security can be achieved with much shorter keys (usually around 128 bit EC is considered the same as 1024 bit RSA). This point makes them interesting to use for limited devices such as sensor nodes or smartcards as shorter keys imply cheaper operations.

4.1.4 Elliptic Curve Diffie-Hellman

The Diffie-Hellman key exchange is a protocol that allows two parties to agree on a shared key over an unsecure communications channel - this works without sending the key itself over the wire. It is defined in RFC 2631 [7]. Elliptic Curve Diffie-Hellman (ECDH) is a variation of the protocol that uses elliptic curve public/private-key-pairs. The algorithm was introduced in [5], a longer description can be found there. Basically the exchange between Alice (A) and Bob (B) goes as following:

Both must have the same elliptic curve (or more concretely a generator P which is a point on the curve). Also each one needs a public/private-key pair (denoted as d_A and d_B for the private part and Q_A and Q_B for the public part). d is a randomly selected value and Q is calculated as $Q = d * P$. After Alice and Bob transmit their respective public keys

(Q) to each other they can both calculate the shared key x as $x_A = d_A * Q_B$ respectively $x_B = d_B * Q_A$. It holds $x = x_A = x_B$ because $d_A * Q_B = d_A * d_B * P = d_B * d_A * P = d_B * Q_A$. Normally some form of hash function is used on x to get the shared key.

4.2 TLS Handshake with IBC and ECDH

With this approach of using TLS with IBC and ECDH [2] the authors have tried to optimize the TLS handshake protocol for use with low-power devices such as sensor nodes. Prior to their deployment inside a network all the nodes need to be equipped with a private key and an identity (IPv6 address). As mentioned above a PKG is needed as a trusted party to generate private keys based on the IPv6 address, it also initializes some parameters for the elliptic curves (namely the generator point P of the elliptic curve).

The start of the TLS Handshake is the same as the original handshake with transmission of the ClientHello and the ServerHello (c.p. Figure 4). Also in accordance to the original specification both nodes generate random keys (let them be rn_C for the client and rn_S for the server). But instead of sending over his certificate the server calculates $rn_S * P$, signs it using his private key and sends it to the client with a ServerKeyExchange message. The client does likewise, computes $rn_C * P$, signs it and sends it to the server using a ClientKeyExchange message. The parties then exchange the normal ChangeCipherSpec and Finished messages to end the handshake. The client now knows $rn_S * P$ and rn_C and therefore can calculate $rn_C * P * rn_S$ which is used as pre-master-secret. Likewise for the server who knows $rn_C * P$ and rn_S and can also calculate the pre-master-secret. Then they both can derive the master secret using the pre-master-secret in the normal way.

The step of exchanging certificates (which is part of the original TLS specification) can be omitted because in this incarnation the IPv6 addresses act as certificates and are already known to the communication partner due to the communication being ip-based. This saves two rather costly messages. The other point to note is that in the original TLS specification the pre-master-secret (a random number calculated by the client) is sent to the server encrypted using the server's public key. With this implementation (using IBC and ECDH) this is not necessary, the parts of the pre-master-secret ($rn_C * P$ and $rn_S * P$) are sent in plaintext because as mentioned above it is for an attacker not feasible to calculate the parts P and rn_C or rn_S of the value sent over the wire.

4.3 TLS Handshake with ECC and bilinear pairing

The authors also propose a second adaption of the TLS handshake using ECC and bilinear pairing which further reduces the number of messages sent.

Prior to deployment of the nodes there is again a PKG needed to choose/compute some parameters for the bilinear pairing. These are a random number $S \in \mathbb{Z}_q^*$, the groups G_1 and G_2 of the same prime order q , a point $P \in G_1$, the bilinear map e and a hash function H returning points on an elliptic curve for identities (IPv6 addresses, named ID).

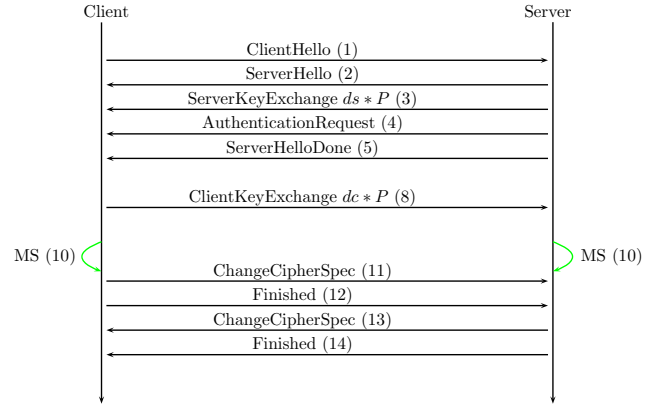


Figure 4: Time-schematic TLS Handshake with IBC and ECDH [2]

For each node j the PKG computes $Q_j = S \times H(ID_j)$ which is the private key.

The TLS handshake again starts with the exchange of ClientHello and ServerHello (c.p. Figure 5). After ending the hello phase with a ServerHelloDone both client and server can calculate the pre-master-secret as follows: Let ID_C be the identity of the client (=IPv6 address) and ID_S the identity of the server. S is the random number chosen by the PKG but not directly known by neither the server nor the client. The client computes the pre-master-secret as $e(Q_C, H(ID_S))$ where Q_C is the private key of the client. The server computes it as $e(H(ID_C), Q_S)$. These two are identical because $Q_C = S \times H(ID_C)$ and $Q_S = S \times H(ID_S)$, so the client computes $e(S \times H(ID_C), H(ID_S))$ and the server computes $e(H(ID_C), S \times H(ID_S))$ which are equal according to the definition of bilinear pairing above with S as a , $H(ID_C)$ as P and $H(ID_S)$ as Q .

Using this common pre-master-secret both the client and the server can calculate the master-secret, and end the handshake after doing a ChangeCipherSpec.

In comparison to the above described solution with IBC and ECDH another two messages can be saved (namely the ServerKeyExchange and the ClientKeyExchange).

But the security of the whole process relies on the S being kept secret so that only the PKG knows its value. If the value would become public a eavesdropper could very easily compute the pre-master-secret and therefore also the master-secret because the identities (ID_C and ID_S) are public.

4.4 Usefulness to WSNs

As already mentioned the sensor nodes of Wireless Sensor Networks usually use embedded processors with a low computing and power capacity. This makes RSA an undesirable element of the TLS Handshake as it is very expensive to compute and the X.509 certificates accompanying the handshake are relatively big and therefore expensive to transmit.

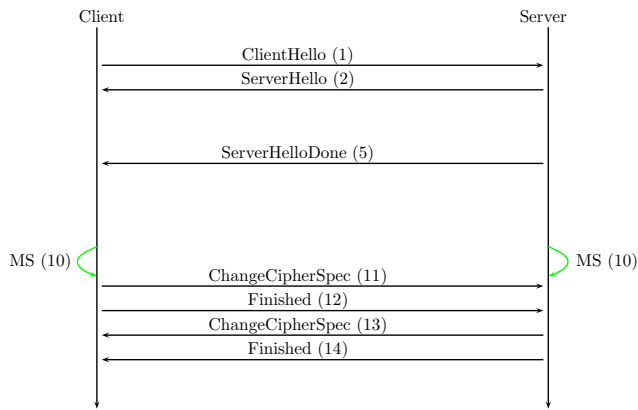


Figure 5: Time-schematic TLS Handshake with ECC and Bilinear pairing [2]

Both approaches deal with this by throwing out RSA and certificates and introducing Elliptic Curves and Identity-based Cryptography as their replacements. Thus reducing the needed computing power and the number of messages which have to be sent (which for wireless connections is extremely power-consuming) for a successful TLS Handshake.

TLS with IBC and ECDH still stays close to the original standard by still doing a key exchange (computing and transmitting random encrypted numbers) while TLS with ECC and bilinear pairing fully dispenses with exchanging keys. So both approaches make beneficial changes to the original TLS standard for use with Wireless Sensor Networks.

But for all the advantages there is also a disadvantage which comes with the design of the Private Key Generator. It is a critical part of the infrastructure as it is responsible for generating the private keys for all nodes. If the PKG were to be compromised an attacker had all the information he had for eavesdropping on the secured connections between the nodes. This is not unlike the Certification Authority used in certificate-based cryptography which also needs to be maintained and kept secure. Which is not so easy as hacks in recent times have shown [8].

5. TINY-3-TLS

As Wireless Sensor Networks usually use some form of gateway node anyway to connect to the outside world (i.e. the Internet) it would make sense to use this gateway node as a helper for establishing secure communication between a sensor node and an outside client seeing as such a gateway node normally has stronger hardware and can therefore shoulder the complex computations for cryptography more easily. One such approach is Tiny-3-TLS whose "goal [...] is to provide an end-to-end secure communication between a remote device and a wireless sensor network"[3].

5.1 Basics

The paper differentiates between a partially trusted gateway, which means the gateway helps in establishing the connec-

tion but should not be able to eavesdrop on the end-to-end secure channel, or a fully-trusted gateway which will possess the shared secret key and therefore be able to listen in on the secure channel.

As a possible use case the authors mention the MAGNET.Care-Project[12]. The scenario is that a patient of a hospital carries medical sensors organized as a wireless sensor network and a physician at the hospital wants to connect to the sensors to get current readings using a security gateway. From a patients viewpoint the security gateway at the hospital is not fully trusted and should therefore not be able to read the transferred medical data. Whereas if the patient is at home and his home router acts as security gateway it is fully trusted and allowed to read the sensitive data.

As already previously mentioned traditional asymmetric cryptography like RSA is relatively expensive in terms of computational needs so Tiny-3-TLS again substitutes RSA with Elliptic Curve Cryptography (ECC). As a means of agreeing on a shared secret key the protocol uses Elliptic Curve Diffie-Hellman (ECDH) which was already explained in an earlier section of this paper. The ECDH public values mentioned below refer to the public key part of the ECC public/private-key pair, above denoted as Q_x .

One basic assumption is made for both approaches: Between the sensor node and the security gateway there is a shared secret key, denoted as K .

5.2 Partially Trusted Gateway

In this scenario the gateway (GW) assists in establishing the secure connection between a remote terminal acting as a client and a sensor node acting as a server but does not possess the TLS session key at the end.

The TLS handshake (as shown in Figure 6) starts with a ClientHello containing the usual CipherSuite offers, the client identity (ID_c) and a nonce (N_c) from the client to the GW which encrypts the entire packet using the shared symmetric key K and sends it on to the server. In response the server sends a ServerHello message (encrypted with K) to the GW additionally containing the server identity (ID_s), a nonce (N_s) and its ECDH public values. The GW does not pass the message on to the client but instead composes his own ServerHello which does not contain the servers ECDH public values but instead contains the GW certificate and a request for a client certificate. To this the client responds with his own certificate, ECDH public values and a second nonce (N_g), called gateway authentication nonce. The entire message is encrypted asymmetrically using the GWs public key found in the certificate. Upon receipt the GW proves its ownership of the private key mentioned in the GW certificate by sending the gateway authentication nonce back to the client, it also includes the ECDH public values of the server which the GW removed from the first ServerHello (all encrypted with the clients public key). Also the GW transmits the ECDH public values received from the client to the server (again encrypted with K).

As both the server and the client now have the ECDH public values of the other partner they can now calculate the pre-

master-secret according to the ECDH algorithm and using the exchanged nonces (N_c and N_s) and identities (ID_c and ID_s) calculate the master key. With this both can encrypt the Finished messages and start using their secure end-to-end channel.

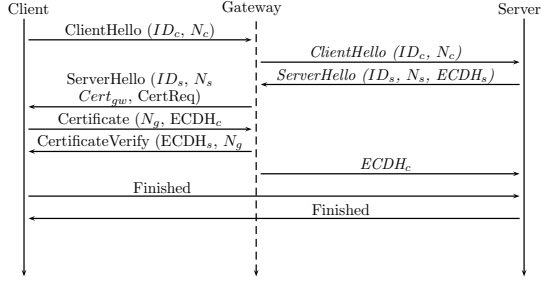


Figure 6: Time-schematic Tiny-3-TLS with partially trusted GW [3]

5.3 Fully Trusted Gateway

5.3.1 TLS Handshake

The procedure for doing the TLS handshake using the fully-trusted gateway (as shown in Figure 7) is at the beginning very similar to the procedure for the partially-trusted gateway. The client sends its ClientHello, the GW passes it on encrypted with K to the server which responds with a ServerHello but this time without ECDH public values. The GW again passes the message along but includes his certificate and a certificate request. The client responds to that request by transmitting his own certificate to the GW. Then - as in standard TLS - it generates a pre-master-secret (usually just a random number), encrypts it with the public key of the GW and sends it on to the GW.

The GW generates a client-read-key and a client-write-key and sends them along with a random number encrypted (using K) to the server. Once the server confirmed it has received and decrypted the keys (by sending back the random number) the GW ends the TLS handshake with the client by sending a Finished message. With that the secure channel is established.

Communication between the client and the GW is done using the master secret derived from the pre-master-secret sent by the client. Between the GW and the server messages are encrypted using the client-read-key and the client-write-key generated by the GW.

5.3.2 Comparison to TLS for more than two entities

Tiny-3-TLS with a fully trusted gateway is very similar to the TLS enhancement for more than two entities from [1] explained in section 3. The main difference to the outcome is that in [2] at the end all entities have and use the same master secret whereas in Tiny-3-TLS a real TLS session is only established between the client and the GW which relays the messages to the server using a special key not known to the client. This means the GW has to do additional cryptography computations for reencrypting all messages passed around. But this can be an advantage. Client and GW can

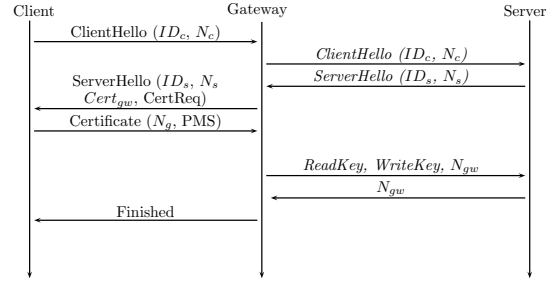


Figure 7: Time-schematic Tiny-3-TLS with fully trusted GW [3]

use strong and complicated encryption algorithms to secure the messages while travelling through the internet or another unsecure network. Whereas between server and GW a cheaper algorithm can be used which is more suited for the low-power processors used in sensor nodes.

6. SUMMARY

In this paper three different extensions/enhancements to the Transport Layer Security Protocol were described which all have their merits and faults (a tabular comparison is shown in Table 1). TLS for more than two entities is the ideal solution for interconnecting multiple entities with one shared secured channel. But it is not very useful for applications in Wireless Sensor Networks as it still uses the computationally heavy standard TLS asymmetric cryptography protocols such as RSA with X.509 certificates. But it provides an interesting basis for connections between multiple nodes which in sensor networks is more common than in the classic internet.

Tiny-3-TLS remedied that weakness by introducing elliptic curves and a gateway node for handling the intensive cryptography computations. It can be seen as an enhanced way for the approach in section 3 - specifically tailored for $N = 3$.

The introduction of Identity-based Cryptography is a different approach also making use of elliptic curve cryptography and other concepts to reduce the number of messages and computations that have to be done by the nodes but keeps the involved number of entities fixed at $N = 2$.

All three approaches enhance TLS for specific applications and should not be considered as a general improvement of Transport Layer Security.

7. REFERENCES

- [1] M. Badra: *Securing Communications between Multiple Entities Using a single TLS Session*, IEEE 2011
- [2] R. Mzid, M. Boujelben, H. Youssef, M. Abid: *Adapting TLS Handshake Protocol for Heterogenous IP-Based WSN using Identity Based Cryptography*, In Proceedings of the International Conference on Wireless and Ubiquitous Systems, 8-10 October 2010, Sousse, TUNISIA
- [3] S. Fouladgar, B. Mainaud, K. Masmoudi, H. Afifi: *Tiny 3-TLS: A Trust Delegation Protocol for Wireless*

Table 1: Advantages & Disadvantages of the different approaches

| Approach | Advantages | Disadvantages |
|----------------|---|---|
| TLS $N > 2$ | <ul style="list-style-type: none"> • little additional effort required • compatible to original TLS | <ul style="list-style-type: none"> • uses X.509 Certificates • uses RSA (computationally expensive) |
| IBC, EC, BP | <ul style="list-style-type: none"> • Elliptic Curves are better than RSA • Bilinear Pairing optimal for number of messages sent | <ul style="list-style-type: none"> • Private Key Generator is needed • Not conformant to TLS standard |
| Tiny-3-TLS | <ul style="list-style-type: none"> • Uses gateway node which is needed anyway • Less number of messages | <ul style="list-style-type: none"> • Gateway needs to be trusted • K must be securely shared |

Sensor Networks, ESAS 2006, LNCS 4357, pp. 32–42, 2006

- [4] T. Dierks, E. Rescorla: RFC 5246 - The Transport Layer Security (TLS) Protocol, Version 1.2, <http://tools.ietf.org/html/rfc5246>
- [5] L. Law, A. Menezes, M. Qu, J. Solinas, S. Vanstone: *An Efficient Protocol for Authenticated Key Agreement*, Technical Report CORR 98-05, Dept. of C&O, University of Waterloo, Canada, March 1998
- [6] S. A. Thomas: *SSL and TLS Essentials - Securing the Web*, Wiley Computer Publishing, USA 2000
- [7] E. Rescorla: RFC 2631 - Diffie-Hellman Key Agreement Method, June 1999, <http://tools.ietf.org/html/rfc2631>
- [8] *What You Need to Know About the DigiNotar Hack*, http://threatpost.com/en_us/blogs/what-you-need-know-about-diginotar-hack-090211
- [9] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci: *Wireless sensor networks: a survey*, Computer Networks 38 (2002) p. 393-422
- [10] C. Eckert: *IT-Sicherheit: Konzepte, Verfahren, Protokolle*, Oldenbourg Wissenschaftsverlag, München 2008
- [11] H. Karl, A. Willig: *Protocols and Architectures for Wireless Sensor Systems*, Wiley 2005
- [12] *IST-MAGNET*, <http://www.ist-magnet.org>

Watermarking in Sensor Data Sets

Sebastian Wiendl
Betreuer: Corinna Schmitt
Seminar Future Internet SS2012
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: wiendl@in.tum.de

ABSTRACT

This paper will give a brief introduction into "Electronic Watermarking" by describing its origins in the 1950s and its gain of importance in the past 10 to 20 years and then especially handle the topic of "Watermarking in Sensor Data Sets".

Electronic watermarking finds practical use in many business applications and is, next to cryptography, part of the security mechanisms against illegal redistribution and use of copyright-protected material. The traditional methods of electronic watermarking handle a great number of applications but struggle with unstructured data sets. Explaining a newly found method to cover that issue the second part of this paper goes into more detail by handling the technique of "Self-Identifying Sensor Data" as described in the paper [7]. Therefore this paper should be seen as a summary and comparison of the traditional methods with a new method in the field of electronic watermarking.

Keywords

Electronic watermarking, self-identifying sensor data

1. INTRODUCTION

Electronic watermarking is a method to implant a provenance mark into data. This is handle ownership and copyright issues. It can be tracked all the way back to the year 1954 in which the first patent had been filed. Since then a good amount of effort has been put into researching more sophisticated ways to watermark electronic products, nowadays digital products. Especially the past two decades stood out in the development because of the increasing need for electronic/digital watermarking. These days various techniques are being used by businesses and researchers are still working on newer and better methods. One of the newly found techniques is called "Self-Identifying Sensor Data" and covers a type of data that had previously not been able to be watermarked by the traditional techniques. In this paper, where past and future techniques are being compared, the reader will find some mathematical formulas as well as general overviews.

In section 2 the topic starts off with the definition of electronic watermarking and its development during the following years until now. After that the various types of commercial applications are shown. That section is then completed by a short summary before, in section 3, the approach of self-identifying sensor data is being discussed and shown in

great detail. This part follows the more general part with mathematical formulas and algorithms. Finally, section 4 concludes with a comparison of the traditional approaches for electronic watermarking and the new technique of self-identifying sensor data.

2. ORIGIN AND DEVELOPMENT OF ELECTRONIC WATERMARKING

2.1 Definition and origin

2.1.1 Definition

Typically watermarks are known in a non-digital manner. Every Euro bill has watermarks.



Figure 1: watermark in a 10 Euro bill

The idea is to clearly mark the bills with a visible mark that cannot (easily) be recreated or removed illegally and that everyone can recognize in order to verify the authenticity of the bill. Different applications exist. Figure 1 shows the watermark used on the Euro bills. Figure 2 shows a screen-shot of electronic watermarking. Google Maps includes small watermarks to their maps which can only be (barely) seen (circled in red) when zoomed in all the way [9].



Figure 2: Screenshot from Google Maps on highest zoom factor [9]

Electronic watermarking, or nowadays rather digital watermarking commits to the same principles. More precisely it can be explained as:

"Digital watermarking is the process by which identifying data is woven into media content such as images, printed materials, movies, music or TV programming, giving those objects a unique, digital identity that can be used for a variety of valuable applications." [1]

A good watermark should not add (much) additional data to the unmarked data. This is to avoid the consequences of being too expensive and uneconomically.

2.1.2 *Origin and progress*

The first patent to ever handle the topic of electronic watermarking was filed in the year 1954. Emil Hembrooke from the Muzac Corporation called his work "Identification of sound and like signals" [12] and wrote that his method would allow to identify the origin of a piece of music and can therefore "be likened to a watermark in paper" [4]. In the following 35 years several more patents have been filed. They were mostly revolving around the topic of music. The Lynch Carrier Systems Inc. designed a system to control telephony equipment and in a patent of the Musicast Inc. They used radio stations to distribute music to other businesses by adding a low frequency to the broadcast which would allow those businesses to remove advertisements [5]. All in all research was done and inventions were made but the big hype did not hit in yet. It was not until the 1990s that watermarking was put much more into focus. Reason for that was the Internet. With the extensive spread of the world wide web more and more people in the private sector gained access to the new media. The possibility of spreading information and data all over the world in almost no time gave opportunity to an exponential increase of illegal activities. Illegal sharing and selling of media with copyright protection led the music industry to immediate actions. Research had to find solutions and then, once again find more solutions when illegal distributors had found a new way to undermine their efforts. That race led to much progress in the topic of electronic watermarking.

The technology industry started groups like the Copy Protection Technical Working Group (CPTWG) (concerning digital video content) and the Strategic Digital Music Initiative (SDMI) (concerning digital music) to deal with those problems, too [4]. In general one can say that people have watermarking expected to be more sophisticated by now but it still provides several applications in commercial use.

2.2 **Commercial applications**

Watermarking can be used in a variety of ways to satisfy copyright and security aspects. The following applications should briefly explain and illustrate the use with several examples that are being used today.

2.2.1 *Transaction tracking*

In transaction tracking, also called fingerprinting, each copy work/device has an unique watermark embedded. That watermark is usually used to identify the origin of a copy. This can, for example, be used to track down the source of an illegally spread video. That way the origin of the distribution can be found and that person can be charged with the copyright issues. That technique is in use and widely spread. The DiVX corporation embedded that system into their DVD

players. So whenever a user creates many copies of a movie they can later be traced back to the initial player/user [8]. The problem with that technology is that collision attacks can easily undergo the system. The vulnerability lies in the big number of devices. If one person buys as few as about 20 copies of a single device an unmarked original can be created from the marked copies. That collusion attack would therefore allow to create watermark-free copies of DiVX DVD players if one had 20 original ones at hand. The effectiveness of fingerprinting for widely spread applications is therefore not perfect. Professional copiers will not be traced back but less talented users can safely be tracked back and held responsible. On the other hand transaction tracking can be well-applied for a small scale use. Pieces of work with a small availability - when it can be guaranteed that not many distributions will end up in the same wrong hands - will profit from the fingerprinting technology.[4] An example for a small scale application that led to success if the watermarking technique created by "Civolution". It "has been used successfully for identifying the source of illegal copies of the 2003 Academy Award screeners". [2]

2.2.2 *Proof of ownership*

The original idea when Muzak invented watermarking was to mark a piece of work in a way that can legally be used to prove ownership, even in a court trial. A major problem with that is that there is a wide variety of watermarks being applied. So how can a watermark be safely and for one hundred percent be connected with a single company/legal owner? The key to this is to not just embed a watermark independent of the original (unmarked) work, but create a cryptographic link between the original and the marked copy. S. Craver had that idea in 1996 and it is also technologically doable.[4] Actually, there is still a lot of research going on covering that topic. One research laboratory is "IBM Research". In one of their papers they describe the issue concerning invisible watermarks. [3]

2.2.3 *Copy Control*

Copy is the attempt to ensure that illegal pieces of work will not be created. Before the process of copying the device would check the specific piece of work for a copyright sign and then only copy if none was embedded. Taking that serious one would have to install watermark detectors in every recording and copying device. The device would then at first check the media for a watermark and only copy it if no watermark was found. Two main problems stand in the way of successfully guaranteeing a total copy control. At first it has to be ensured that truly everyone is able to detect the watermark and with that it would end up being only a weak security application. Some companies must still see an economical gain and apply copy control. Secondly, it makes sense that the whole idea only works perfectly if every company established a decoder in their devices. A couple of reasons strongly stand against that from the companies point of view: A detector is a piece of hardware (and software) that has to be added to every device. It will therefore add costs but no practical value to each device. Some people would then rather buy equipment that has no decoder in order to make their desired copies and this will lead to lower sales rates for the participating companies. The only way to ensure that every company joins the initiative would

be to force them by "a combination of laws and contractual obligations." [4]

2.2.4 Authentication

For authentication purposes a digital watermark functioning as a signature can visibly be added to, e.g. an image, as seen in Figure 3.



Figure 3: Digital image with a watermark

This type of visible watermark is not the only way to use authentication. There are many ways to invisibly embed a digital signature. One approach for this method can be found in [6]

2.2.5 Legacy system enhancement & database linking

Watermarking can not only be used as in a security type of application as mostly described before. Record companies can embed a digital watermark in their audio files that devices of cooperating companies can record, filter and decode. That would, for example, give them the song title, the artists name and the album name. More modern applications are not necessarily depending on that. The application "Shazam"[10], which can be downloaded on PC, MAC and as well on mobile devices using the common markets for apps, has its own way to do that. As described in a paper released by the "Shazam"-company [11] they have created their own database containing self-created acoustic fingerprints. The basic idea is that the "fingerprint hash is calculated using audio samples near a corresponding point in time, so that distant events do not affect the hash" [11]. Basically once the database is created the same algorithm used to create the hash can be applied to any recorded song and then the hash created by the algorithm will point to the song in the database. Now song title, artist and more information is available about the previously unknown piece of music - without the use of watermarking. But one or more decades in the past there were not as sophisticated devices as there are available these days. That is when patents were filed and some companies still used watermarking for the distribution of music. As already stated in section 2.1.2 the Musicast Inc. used radio stations to distribute music to other businesses by adding a low frequency to the broadcast which would allow those businesses to remove advertisements. [5]

2.3 Summary and Conclusion

As seen in the previous chapters there is a wide variety of applications for electronic watermarking. With the big hype in the 1990s various fields established themselves and there still is research and development of new methods going on. Some applications are slowly becoming redundant because of new inventions ("Shazam") and

others are less safe than originally desired. Although only weak security can be provided by the techniques many companies still see an economical advantage in using them and will continue to do so. In many fields electronic watermarking is used as an addition to cryptographic methods. That combination is very popular and provides good services. All in all there are no reliable suggestions to make about the future development and use of electronic watermarking. As I.J Cox and M.L. Miller say: "If the past is a prediction to the future, then it is clear that watermarking technology will continue to be used in businesses." [4]

3. SELF-IDENTIFYING SENSOR DATA

For many fields of research a lot of datasets are needed and are not necessarily gathered by the researchers themselves. There are big companies / institutes that generate those datasets and offer them for research. Of course, they still have the copyrights on it and want to ensure that their work is not used without references or worse. That issue is being handled in the following.

After the previously handled more general approach and overview this chapter will go more into the details of one specific application of electronic watermarking. As described in the paper "Self-Identifying Sensor Data" [7] the basic idea and fields of application of a new way to watermark sensor data sets will be laid out and then explain the mathematical background behind the approach. After that a summary of the papers evaluation and analysis will explain the quality trade offs of the approach. Finally the whole concept will be looked at in matters of deployment issues and future work.

3.1 General approach

Transmitted sensor data will always carry some noise with it. That means that, e.g. a thermometer might tell you the temperature precisely in a scale of 10^{-3} degrees Celsius but the sensor could still deliver you the temperature up to an accuracy of 10^{-8} degrees. Therefore the last few digits (least significant digits) are noise and not of any (big) importance. Because of that the authors of the paper [7] suggest to embed a provenance mark into the data by replacing the noise. Three important categories will be used to rate the technique:

Perceptibility

Perceptibility in this case means that the embedded watermark should not (significantly) change the data set. If an embedded provenance mark changed the data radically then it would not fit the criteria. Of course, including a mark will change data, but - as previously stated - including a mark only into the parts that contain noise will have no effect on the true data itself.

Robustness

Robustness is a characteristic that allows the data set to counter several transformations without being changed in a way that makes the data useless. This method is robust against: truncation and quantization of digits, random bit flips, sampling and reordering of data within the sets. All those transformations can, of course, only be withstood up to a certain point. e.g. random bit flips on each bit of the data would change it into a totally new dataset, etc.

Capacity

Capacity includes two factors:

One-bit watermarking

This allows to determine whether a data set is marked or not. The technique described in [7] uses two bits in each bit vector to save information. These *check bits* are the two most significant bits of all the least significant bits. The parameter check bit (*pc*) contains information about the parameters used for embedding the watermark and the mark check bit (*mc*) is a hash of the provenance mark and the significant bits. The more uncorrupted data points a set contains the more likely the two bits can be retrieved. Then one-bit checking can successfully be applied.

Blind watermarking

Blind watermarking allows one to extract an embedded watermark without having any knowledge of the specific mark. The approach to support that is as following:

In order to guarantee that an uncorrupted provenance mark can be found and extracted the provenance mark will be split in several pieces. Each bit vector will contain a piece (which specific one will depend on the significant bits of the vector). Overall each and every piece of the provenance mark will appear in several bit vectors at different bit sections (all within the least significant bits). This creates a lot of wanted redundancy. If one data point is corrupted (or even if all data points suffer from e.g. truncation) the provenance mark can still be extracted because the pieces appear multiple times at different locations. In order to put the provenance mark together after retrieval the least significant bits will be used to make an educated guess on the provenance mark. If the guess is wrong the next step would be to check various similar marks.

3.2 Detailed Description

3.2.1 Formal Problem Statement and Preconditions

Before diving straight into the details of the technique some mathematical foundation has to be stated:

A *data set* is a list of *vectors* where each vector is a bit vector representing a data point. *Transformations* will be seen as *functions* taking a list as an input and returning a list different from the input list. Here the two main types of functions will be encoding and retrieval functions. An *encoding* transforms a list to a coded list whereas a *retrieval function* will return the original list when getting the encoded list as an input. An *encoding-retrieval function pair* is robust when a retrieval of an encoded dataset, which had been corrupted, still returns the original dataset. Two final assumptions will help the process: The length of the provenance mark L_m will be known and additionally all data/bit vectors will represent positive integral data (integers) of a certain length.

The latter assumption leaves special cases to be handled separately:

All bit vectors that do not have the specified length can be changed by adding leading zeros. Other cases are:

Negative integers

Negative numbers can be transformed (and later also transformed back to their original representation) to signed integer bit vectors. Embedding a provenance mark will not have any effect on the sign-bit because only the least-significant bits will be changed.

Floating point numbers

Floating point numbers can be transformed to a decimal point representation and then multiplied by a power of ten (10^n) big enough that all bit vectors will finally only represent integers. After having embedded the watermark a multiplication with 10^{-n} will return the floating point representation. *Important:* If truncation has changed the length of a bit vector then it won't be 10^n ; similar exponents should be tried then (e.g. 10^{n+1} , 10^{n+2}). Sure enough, one could say that there is no need to transform the bit vectors. It is possible to simply use transformation matrices *during* the encoding and decoding process. In this paper/approach that will not be done. A pre-formatting of the data will keep the calculations during the application simpler and easier to understand!

Low-entropy datasets

Data sets with only a small amount of distinct numbers provide the problem that many provenance pieces would be the same (as they are created from the significant bits which are equal/very similar in such data sets). The solution is to introduce smaller provenance pieces. Some of the least-significant bits can then be seen as significant bits used to create the provenance piece. This will deliver the wanted diversification.

3.2.2 Possible Transformations

The corruption model consists of several transformations than can change the data.

Rounding occurs when bit vectors are rounded to the closest multiple of an integer n . Similar to that it might happen that some of the least significant bits are being cut off. That is called *Truncation*. Many datasets might contain a high number of single data points but there is not always a need for such an extensive amount of single data points. During *Deletion/Sampling* data points are being removed from the dataset or a simple subset is being used. In an unstructured set an order can often not be found and then *Reordering* might happen to the set. The new set is then a permutation of the original set. Finally, a phenomenon that computer scientists know very well can occur randomly. A *Bit flip*: changes a number of bits in a bit vector. Bit flips are much less likely to appear than the other transformations. This is a big advantage because e.g. rounding only affects the least significant bits where-else random bit flips can also affect the important data contained in the more significant bits!

3.3 Embedding, Checking and Retrieving of a Provenance Mark

This section goes into the (mathematical) details of embedding, checking and retrieving of a provenance mark. At first it will lay out how provenance pieces are created and how they will be embedded into a bit vector. The next step is to explain how the check bits derive from the provenance

pieces and the significant bits. After that the final big step is to retrieve a provenance mark and handle those cases in which corruption has occurred in the data set.

3.3.1 Embedding a provenance mark

Figure 4 shows how a bit vector will look like after the insertion of a provenance mark.

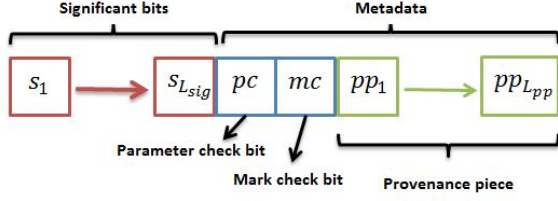


Figure 4: Bit vector with an embedded provenance piece [7]

The significant bits stay unchanged and only the least significant bits, also called metadata, will contain the various provenance pieces as well as the check bits.

Requirements and Terminology

- The number of the insignificant bits is called: L_{md}
- The number of the significant bits is called: L_{sig}
- The length of a data point therefore is: $L_{sig} + L_{md}$
- Having the two check bits, requires: $L_{md} \geq 3$

Provenance Pieces

If the length of a complete provenance mark is greater than $L_{md} - 2$ then it will be split up in smaller pieces. Each piece contains a part of the whole provenance mark and in total there will be N_{pp} pieces. For each data point a *hash* function will take N_{pp} and the significant bits s as input and return a value k . That means that the specific data point will have the k th provenance piece embedded into it.

```

provenance mark m:      0100110001001011
1st provenance piece pp0: 01001100
2nd provenance piece pp1: 11000100
3rd provenance piece pp2: 01001011
4th provenance piece pp3: 10110100
Lm = 16, Npp = 4, Lpp = 8

```

Figure 5: Provenance pieces example [7]

As you can see in Figure 5 the provenance pieces basically overlap. In the example from [7] the provenance mark technically could be split in only two different pieces that would cover all its data, already (pp_0 and pp_2). pp_1 and pp_3 cover redundant information. If, for example, rounding appears in the process then all pieces could, e.g. lose their last two bits. Without redundancy the data would be lost. But having pp_1 and pp_3 information that got lost in pp_0 will not be lost in pp_1 because the same information is placed into

the bits of higher significance. Same counts for pp_2 and pp_3 and, of course, vice versa. This will allow to still retrieve a correct provenance mark, even if corruption has annotated the dataset. Given a big enough number of data points a high redundancy will make this approach very robust.

Mathematically:

A provenance mark m has the length L_m and m_j stands for the j th bit of m . For each provenance mark pp^k the i th bit pp_i^k is calculated like this:

$$pp_i^k = m_j \text{ with } j = (k \frac{L_m}{N_{pp}} + i) \bmod L_m$$

Conditions: In order to guarantee that each bit of the mark will appear as often as every other one in all the provenance pieces L_{pp} has to divide $N_{pp} \times L_m$.

$N_{pp} \leq L_m$ to ensure that no two provenance pieces are equal.

Check bits

The check bits both are calculated by creating a hash of the significant bits s with either N_{pp} (for the parameter check bit pc) or with the provenance mark m (for the mark check bit mc).

Mathematically:

$$\begin{aligned}
 pc &= \text{hash}(2, s @ N_{pp}) \quad (1) \\
 mc &= \text{hash}(2, s @ m) \quad (2)
 \end{aligned}$$

3.3.2 Checking a provenance mark

One-bit watermarking is used to figure out if a provenance mark m' is actually the provenance mark m that originally had been embedded. For all normal cases this would not be needed but since transformations can corrupt a dataset and change it into an annotated dataset we need one-bit watermarking. This will allow us to still use the corrupted dataset.

Retrieving L_{sig} and N_{pp}

Remember:

- L_{sig} is the number of significant bits in a data point
- N_{pp} is the number of distinct provenance pieces

For the retrieval process values for L_{sig} and N_{pp} will be guessed. Then using that the equation (1) from above is being applied again with the respective values of the current (corrupted) data point d .

$$\text{hash}(2, d_0 \dots d_{L_{sig}-1} @ N_{pp}) = d_{L_{sig}}$$

If L_{sig} and N_{pp} have been guessed correctly and the data point was uncorrupted this equation will work and then $d_{L_{sig}}$ is exactly the parameter check bit pc . In all the other cases there is a probability of about $\frac{1}{2}$ that the data point is *pc-consistent*. This is because of the specific hash function used to calculate pc .

Definition:

The *pc-consistency score* will be "the proportion of data points" in a dataset "that are pc-consistent for guesses L_{sig} and N_{pp} " [7]. Correct guesses and uncorrupted first $L_{sig} + 1$ bits will lead to a score of 1, where-else a score of about $\frac{1}{2}$ can be expected. An incorrect guess when having a score of

1 only appears with a probability of 2^{-n} (with n being the amount of data points in the set). Having a finite amount of data points with finite length in the datasets there are limited guesses. Calculating with different parameters L_{sig} and N_{pp} will allow to finally pick the combination of L_{sig} and N_{pp} which led to the highest *pc-consistency score*. Those will be used for a check of the provenance mark.

Checking

Similarly to the *pc-consistency (score)* a *mc-consistency (score)* can be calculated. This equation is derived from equation (2) (Check bits) from above:

$$\text{hash}(2, d_0 \dots d_{L_{sig}-1} @ m) = d_{L_{sig}+1}$$

Also similarly a *mc-consistency* is defined as the proportion of datapoints in a dataset that are *mc-consistent* for L_{sig} and m . If L_{sig} and m are correct and the first $L_{sig}+2$ bits are uncorrupted the score will be 1, otherwise $\sim \frac{1}{2}$.

3.3.3 Retrieving a provenance mark

Retrieving a provenance mark from a corrupted dataset (*Blind Watermarking*) is a process that includes several steps. This is because each bit vector representation of a data point only contains a specific part of the whole provenance mark. Additionally, each piece might contain corrupted bits. For the retrieval of the provenance mark each bit vector will be split up in its specific pieces s , pc , mc , pp , and k , which tells us which provenance piece had been implanted in the current bit vector. That can easily be done using L_{sig} and N_{pp} . Even if some marks are corrupted that is no problem because each mark appears several times and also at several places (different bits). Now, in order to find the correct mark guesses have to be made and rated. For that a "suggest-function" [7] is being used:

$$\text{suggest}(d, i) = \begin{cases} (pp_j, j) & \text{if } j + k \bmod L_m = i \\ & \text{and } 0 \leq j < |pp| \\ * & \text{otherwise} \end{cases}$$

Each provenance piece d contains parts of the whole provenance mark. The suggest-function $\text{suggest}(d, i)$ tells whether the i th bit of the whole provenance mark is contained in d . Additionally it returns a number representing the confidence of the result (higher numbers represent a lower confidence). This is because a smaller bit pp_j (j th bit of the provenance piece d) has higher significance and is less likely to be corrupted. *Suggest* will return $*$ if d does not contain i or the bit pp_j with the corresponding confidence.

Example:

provenance mark $m = 0011$
 provenance piece $pp = 01$
 Then $\text{suggest}(pp, 0) = *$
 Then $\text{suggest}(pp, 1) = (0, 1)$
 Then $\text{suggest}(pp, 2) = (1, 2)$
 Then $\text{suggest}(pp, 3) = *$

But since the suggestion of a single data point is not the final goal suggest must be a function on the whole dataset DS . The new $\text{suggest}(DS, i)$ will only take those values into account that are actual values, i.e. $*$ will not be taken into the equation. The goal is to have $\text{suggest}(DS, i)$ return the overall best guess for bit i of the complete provenance mark.

A best guess can be defined in many ways. There are two main ways ([7]):

- *All-Vote*: pick the suggestion that most data points suggest (independently of the respective confidences)

$$\text{allVote}(L) = \text{round}\left(\frac{\sum_{(b,c) \in L} b}{|L|}\right)$$

- *Best-Vote*: pick the suggestion that has the most confident suggestions in data points

$$\text{bestVote}(L) = \text{round}\left(\frac{\sum_{(b,c) \in L'} b}{|L'|}\right) \text{ with } L' \text{ being the subset of } L \text{ containing only the best confidence values}$$

Using either of the two methods for $\text{suggest}(DS, i)$ the provenance mark can be constructed:

$$m_i = f(\text{suggest}(DS, i))$$

The newly created mark now needs to be checked using one-bit checking. This method is very robust. From a corrupted dataset with mostly uncorrupted check bits the provenance mark can still be constructed. Search using *mc-consistency* will then hopefully lead to a correct provenance mark.

3.3.4 Directed Search

Searching can become a very expensive and extensive process. Certain aspects have to be taken into account in order to keep the time consumption in a decent limit. For a directed search all the bits of a (guessed) provenance mark will be ordered by confidence. This will (as before) now lead to the main guess. But in case the best guess is not correct it is easy to try out similar possibilities.

| | | | | | |
|-----------------------------|-----|-----|-----|---|-----|
| Provenance mark: | 1 | 1 | 0 | 1 | 0 |
| Confidence in bits: | 0.5 | 0.8 | 0.2 | 1 | 0.4 |
| Order by confidence: | 1 | 1 | 1 | 0 | 0 |

Figure 6: Example for ordering by confidence

As seen in the example above (Figure 6) this order is easily done. Now, after an incorrect guess the bit with the lowest confidence will be flipped (green bit) and then the new mark will be tried out.

The new mark would be: 11110

Using a recursive algorithm as seen in [7] will try out all possible marks (starting with the most confident guess and ending with the least confident guess):

```
search(n, m):
  if n=0 then
    check possible provenance mark m
  else
    search(n-1, m)
    flip bit  $i_n$  of m
    search(n-1, m)
```

$\text{search}(n, m)$ will be called with the best guess for the provenance mark (m) and n , which is either the length of m or a limit l set by the user ($l < |m|$).

l would lead to a checking of the 2^l most confident guesses.

3.4 Evaluation and Analysis

The authors of [7] included several pages of evaluation and analysis in their paper. Their algorithm was tested with a dataset that suffered the typical transformations (rounding, truncation, sampling of data points). This subsection provides a short summary of their results. One big point is that the method becomes the more robust against transformations the bigger the dataset is. A very big dataset will provide more uncorrupted data points and, in general, more redundancy. Secondly, embedding more metadata in every single bit vector will lead to better results but decrease the perceptibility. Although the mean over all data points will stay pretty much the same one might not want to replace too much data with a provenance mark. A compromise will result in good robustness and perceptibility. One would think, and that is clearly correct, that a higher number of provenance pieces k provides more redundancy and will therefore lead to a higher robustness because each bit of the provenance mark will appear in more distinct data points. This idea makes perfect sense, but only up to a certain point. Having too many distinct provenance pieces the method becomes less robust against sampling. More data points would be needed to guarantee that all different provenance pieces are included in the subset. As already mentioned earlier the *pc-consistency score* and the *mc-consistency score* end up being binomially distributed given a big enough dataset.

3.5 Issues and Summary

The presented technique of embedding a provenance mark into a sensor-dataset appears to be robust against transformations (up to a certain point) and is meant to be available as an open-source tool. This sounds like a perfect solution but some issues remain:

First of all, the mechanism is no security measurement. Given enough criminal energy a provenance mark could be removed from the dataset. But the authors of [7] see their technique rather as support for the "fair use" policies typical of publicly available sensor network data" [7]. Another point is that, even if the provenance mark is only embedded into the metadata part of each data point, some publishers might not want that. It still would (not significantly) change their data and with them offering it they might feel like they were offering some wrong data. Two more factors are that it is not clear at what stage of the process the provenance mark should be embedded (raw sensor data or processed data). With that being unclear confusions and problems could arise. Also might the presented algorithm (especially the use of the hash-function) lead to a high time consumption in very big datasets. This would force one to use a cryptographically less expensive (and therefore less secure) method. Overall there are good applications for the technique and, overcoming the few issues mentioned, it can be applied for real-life applications.

4. CONCLUSION

This paper presented the topic "Electronic Watermarking" starting with its origin, development and applications. After that general approach one specific, new method is presented: Self-identifying sensor data uses the noise in a dataset to embed provenance marks into each data point. This approach is to guarantee that data can successfully be related to a creator/owner, even if several transformations corrupt the dataset. In comparison to the "traditional" applications of

watermarking the technique for sensor data is new and quite different. The main difference in the new technique derives directly from the need for it. Oppositely to the common datasets which watermarking is being applied to sensor data will be given in an unstructured way. Normal database watermarking techniques can therefore not be applied to it. Secondly, sensor data also varies between user communities and therefore an adaption providing more robustness had to be provided. Summarized one can say that the new technique is stronger because it opposes no structural prerequisites to the raw data. It can therefore find applications in many fields where the traditional methods could not work.

5. REFERENCES

- [1] *Digital Watermarking Alliance*, <http://www.digitalwatermarkingalliance.org/>, Webpage found on: April, 24th, 2012
- [2] *Civolution*, <http://www.civolution.com/technology/digital-audio-and-video-watermarking/>, Webpage found on: April, 24th, 2012
- [3] *IBM Research*, <http://domino.watson.ibm.com/library/cyberdig.nsf/0/21bddc7a17e34fd1852565930070afbb?OpenDocument>, Webpage found on: April, 24th, 2012
- [4] Ingemar J. Cox, Matt L. Miller: *The first 50 years of electronic watermarking*, NEC Research Institute, 4 Independence Way, Princeton, NJ 08540, USA, April 19, 2002
- [5] William M. Tomberlin, Louis G. MacKenzie, Paul K. Bennett: *System for transmitting and receiving coded entertainment programs*, United States Patent, 2,630,525, 1953
- [6] Anoop M. Namboodiri, Anil K. Jain: *Multimedia Document Authentication using On-line Signatures as Watermarks*, Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824
- [7] Stephen Chong, Christian Skalka, Jeffrey A. Vaughan: *Self-Identifying Sensor Data*, Stockholm, Sweden, April 12-16, 2010
- [8] Prof.Dr.Eng. Monica Borda: *Fundamentals in Information Theory and Coding*, ISBN: 978-3-642-20346-6, Springer-Verlag, Berlin Heidelberg, 2010
- [9] *Google Maps*, <http://maps.google.de/maps?hl=de&tab=w1>, Webpage found on: April, 24th, 2012
- [10] *Shazam*, <http://www.shazam.com>, Webpage found on: April, 24th, 2012
- [11] Avery Li-Chun Wang: *An Industrial-Strength Audio Search Algorithm*, Shazam Entertainment, Ltd., Palo Alto, CA, USA
- [12] Emil Frank Hembrooke. Identification of sound and like signals. United States Patent, 3,004,104, 1961.

How does an App Store / Market work?

Thomas Behrens
Betreuer: Dipl.-Inf. Marc-Oliver Pahl
Seminar Future Internet SS2012
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: behrenst@in.tum.de

KURZFASSUNG

Ein Application Store (App Store) ist ein internetbasiertes Vertriebssystem für Software, das vor allem durch die steigende Verbreitung von Smartphones und Tablet-PCs Beliebtheit erlangt. Es ermöglicht dem Anwender sein Gerät mit zusätzlichen Programmen (Apps) funktional zu erweitern. Die vorliegende Arbeit stellt einzelne Bestandteile des Systems „App Store“ vor und geht auf Fragestellungen und Konzepte hinter diesem System ein. Diese sind für die Implementation eines App Store wichtig, aber auch um die Komplexität eines solchen Systems zu verstehen. Außerdem wird auf Vertreter sowie deren Unterschiede und Gemeinsamkeiten eingegangen.

Schlüsselworte

App Store, App Market, Google Play, Nokia Ovi Store, Microsoft Marketplace, BlackBerry App World, App

1. EINLEITUNG

Der erste App Store wurde im März 2008 [1] von Apple unter dem Namen „Apple App Store“ eröffnet. Seitdem wird von verschiedenen Unternehmen versucht eigene App Stores zu implementieren und das Konzept zu verbessern. Viele Konzepte, auf denen der App Store beruht, existierten schon vorher und wurden von Apple lediglich weiterentwickelt und auf mobile Endgeräte portiert. Seit 2001 wird versucht mit der Einführung neuer Geschäftsmodelle auf die hohe Anzahl an Raubkopien von Musik und anderen digitalen Inhalten im Internet zu reagieren. Die ersten Vertreter von sogenannter Electronic Software Distribution (ESD) sind die Musikanbieter Napster (seit 2001) [2], Musicload (seit 2003) [3] und Apples iTunes Store (seit 2003) [4]. Auch werden Computerspiele seit dieser Zeit über eine Form von App Store für PCs verteilt (Steam: seit 2003) [5]. Die stetig steigenden Downloadzahlen u.a. in Apples App Store zeigen wie erfolgreich das Konzept ist. [6]. Die Anzahl der Apps vervielfacht sich seit der Einführung des Stores jedes Jahr, genauso wie die Zahl der Entwickler und der mit dem Store generierte Umsatz, der 2011 bei 1,78 Milliarden US-Dollar lag [7, 8, 9].

Als Folge von Apples Erfolg mit diesem Konzept und der daraus resultierenden Nachfrage nach deren Geräten [10, 11] gab es bald viele Konkurrenten. Im August 2008, im selben Jahr wie Apple, eröffnete Google den Android Market (seit 2012 Google Play). Neben diesen zwei größten, gibt es mehrere kleinere App Stores, im Juni 2010 circa 80 [12]. Da nur wenige dieser App Stores wirklich erfolgreich wurden, muss es Unterschiede geben, die den Erfolg wesentlich beeinflus-

sen. Solche Unterschiede gibt es in Realisierungskonzepten, die sich u.a. im Tätigkeitsfeld der App Store-Betreiber unterscheiden. Damit werden bestimmte Spielräume für Entwickler und Möglichkeiten für Endanwender festgelegt. Auch gibt es für App Stores technologische Grundkonzepte und Funktionen, die vorhanden sein müssen, aber verschieden realisiert werden können.

Im Folgenden werden zunächst Grundtechnologien von App Stores erläutert und aktuelle Entwicklungen zu deren Verbesserung vorgestellt. Kapitel 3 beschäftigt sich mit Vergütungsmodellen, also den verschiedenen Möglichkeiten für App Store-Betreiber und App-Entwickler, mit ihrer Arbeit Geld zu verdienen. Darauf folgend werden in Kapitel 4 Betreibermodelle vorgestellt und verglichen. In Kapitel 5 werden die bekanntesten App Stores vorgestellt und auf die größten Unterschiede und Gemeinsamkeiten zwischen diesen eingegangen. Die Arbeit endet mit der Vorstellung von ähnlichen Arbeiten und einem Ausblick auf die Entwicklung von App Stores.

2. TECHNISCHE GRUNDLAGEN EINES APP STORES

Ein App Store ist ein System zur Erweiterung eines Geräts mit Apps und deren Verwaltung. Die Funktionalität eines App Stores beschränkt sich aber nicht auf die Bereitstellung von zum jeweiligen Gerät kompatible Anwendungen. Dieser Abschnitt beschreibt verschiedene Funktionen, die ein App Store typischerweise bereitstellt, bzw. die bei der Realisierung zu berücksichtigen sind. Falls es neuartige Entwicklungen zu bestimmten Funktionen gibt werden diese vorgestellt. Dabei wird vor allem auf die Nutzerfreundlichkeit und die Einfachheit der einzelnen Funktionen geachtet. Dies liegt darin begründet, weil das Ziel eines App Stores es ist, durch möglichst alle Nutzergruppen bedienbar zu sein, egal wie hoch das technische Verständnis ist. Apple bewirbt seinen Mac App Store beispielsweise damit, dass man sich „die gewünschten Apps einfacher holen [kann] als je zuvor. Keine Verpackung, keine weiteren CDs oder DVDs und keine zeitraubende Installation. Ein Klick reicht, um Apps auf deinen Mac zu laden und zu installieren.“ [13]

Die ausführliche Vorstellung der Grundfunktionen in App Stores wird mit drei wesentlichen Funktionen eines App Stores, dem Finden einer bestimmten App, dem Herunterladen und der anschließenden Installation begonnen. Anschließend werden die Aktualisierung und die Funktionen für Bewertungen und Kommentare vorgestellt und erläutert, welchen

Einfluss letztere innerhalb des App Stores haben. Darauf werden Sicherheitskonzepte vorgestellt, die bei der Realisierung eines App Store mitberücksichtigt werden sollten. Dabei wird auf Kernunterschiede in aktuellen Realisierungen eingegangen. Danach werden Möglichkeiten vorgestellt, App-Entwickler zum Programmieren für die eigene Plattform zu motivieren. Darauf folgen eine Vorstellung von Möglichkeiten der Qualitätssicherung der Apps, geräteunabhängige Apps und Vorteile, die sich durch die Unterstützung eines Stores von mehreren Geräten ergeben.

2.1 Suchen

Die Suchfunktion ist eine Grundvoraussetzung um aus vielen tausenden Apps die Gewünschte finden. Mit einer stetig steigenden Zahl von derzeit über 500.000 Apps im Apple App Store ist dies keine leichte Aufgabe [14]. Ziel dieser Funktion ist es ohne spürbare Wartezeit ein möglichst gutes Ergebnis anzuzeigen. Damit wird Frust verhindert, der ansonsten bei Nutzern zu einer ablehnenden Haltung in Bezug auf den App Store führen würde. Heutige App Stores optimieren dies, indem sie schon während der Eingabe des Suchbegriffs (inkrementelle Suche) per Autovervollständigung versuchen, das richtige Suchergebnis anzuzeigen. In dieselbe Richtung geht der Versuch, Rechtschreibfehler im Suchbegriff zu erkennen und mit Hilfe einer Autokorrekturfunktion zu berichtigen [15]. Des Weiteren lassen sich Bewertungen dazu einsetzen, die Suchergebnisse in ihrer Relevanz zu sortieren und damit zu verbessern. Als weitere Eingabemöglichkeit für Suchbegriffe ist teilweise eine Spracheingabe vorhanden [16].

2.2 Herunterladen

Wenn der Anwender eine gewünschte App gefunden hat, muss sie als danach aus dem Internet heruntergeladen werden. Da eine Benutzerinteraktion für diesen Vorgang überflüssig ist und außerdem zusätzliche Nutzerdialoge nur zu mehr Komplexität führen würden, erfolgt der Download automatisch. Um diese Funktion dauerhaft gewährleisten zu können, muss eine ausreichende Infrastruktur zur Verfügung stehen. Es gibt mehrere Entwicklungen, die die Infrastruktur entlasten und damit die Kosten für App Store-Betreiber verkleinern. Dazu zählen z.B. dynamisch erweiterbare Cloud-Server, die Verwendung von Content Delivery Networks (CDN) und GeoTargeting.

2.3 Installieren

Um die heruntergeladene App auszuführen, ist deren Installation nötig. Darin wird das komprimierte, heruntergeladene Paket entpackt und für die Ausführung auf dem Gerät konfiguriert. Für den Anwender bietet die Installation durch einen App Store den Vorteil, dass sie einfacher ist gegenüber anderen Installationen. Während z.B. unter Microsoft Windows in mehreren Dialogfenstern u.a. der vom Anwender gewünschte Speicherort und Umfang der Installation abgefragt wird, ermöglicht beim App Store eine festgelegte Programmierschnittstelle (API), auf nötige Benutzerinteraktionen während der Installation vollständig zu verzichten. Ein Nachteil davon sind die mangelnden Konfigurationsmöglichkeiten, die es bei Smartphones auf Dateiebene aufgrund von eingeschränkten administrativen Möglichkeiten kaum gibt. Diese werden aber nicht benötigt, da in den Spezifikationen festgelegt ist, wie Apps auf dem Gerät installiert werden sollen, also wie die empfangenen Daten entpackt und

in das Dateisystem integriert werden sollen [17]. Außerdem wird durch die API eine Laufzeitumgebung erzeugt, welche es ermöglicht, dass Apps auf verschiedenen Geräten mit unterschiedlicher Hardware gleich laufen. Die API stellt dabei für App Entwickler einheitliche Funktionen zur Verfügung, die Gerätehersteller an die jeweilige Hardware übersetzen müssen. Deshalb sind gerätespezifische Konfigurationen nur noch in Ausnahmen erforderlich.

2.4 Aktualisieren

Mit dem Aktualisieren der Apps nimmt der App Store dem Anwender eine sonst notwendige Aufgabe ab. Dies ist notwendig, da immer wieder Schwachstellen in Apps oder kritische Fehler auftreten und mit so behoben werden können. Aktualisierungen werden durch eine Bestätigung des Anwenders gestartet und laufen ab dem Zeitpunkt automatisch. Möglich ist auch wie bei Google Play die Option anzubieten, auf die Bestätigung zu verzichten und automatisch zu aktualisieren. Ein App Store nimmt dem Anwender bei der Aktualisierung viele Aufgaben ab. Das lässt sich am besten am Beispiel eines Nutzers mit einem Betriebssystem ohne App Store z.B. Microsoft Windows verdeutlichen. Dieser muss, wenn er seine Programme aktuell halten will, die aktuelle Version über das Internet suchen, diese mit der installierten vergleichen und im Fall eines vorhandenen Updates die neue Version herunterladen. Danach muss die alte Version deinstalliert und in mehreren Installationsschritten die neue Version installiert werden. App Stores führen diese Schritte automatisch ohne Eingreifen des Anwenders durch. Die Version jeder App wird im App Store mit der jeweils aktuellen verglichen und, falls eine neuere zur Verfügung steht, die Möglichkeit gegeben, ein Update durchzuführen.

2.5 Bewerten & Kommentieren

Um andere Nutzer vor schlechten Apps zu warnen bzw. sie auf gute aufmerksam zu machen, lassen sich Apps bewerten. Bei den zwei größten App Stores, dem Apple App Store und Android Play, aber auch anderen, finden Bewertungen auf einer Skala von 0 bis max. 5 Sternen statt. Der Durchschnitt dieser Bewertung wird in den Bestenlisten, sowie vor jeder Installation auf der Infoseite der App angezeigt. Es gibt aber eine Bewertungsverzerrung, da Nutzer vor allem besonders gute und besonders schlechte Apps bewerten. Außerdem bewerten nur eine sehr geringe Zahl an Nutzern eine App, in einem repräsentativen Test durchschnittlich nur zwischen 0,008 und 0,056 Prozent [18]. Um diesem Problem zu begegnen, kann man zusätzlich zu den oben beschriebenen expliziten Bewertungen implizite Bewertungen einführen. Dabei nimmt der App Store oder ein im Hintergrund laufendes Programm das Benutzerverhalten auf und sendet es zur Auswertung an einen Server. Implizite Bewertungen setzen sich zusammen aus Ereignissen wie z.B. Installationen, Aktualisierungen und Deinstallationen, sowie Erkenntnissen über die Installationsdauer, die Anzahl der durchgeführten Updates sowie die Häufigkeit und Dauer der jeweiligen Ausführung einer App. Diese Informationen werden an einen Server geschickt, auf welchem die Informationen dann zusammen mit denen von anderen Nutzern ausgewertet und abgeglichen werden. Damit ist es u.a. möglich Empfehlungslisten zu verbessern. Diese können einzelnen Anwendern Apps empfehlen, die andere User mit einem ähnlichen Verhalten und ähnlichen Apps gerne nutzen. [18]

Eng mit dem Rating verknüpft ist die Kommentarfunktion. Hier können Nutzer über ihre Erfahrungen mit der App berichten und bei ihnen aufgetretene Fehler melden. Der Entwickler kann so erkennen, was die Nutzer an der App kritisieren und an diesen Stellen die App verbessern [19]. Das ist z.B. bei Google Play wichtig, da Android auf vielen verschiedenen Plattformen von verschiedenen Herstellern läuft. Diese haben jeweils die Android-Version angepasst und verwenden zum Teil unterschiedliche Displayauflösungen, was teilweise zu Problemen und Layout-Fehlern bei bestimmten Apps führen kann [20].

Ein Problem stellt die Zusammengehörigkeit von Wertung und Kommentar dar [21]. Wird eine App in einer früheren Version mit vielen Fehlern schlecht bewertet, ist es für den Entwickler sehr schwer, eine gute Gesamtbewertung zu erlangen, obwohl die bewerteten Fehler schon behoben wurden. Die Möglichkeit, dass der Entwickler überholte Kommentare und Bewertungen entfernen kann, würde jedoch dazu beitragen, dass Bewertungen manipulierbar wären. Eine Lösung ist es, implizite Bewertungen zu sammeln und solche Wertungen weniger stark zu gewichten. Eine andere Möglichkeit ist, eine Bewertungsfunktion für Kommentare einzusetzen und besonders schlecht bewertete Kommentare zu entfernen oder abwertend zu gewichten.

2.6 Sicherheitskonzepte

Da Smartphones und andere mobile Geräte vollwertige Computer sind und viele Informationen über den Anwender speichern, müssen sie gut geschützt werden [22].

Zugriff auf die Daten kann ein Angreifer z.B. durch das Ausnutzen einer Sicherheitslücke bekommen. Um solche schnell zu schließen, ist die automatisierte Verteilung von Patches bzw. Updates über den App Store sinnvoll. Das bedeutet, die Updates werden aktiv dem Anwender gemeldet, ohne dass dieser nach Updates suchen muss. Ein Beispiel hierfür ist Steam, da sich hier Anwendungen nur starten lassen, wenn sie sich vorher auf den aktuellen Stand gebracht haben.

Ein anderes Sicherheitsfeature kann die Erstellung von Backups im App Store und deren Wiederherstellung sein. Entweder es werden dabei nur die installierten Apps gespeichert oder zusätzlich noch Einstellungen und gespeicherte Inhalte hinzugefügt. Man könnte nur speichern, welche Apps man installiert hat oder noch die Einstellungen und gespeicherten Inhalte dazu nehmen.

Ein großes Risiko steckt für den Anwender im Download von bösartigen Apps, z.B. Spyware oder Viren, aus dem App Store. Es gibt verschiedene Ansätze das Risiko durch solche Apps in den Griff zu bekommen. Google Play verwendet ein Suchprogramm mit dem Namen „Bouncer“. Das analysiert neu in den App Store eingestellte und vorhandene Apps bzw. deren Code und überprüft Entwickler-Accounts auf Auffälligkeiten. Laut Google ließ sich damit der Anteil dieser Apps innerhalb von 2011 um 40% reduzieren. [23]

Apple hingegen lässt nur zusätzlich von Personal überprüfte Apps in den Store [24]. Dazu waren im Mai 2010 40 Leute angestellt, die jeweils zu zweit jedes App vor der Zulassung in den Store überprüfen [25]. Da diese 40 Personen aber an einem Tag ca. 2.000 Apps überprüfen, kann es vorkommen, dass sie trotz der zusätzlichen Prüfung schadhafte Apps übersehen. Zu den Prüfungsmechanismen sind kei-

ne Details bekannt, da diese dann leichter umgangen werden könnten. Es gibt aber so genannte Review Guidelines, die auflistet aus welchen Gründen eine App zurückgewiesen werden kann. Denen ist zu entnehmen, dass u.a. Abstürze und gewalttätige, rassistische oder jugendgefährdende Inhalte zum Ausschluss führen. Außerdem muss sie für den Anwender einen Mehrwert bieten und sich von bestehenden Apps unterscheiden. Kritisiert wird Apple dafür, dass Apps nach deren Ermessen und aus unverständlichen Gründen abgelehnt werden, z.B. weil sie sich zu wenig von existierenden und vor allem firmeneigenen Apps unterscheiden. [24]

Trotz aufwendiger Prüfungen gelangen, u.a. durch die große Zahl an neuen Apps, immer wieder Schadprogramme in den App Store. Deshalb schottet Android die Apps mit Hilfe von Sandboxing voneinander ab, so dass der Absturz einer App keine Auswirkung auf andere Apps hat. Somit gibt es auch jede App nur auf ihre eigenen Daten und nicht die anderer Apps zugreifen. [26]

Des Weiteren gibt es in Android ein Sicherheitsmodell, in dem jede App vor der Installation nach den erforderlichen Zugriffsrechten auf die Hardware fragt (Permission Modell) [26]. Wenn z.B. ein Spiel die Rechte für den Zugriff auf die Kontaktdaten erhalten möchte, sind Zweifel über die Absichten des Apps angebracht. Der Anwender kann so dubiose Apps erkennen und daraufhin entscheiden, ob er der App den Zugriff gestatten will. Kritisiert wird, dass die letzten beiden Schutzmechanismen bei Android nicht zwingend zwischen Anwendungen des gleichen Entwicklers gelten. Diese können über sogenannte Intents Daten austauschen, wodurch sich Zugriffsrechte kombinieren lassen und somit umgangen werden können. [26]

Deshalb wird an Software geforscht, die es dem Endanwender erlaubt zu sehen, auf welche Daten und Hardwarefunktionen das App wann zugreift, und welche Daten es versendet. Dazu wurde von der TU Berlin für Android eine Sandbox entwickelt (AASandbox), die es dem Anwender ermöglicht, für eine beliebige App eine statische und dynamische Analyse durchzuführen und bösartige Programme automatisch zu erkennen. Die statische Analyse wird dabei auf den Installationsdateien ausgeführt, die dynamische über den von Google zur Verfügung gestellten Android Emulator. [27]

Ein weiteres Sicherheitsfeature erlaubt es Google Play, als Malware erkannte Apps per Fernzugriff von allen Geräten zu löschen, auf denen diese App installiert ist [26].

2.7 Entwicklerunterstützung

Der Erfolg eines App Stores hängt neben der Qualität auch von der Anzahl der angebotenen Apps ab. Deshalb ist es für App Store-Betreiber wichtig, Entwicklern Anreize zu schaffen und sie bestmöglich zu unterstützen, Apps für die eigene Plattform zu programmieren.

Dazu gehört die Bereitstellung eines Software Development Kits (SDK), mit dem die Entwickler arbeiten können. Der Umfang der Programme in diesem Paket ist stark abhängig vom App Store-Betreiber. Im Idealfall finden sich ein guter Debugger, Bibliotheken, ein Emulator, eine ausführliche Dokumentation, Beispielcode und Anleitungen (Tutorials). Die Bibliotheken sind wichtig, um den standardisierten Zugriff auf Hardware zu vereinfachen und um die Entwick-

lungszeit zu verkürzen. In ihr enthalten sind Grundfunktionen des Geräts, wie z.B. Interface-Elemente, Zugriffe auf Telefonbuch/Kamera/Sensoren, die mit wenigen Befehlen in die eigene App integriert werden können. Eine ausführliche Dokumentation der zur Verfügung stehenden Schnittstellen und Richtlinien ist auch von Vorteil, da es z.B. die Einarbeitungszeit verkürzt. Mit vorhandener API und Bibliothek lässt sich u.a. auch die Benutzerführung homogener gestalten, was den Vorteil hat, dass Anwender schneller die Bedienung einer beliebigen App erlernen.

Unter bestimmten Voraussetzungen ist es für Entwickler einfacher mit einer App erfolgreich zu werden. Grundsätzlich sollte eine Nachfrage nach neuen, innovativen Apps vorhanden sein, also der Markt noch nicht übersättigt sein. Je mehr Anwender eine Plattform hat, umso mehr steigt die Wahrscheinlichkeit, mit der eigenen App erfolgreich zu sein. Die Betreiber von App Stores mit weniger Anwendern versuchen diesen Wettbewerbsnachteil u.a. durch direkte Zahlungen an gute Entwickler, auf die sie im Konkurrenzkampf dringend angewiesen sind, auszugleichen. Hier lassen sich Nokia und Microsoft nennen, die im März 2012 ein Förderungsprogramm für App-Entwickler in Höhe von 18 Mio. Euro beschlossen haben [28].

Des weiteren ergeben sich durch eine Zusammenarbeit von App-Entwicklern und dem App Store-Betreiber beim Marketing für beide Seiten Vorteile. Der App Store-Betreiber hat das Ziel, seinen Store mit möglichst herausragenden Apps zu bewerben. Der App-Entwickler strebt eine möglichst hohe Aufmerksamkeit für seine App an. Es kann z.B. der App Store-Betreiber in einer Kategorie „Featured Apps“ für die App werben oder direkt im App Store Werbung dafür schalten und dafür z.B. einen höheren Anteil am Umsatz dieses Apps erhalten.

Für die Entwicklung von Apps ist es von Vorteil, die Anzahl der Geräte und deren Betriebssystemversionen zu kennen, die zurzeit auf den Store zugreifen. Damit kann der Entwickler abschätzen, wie groß die Zielgruppe bei bestimmten Hardwareanforderungen an die App ist. Außerdem kann damit die Entscheidung der Betriebssystemversion erleichtert werden, die zwar Zugriff auf zusätzliche Bibliotheksfunktionen liefert, aber inkompatibel zu älteren Geräten ist. Das ist auch für die Entwicklung von Updates interessant, wobei die Statistik über die Geräte auf denen die App installiert wichtiger ist. Dazu gibt es z.B. von Google das „Android Developer Device Dashboard“ [29] oder von Steam die „Steam Hardware & Software Survey“ [30].

Noch besser als nur die Geräte zu kennen, ist die Möglichkeit für unterschiedliche Versionen eines Betriebssystems seine App zur Verfügung zu stellen. Damit kann man Nutzern eines besseren Geräts eine anspruchsvollere Version zur Verfügung stellen, ohne die Anwender mit schwächeren Geräten zu verlieren.

Weil die Ablehnung einer App durch den App Store-Betreiber in vielerlei Hinsicht ein schwerer Schaden für den Entwickler ist, sollte außerdem der Review-Prozess so klar definiert und durchgeführt werden, dass eine Ablehnung für den Entwickler kalkulierbar und nachvollziehbar ist.

2.8 Qualitätssicherung

Es genügt aber nicht nur eine große Anzahl an neuen Apps, diese sollten auch die qualitativ hochwertig sein. Denn wenn der Großteil der Apps nur einen geringen Mehrwert bietet, als Negativbeispiel lässt sich hier eine „Farting App“ [24] nennen, ist der von einem Anwender gewonnene Gesamteindruck auch negativ. Wie bereits dargelegt, kann man die Zulassung zum Store verschärfen, und damit neben Viren und Trojanern auch schlecht programmierte Apps herausfiltern. Es ließe sich aber auch z.B. ein Zertifizierungsverfahren durchführen, bei dem sich Apps von einem Kontrollgremium gegen eine kleine Gebühr nach festen Kriterien auf Qualität von Software und Inhalt prüfen lassen. Dem Anwender kann damit die Möglichkeit geben werden, sich nur zertifizierte Apps anzeigen zu lassen. Eine solche Zertifizierung wäre aber recht aufwendig zu realisieren, da im optimalen Fall nicht nur überprüft wird, ob die App die Qualitätsstandards des App Stores erfüllt, sondern z.B. auch ob Anwendung für ihren Zweck effizient programmiert ist und wie hoch die Benutzbarkeit und Bedienbarkeit ist. Der TÜV Süd bietet z.B. eine App-Zertifizierung für derartige Kriterien an [31].

Zur Qualitätssicherung hat der Verband der deutschen Internetwirtschaft e.V. eine Umfrage durchgeführt. Es wurde ermittelt, „wie [man] die Qualitätssicherung bei mobilen Anwendungen in Shops sicherstellen [kann]“ [32]. Dabei sprachen sich über 70% der Befragten für eine Qualitätssicherung bei der Annahme in den App Store aus, fast 20% hielten eine Qualitätssicherung für nicht erforderlich und ca. 8% befürworteten verschiedene Qualitätsstufen, die entsprechend Geld kosten. Da aber solche Zertifizierungen sehr teuer sind, und damit die Vorteile des leichten Marktzugangs verschwinden würden, ist die Umsetzung, wie schon die Umfrage gezeigt hat, fragwürdig.

2.9 Geräteunabhängige Apps

Ein Problem der heutigen Smartphones ist die Beschränkung eines Apps auf normalerweise nur ein einzelnes Betriebssystem. Damit ist es für Entwickler unmöglich, ohne Adaption des Apps auf andere Betriebssysteme den Großteil des Marktes zu erreichen. Um diese Beschränkungen zu umgehen werden geräteunabhängige Programmiersprachen, wie z.B. Java, Flash oder HTML5 verwendet.

HTML5 ist die Weiterentwicklung der von Webseiten bekannten HTML-Sprache, bietet aber weitere Möglichkeiten der Programmausführung. Dabei findet nur eingeschränkter Hardwarezugriff statt, aber auf Funktionen wie Kamera und GPS kann über API-Schnittstellen trotzdem zugegriffen werden. Zulassungsbeschränkte App Stores wie z.B. der von Apple weigern sich u.U. solche plattformunabhängigen Apps aufzunehmen. Das liegt u.a. daran, dass diese Apps nicht auf eine bestimmte Plattform optimiert sind.

2.10 Unterstützung mehrerer Geräte

Ein weiteres häufig beworbenes Feature ist die Vernetzung mehrerer Geräte über den App Store. Derzeit ist standardmäßig die Lizenz, die man beim Kauf einer App erhalten hat, sowie eine Liste der kostenlos installierten Programme beim App Store-Betreiber gespeichert [33]. Das hat den Vorteil, dass beim Kauf eines Geräts mit demselben App Store auf Wunsch die bereits auf dem anderen Gerät installierten Programme überspielt werden können.

Ein weiterer Schritt in diese Richtung kommt durch die starke Verbreitung von Cloud-Diensten. Es ist möglich nach jedem Beenden einer App, im Falle von ausreichend Netzwerkkapazitäten (WLAN), diese in auf einen Server bzw. in die Cloud hochzuladen. Von dort wird diese an die anderen eigenen Geräte verteilt. Somit lässt sich an allen Geräten immer an dem aktuellen Stand und den aktuellen Einstellungen weiterarbeiten.

3. VERGÜTUNGSMODELLE

Zur Motivation von Entwicklern gehört es auch finanzielle Anreize zu schaffen. Ein App Store sollte deswegen attraktive Vergütungsmodelle anbieten, die für den Entwickler einfache Kalkulationen ermöglichen, um sein Risiko und seinen möglichen Gewinn einschätzen zu können. Allgemein gibt es bei der Vermarktung von Apps fünf Ansätze. Der Verkauf von Daten über den Anwender, Werbung, Partnerschaften, das Einwerben von Spenden und der Verkauf der Apps. Weil Spenden unüblich und unkalkulierbar sind und sich Partnerschaften immer zusätzlich zu den Ansätzen abschließen lassen, ergeben sich daraus drei Hauptmodelle, die zurzeit von den größten App Stores angeboten werden.

1. Kostenpflichtige App:

Eine Möglichkeit mit Apps Geld zu verdienen, ist die App zu verkaufen. Dabei lässt sich ab einem gewissen Bekanntheitsgrad des Entwicklers, z.B. „EA Games“, die App auch zu niedrigen Preisen verkaufen, um einen guten Umsatz zu erzielen. Das gilt auch für Apps mit einem wirklichen Mehrwert, z.B. als Ersatz für ein Navigationssystem. Damit erklärt sich auch der niedrige Durchschnittspreis für kostenpflichtige Apps von unter zwei US-Dollar in den größten Stores [34].

2. Kostenlose App mit der Möglichkeit, In-App-Inhalte zu kaufen:

Da der Kauf einer kostenpflichtigen App mit einer gewissen Überwindung verbunden ist, wurde das Modell der In-App-Käufe eingeführt. Dabei ist die App für den Anwender kostenlos, kann aber über zusätzliche Inhalte erweitert werden. Damit der App Store-Betreiber aber trotzdem am Gewinn des Entwicklers beteiligt bleibt, ist hier die gleiche Kostenaufteilung wie im kostenpflichtigen Modell üblich. Diese Art von Apps werden vor allem dort eingesetzt, wo die App selber als Store fungiert, z.B. zum Kauf von E-Books oder Musikstücken.

3. Sonstige kostenlose App:

Apps, bei denen der Anwender zu keinem Zeitpunkt zahlen muss, erwirtschaften den Gewinn anders. So können über eine Werbefläche, die während der App-Ausführung angezeigt wird, Kleinstbeträge verdient werden, die sich aber bei der vielfachen Verwendung der App zu großen Beträgen aufsummieren. Diese Art von Apps bieten sich für Entwickler an, die noch nicht so erfolgreich sind. Wenn es dem Entwickler gelingt viele Nutzer zu gewinnen, kann er sich über die vielen Kleinstbeträge aus den Werbeeinnahmen finanzieren.

Der App Store-Betreiber behält meist einen Anteil vom Gewinn des Entwicklers, um sich zu finanzieren. Im App Store

und bei Google Play erhält der Entwickler 70% und der Betreiber 30% der Einnahmen [35]. In anderen Stores kann die Aufteilung z.B. je nach Zahlungsart variieren.

Der App Store-Betreiber sollte verschiedene Zahlungsmöglichkeiten anbieten, z.B. Kreditkarte, Micropayment-Dienste und Prepaid. Zusätzlich kann bei Kooperation zwischen den Betreibern von Mobilfunk und App Store die Zahlung per Handyrechnung vereinbart werden. Je mehr Zahlungsmöglichkeiten es gibt, umso leichter wird es dem Anwender fallen zu zahlen. Bei den meisten Betreibermodellen muss der App Store-Betreiber für die Abwicklung der Zahlungen diese erst absichern.

4. BETREIBERMODELLE

Es existiert eine Vielzahl an unterschiedlichen App Store-Betreibermodellen in den zu diesen Themen entstandenen technischen Studien (siehe verwandte Arbeiten). Dabei sind die Unterschiede zwischen den Modellen gering. Der Grund für die vielen Modelle ist, dass sie nach unterschiedlichen Gesichtspunkten entstanden sind und deshalb auf unterschiedliche Weise auseinander dividiert wurden. Im Folgenden wird auf ein Modell eingegangen, das an der Freien Universität Brüssel entstand [36]. Es unterscheidet die Plattformen nach der Kontrolle des App Store-Betreibers über die Inhalte seines Stores und die Kontrolle über die Kundenbeziehung zum Endanwender.

4.1 Enabler Plattform

Die Enabler Plattform kontrolliert die Inhalte im App Store, um den eigenen Wert sicherzustellen. Es findet aber nicht zwingend eine Kundenbeziehung statt, der Anwender kann also auch über andere Quellen seine Apps beziehen. Für diesen Typ von Plattform ist es wichtig, dass der Betreiber zum einen gute Beziehungen zur Industrie hat, zum anderen aber auch Wissen und Erfahrung besitzt, um für Verbraucher und Entwickler attraktive Plattformen zu entwickeln. Es sollte also eine IT-Infrastruktur und Werkzeuge zur Verfügung gestellt werden, die den App-Entwickler während des gesamten Entwicklungsprozesses unterstützen. Damit versucht der Betreiber, die Plattform möglichst attraktiv und damit beliebt für Entwickler, Industriepartner und auch Anwendern zu machen. Eingesetzt wird dieser Plattfortmtyt häufig für mobile Betriebssysteme. Beispielsweise stellt Google mit Android und Microsoft mit Windows Mobile eine Enabler Plattform zur Verfügung. [36]

4.2 System Integrator Plattform

Wie man in Tabelle 1 sehen kann, kontrolliert die System Integrator Plattform Anwender und Inhalte und muss damit die meisten Dienste der hier vorgestellten Plattformen übernehmen. Zusätzlich zur Entwicklung der Plattform und deren Support werden Aufgaben des Geräteherstellers, die Kundenbetreuung sowie die aktive Vermarktung für das Anwerben neuer Entwicklern übernommen. Dabei muss der Plattformbetreiber nicht unbedingt das Gerät selbst fertigen, er kann es z.B. auch bei einem Hersteller in Auftrag geben und dann unter seinem Markennamen (Branding) vertreiben. Verfügt der Betreiber einer Plattform über genügend Kompetenzen, um die unterschiedlichen Dienste zur Verfügung zu stellen, kann er u.U. mit diesem Modell erfolgreich werden, was z.B. Plattformen wie Apples iPhone bzw.

iOS, Microsoft Windows Phone und Nokia Ovi gelungen ist. Vorteil ist, dass weniger Absprachen zwischen den einzelnen Teilnehmern nötig sind, da die meisten Kompetenzen beim Plattformbetreiber gebündelt sind. [36]

4.3 Neutral Plattform

Die Neutralen Plattformen bieten vom Typ her den anderen Teilnehmern am wenigsten Dienste an, sind also komplementär zum System Integrator-Modell zu sehen. Es gibt weder eine Kontrolle über die Inhalte noch über die Endanwender. Neutrale Plattformen stellen für die Entwicklung der Apps grundlegende Werkzeuge, meist auf Open Source-Basis, zur Verfügung um eine einfache Softwareentwicklung zu fördern. Den Entwicklern werden aber ansonsten keine weiteren Dienste zur Verfügung gestellt und es wird auch keine Kundenbeziehung aufgebaut. Die Neutrale Plattform ist häufig ein Zusammenschluss von Mobilfunknetzbetreibern (Mobile Network Operators, MNOs), Geräteherstellern und Gerätevertreibern. Dies beruht darauf, dass die Plattform selbst nicht auf viel Gewinn ausgelegt ist, aber alle Beteiligten die Möglichkeit haben, zusätzliche Dienste anzubieten, mit denen sie jeweils Gewinn erzielen können. Haupterfolgskriterium ist die Zusammenarbeit zwischen den beteiligten Unternehmen und das Potential der einzelnen Unternehmen. Das Ziel des Zusammenschlusses kann zusätzlich auch das Erreichen neuer Standards und der Gewinn an technologischem Fortschritt sein. Beispielpattformen hiervon sind Bondi und die Linux Mobile (LiMo). [36]

4.4 Broker Plattform

Der Unterschied zwischen Enabler Plattform und Neutral Plattform ist die hinzugekommene Kontrolle über den Anwender. Hier hat der Betreiber keine Kontrolle mehr über die Inhalte in seinem Store, bietet diese aber den Endkunden an und erhält für jede Verkaufsabwicklung eine Provision. Die angebotenen Apps können je nach Betreiber unabhängig vom Endgerät sein, also mehrere Endgeräte unterstützen. Typischerweise setzen MNOs diese Art von Plattform um, weil sie Kunden mit unterschiedlichen Geräten haben und die Geschäftsbeziehung zu diesen Kunden aufrechterhalten wollen. Vertreter dieses Modelltyps sind der Vodafone Apps Shop, Handango und GetJar. [36]

| Plattformtypen: | | Kundenverhältnis zum Endanwender | |
|------------------------|------|----------------------------------|----------------------|
| | | Ja | Nein |
| Kontrolle über Inhalte | Ja | Enabler P. | System Integrator P. |
| | Nein | Neutral P. | Broker P. |

Tabelle 1: Überblick über vorgestellten Plattformtypen

Die Entscheidung für den Plattformtyp hängt also stark von den Kompetenzen und den Zielen des App Store-Betreibers ab. Die klare Strukturierung lässt sich im Vergleich der vorhandenen App Stores gut veranschaulichen.

5. APP STORES IM VERGLEICH

Im Folgenden werden die bekanntesten App Stores vorgestellt und jeweils Merkmale herausgestellt, mit denen sie sich von ihrer Konkurrenz abheben.

5.1 Apple App Store

Bis März 2012 wurden ca. 25 Milliarden Apps heruntergeladen, was ihn zu einem der erfolgreichsten App Stores macht [37]. Da der Umsatz und damit die Verkäufe von iPhones, iPads und Macs immer weiter ansteigen [10, 11], ist auch kein Einbrechen der guten Zahlen zu erwarten. Dabei schafft Apple über iCloud die Möglichkeit, Apps zwischen diesen Produkten zu transferieren, und somit seine Kunden stärker an sich zu binden. Konkurrenz bekommt der App Store von alternativen App Stores wie z.B. Cydia, die über einen sogenannten Jailbreak auf Apple-Geräten installiert werden können. Diese Konkurrenz entstand u.a. aus verschiedenen Kritikpunkten, wie z.B. den für viele Entwickler unverständlichen Zugangsbeschränkungen und Zensur. Als Beispiel lässt sich die Entfernung von mehreren Apps des Konkurrenten Google im Nachhinein nennen oder die Verweigerung Apps aufzunehmen, die nach Ansicht von Apple eine Konkurrenz gegenüber den eigenen Produkten und Sicherheitsrichtlinien darstellen. Auch die umfassende Datensammlung wird stark kritisiert [38].

5.2 Android

Die größte Konkurrenz zu Apple ist momentan Googles Open Source-Plattform Android. Die Anzahl an Smartphones dieses Betriebssystems wächst aufgrund von verschiedenen Herstellern und Modellen in verschiedenen Preissegmenten sehr viel stärker als die von Apple iPhones [39]. Im Juli 2011 war der Marktanteil von Android mit über 44% schon mehr als doppelt so hoch wie der von iOS [40].

5.2.1 Google Play

Google verfolgt ein offeneres Konzept mit seinem offiziellen App Store als Apple. Es dürfen zusätzliche App Stores auf den Geräten installiert und verwendet werden. Apps können z.B. auch ohne App Store installiert werden und Entwickler auf verschiedenen Wegen bei der Entwicklung unterstützt werden. Aus dem App Store werden Apps nur bei gemeldeten Verstößen oder einer Erkennung durch den Bouncer entfernt (siehe Kapitel 2.6)

5.2.2 Amazon Appstore for Android

Dieser App Store weitete Amazons Online-Vertriebssystem auf mobile Anwendungen aus und stellt zusätzliche Angebote dessen Gerät Kindle Fire zur Verfügung. Dabei ist es von Vorteil, dass viele schon einen Benutzeraccount bei Amazon haben und dort Zahlungsmittel hinterlegt haben. Amazon versucht bei der Zulassung zum Store ähnlich wie Apple die Apps zu überprüfen. Kritik kommt vor allem für ein Modell, nach dem Amazon eine beliebige App einen Tag kostenlos anbietet, was ohne Absprache mit dem Entwickler passiert und dieser dafür auch nicht entlohnt wird [41].

5.3 Windows Phone Marketplace

Der Windows Phone Marketplace kam 2010 für Geräte des Windows Phone 7 auf den Markt. Neben den üblichen Funktionen gibt es die Möglichkeit Apps mit begrenzter Laufzeit und Demo-Apps herunterzuladen und so auszuprobieren bevor man sie kauft.

5.4 BlackBerry App World

Ein weiterer Smartphone-App Store ist die von RIM für das Handbetriebssystem BlackBerry OS entwickelte BlackBerry App World. Besonderheiten sind die Möglichkeiten Apps an seine Freunde zu verschenken und neben Kreditkarte auch über Telefonrechnung oder PayPal zu zahlen. Obwohl es den Store erst seit 2009 gibt, ist er für Entwickler hoch profitabel. Mehr als 13% der Entwickler sollen laut einer Studie mit ihrem App bereits mehr als 100.000 Dollar verdient haben. [42]

5.5 Nokia OVI Store

Der mittlerweile in Nokia Store umbenannte App Store bietet Apps für die Betriebssysteme Symbian, Maemo und MeeGo an. Es ist aber auch möglich, andere Inhalte wie z.B. Java ME-, Flash-Anwendungen und Klingeltöne anzubieten. Als Besonderheit lässt sich das „Store-in-Store“-Konzept sehen, mit dem Nokia MNOs die Möglichkeit bietet, eigene Stores dort zur Verfügung zu stellen. [43]

5.6 Andere App Stores

Neben den größten App Stores für mobile Endgeräte gibt es noch ähnliche Konzepte auf anderen Systemen. Da diese einige interessante Besonderheiten aufweisen werden sie im Folgenden kurz vorgestellt werden.

5.6.1 Steam

Steam ist mit über 35 Mio. Benutzerkonten einer der führenden App Stores für Spiele an PCs. Das System unterstützt Raubkopie-Erkennung, also die Überwachung der Spiele auf Manipulationen, Online-Vertrieb, Verteilung und Aktualisierung der Spiele und mehrere soziale Features, wie z.B. die Möglichkeit Gruppen zu bilden und miteinander zu chatten. [5]

5.6.2 Origin

Dieses von Electronic Arts (EA) betriebene System orientiert sich an der Funktionalität des bereits vorgestellten Steam, hat aber mit 3,9 Mio. Nutzern wesentlich weniger Nutzer als dieses. Kritik bekommt das System, weil man für Software von EA gezwungen wurde, dieses System zu benutzen und der Lizenzvertrag EA weitreichende Rechte auf dem Computer des Anwenders einräumte. Dazu gehört z.B. die Sammlung von technischen und verwandten Informationen und deren Übertragung an EA und deren Partner. [44]

5.6.3 Ubuntu Software Center

Dieses für das Betriebssystem Ubuntu konzipierte Software Center ist das erste, was das App Store Prinzip vollständig für PC-Betriebssysteme umsetzt. Das System lässt sich über eine App Store-typische Oberfläche bedienen und greift bei der Installation von neuen Apps auf die Paketverwaltung Synaptic zurück [45]. Dabei fällt der große Nachteil von früheren Linux-Distributionen weg, dass sich Anwendungen nur schwer, z.B. über bestimmte Makefiles, installieren lassen, welche auch nicht immer auf jeder Distribution richtig funktionieren haben. Mit der gewonnenen Einfachheit, bringt das kostenlose System teurere Konkurrenzsysteme wie Microsoft Windows und Apples Mac OS in Zugzwang. Nachdem bereits ein Mac App Store eingeführt wurde [13], wird auch Microsoft für die nächste Windows Version einen App Store anbieten [46].

5.6.4 GetJar

GetJar gehört zu den größten Multi-Plattform Stores mit über 2,5 Mrd. Downloads und über 400.000 Entwicklern [47]. Wesentlicher Erfolgsfaktor ist die Unterstützung sehr vieler Plattformen u.a. Java ME, BlackBerry, Symbian, Windows Mobile und Android.

6. VERWANDTE ARBEITEN

Aufgrund der vielen unterschiedlichen Teilkomponenten von App Stores, konnte im Rahmen der Arbeit nicht auf alle im Detail eingegangen werden. Als verwandte und weiterführende Arbeiten seien folgende empfohlen:

- In [36] wird das unter „Betreibermodelle“ erklärte Modell ausführlicher vorgestellt und dabei u.a. noch genauer auf die nötigen Kompetenzen der Plattformbetreiber eingegangen, die sie für das jeweilige Modell mitbringen müssen.
- Auf das IISI(n)-Modell, ein Modell was andere Blickwinkel an App Stores beleuchtet, wird in [48] genauer eingegangen. Dort werden die zwei Plattformen Nokia OVI und Apple App Store genauer miteinander verglichen, wodurch sehr gut kenntlich gemacht wird welche Aspekte dieses Modell betrachtet.
- Bei [49] handelt sich um eine Arbeit, die für den Erfolg von App Stores wesentliche Aspekte vorstellt. Dabei wird vor allem auf mögliche Geschäftsmodelle eingegangen und ein entwickeltes dreidimensionales Modell vorgestellt, dass die treibenden Faktoren für Erfolg beschreiben soll.
- Wie bereits angedeutet gibt es eine Vielzahl an anderen Betreibermodellen und Aufteilungen in diese Modelle, auf die aufgrund der begrenzten Länge nicht eingegangen werden konnte. Bei Interesse seien die folgenden Arbeiten empfohlen: [50, 51, 52, 53, 54, 55].
- In [26] werden Sicherheitsmechanismen auf Android-Systemen detailliert behandelt. Mögliche Angriffe werden dabei vorgestellt und Maßnahmen aufgezeigt, wie diese verhindert werden können.

7. FAZIT & AUSBLICK

Hinter einem App Store, wie ihn ein Anwender wahrnimmt, stecken eine große Zahl an Funktionen und technischen Hintergründe. Durch die deutliche Vereinfachung für den Anwender im Umgang mit Programmen ist sowohl der App Store, als auch die App an sich zum Erfolgsmodell geworden. Die Grundfunktionen sind größtenteils mit einem Klick ausführbar, man muss sich weder um Updates kümmern, noch darum ob ein Programm überhaupt die Mindestanforderungen für ein Gerät erfüllt. Größter Nachteil von heutigen App Stores ist noch die große Inhomogenität bei den zur Programmierung der Apps für einen Store zugelassenen Programmiersprachen. Die Zukunft wird zeigen, ob sich Entwickler in absehbarer Zeit darauf konzentrieren können, gute Apps und nicht nur eine App in 7 verschiedenen Programmiersprachen zu schreiben. Denn obwohl jede Plattform so viele gute Apps wie möglich haben will, gibt es bislang noch keine großen Kooperationen zwischen den größten App Stores, um eine Unterstützung von Apps anderer Stores zu gewährleisten.

Obwohl es scheint als wäre der Markt an App Stores gesättigt, werden vielleicht noch weitere auf den Markt drängen und das erfolgreiche Modell an verschiedenen anderen Stellen in unserem Alltag integrieren. Ob es irgendwann ganz normal sein wird über App Stores Autos, Drucker oder Kameras um Funktionen zu erweitern, lässt sich nicht sagen. Fest steht, dass es sich um ein Erfolgsmodell handelt, das sich noch stärker verbreiten wird.

8. LITERATUR

- [1] Apple, "Apple Announces iPhone 2.0 Software Beta," 6. März 2008.
<http://www.apple.com/pr/library/2008/03/06AppleAnnounces-iPhone-2-0-Software-Beta.html>.
- [2] Roxio, "Napster's Launch Party To Feature Hottest New Music Artists," 23. Oktober 2003.
http://blog.roxio.com/press/divisioninvestor/2003/10/napsters_launch_party_to_feature_hottest_new_music.html.
- [3] Musicload, "Musicload – Zahlen und Fakten," 15. Juni 2011. <http://musicload.newsroomloads.de/facts/musicload-zahlen-und-fakten/06/2011/>.
- [4] Apple, "Apple Launches the iTunes Music Store," 28. April 2003.
<http://www.apple.com/pr/library/2003/04/28AppleLaunches-the-iTunes-Music-Store.html>.
- [5] Valve, "Steam Client Released," 12. September 2003.
<http://store.steampowered.com/news/183/>.
- [6] Apple, "Apple's App Store Downloads Top 15 Billion." Online, Juli 2011.
<http://www.apple.com/pr/library/2011/07/07Apples-App-Store-Downloads-Top-15-Billion.html>.
- [7] Statista (über 148Apps.biz), "Statistik: Anträge auf Freigabe von neu entwickelten Apps im iTunes App Store von 2008 bis 2011."
<http://de.statista.com.eaccess.ub.tum.de/statistik/daten/studie/157931/umfrage/antraege-der-entwickler-auf-veroeffentlichung-von-apps/>.
- [8] Statista (über 148Apps.biz), "Statistik: Anzahl verfügbarer Apps im US iTunes App Store bis Juli 2011 (in 1.000)."
<http://de.statista.com.eaccess.ub.tum.de/statistik/daten/studie/157934/umfrage/anzahl-der-apps-im-itunes-app-store-seit-2008/>.
- [9] Statista (über IHS), "Statistik: Globaler Umsatz ausgewählter App-Stores in den Jahren 2009 und 2010 in Millionen US-Dollar."
<http://de.statista.com.eaccess.ub.tum.de/statistik/daten/studie/180896/umfrage/weltweiter-umsatz-fuehrender-app-stores-seit-2009/>.
- [10] Statista (über Apple), "Statistik: Absatz von Apples iPhone weltweit vom 3. Geschäftsquartal 2007 bis zum 1. Geschäftsquartal 2012 (in Millionen Stück)."
<http://de.statista.com.eaccess.ub.tum.de/statistik/daten/studie/12743/umfrage/absatz-von-apple-iphones-seit-dem-jahr-2007-nach-quartalen>.
- [11] Statista (über Apple), "Statistik: Absatz von Apples iPhone weltweit vom 3. Geschäftsquartal 2007 bis zum 1. Geschäftsquartal 2012 (in Millionen Stück)."
<http://de.statista.com.eaccess.ub.tum.de/statistik/daten/studie/12743/umfrage/absatz-von-apple-iphones-seit-dem-jahr-2007-nach-quartalen/>.
- [12] T. Husson, "App store markets overhyped," Juli 2010.
<http://cat-iqconference.com/2010/07/26/app-store-markets-overhyped/>.
- [13] Apple, "App Store - Tolle Mac Apps," 30. März 2012.
<http://www.apple.com/de/mac/app-store/great-mac-apps.html>.
- [14] Apple, "The App Store," 30. März 2012.
<http://www.apple.com/iphone/built-in-apps/app-store.html>.
- [15] Apple, "Apple in Education," 23. März 2012. <http://www.apple.com/education/special-education/>.
- [16] Google, "Just speak it: introducing Voice Actions for Android," August 2010.
<http://googlemobile.blogspot.de/2010/08/just-speak-it-introducing-voice-actions.html>.
- [17] Google, "Debugging: Using the Dev Tools App," 20. März 2012. <http://developer.android.com/guide/developing/debugging/debugging-devtools.html>.
- [18] A. Girardello and F. Michahelles, "Explicit and Implicit Ratings for Mobile Applications," in *GI-Jahrestagung 2010*, pp. 606–612, Information Management - ETH Zürich, September 2010.
- [19] A. Hammershoj, "Challenges for mobile application development," in *Intelligence in Next Generation Networks (ICIN), 2010 14th International Conference*, IEEE, Oktober 2010.
- [20] Google, "Android Developers - Dev Guide," 21. März 2012. <http://developer.android.com/guide/appendix/market-filters.html>.
- [21] C. Dellarocas, "The Digitization of Word-of-Mouth: Promise and Challenges of Online Reputation Systems," in *Management Science Vol. 49, No. 10, Special Issue on E-Business and Management Science*, pp. 1407–1424, INFORMS, Dezember 2001.
<http://www.jstor.org/stable/4134013>.
- [22] L. H. Marcial, "A comparison of screen size and interaction technique: Examining execution times on the smartphone, tablet and traditional desktop computer," 28. September 2010.
http://marcial.web.unc.edu/files/2011/05/Marcial_lit_review_for_cmte.pdf.
- [23] H. Lockheimer, "Android and Security - Official Google Mobile Blog: VP of Engineering, Android," 2. Februar 2012. <http://googlemobile.blogspot.de/2012/02/android-and-security.html>.
- [24] Apple, "App Store Review Guidelines," 30. März 2012. <http://developer.apple.com/appstore/guidelines.html>.
- [25] Heib, A., "Ein Blick hinter die Mauer des App-Stores," 2010.
<http://www.tagesanzeiger.ch/digital/mobil/Ein-Blick-hinter-die-Mauer-des-AppStores/story/25463932>.
- [26] R. Fedler, C. Banse, C. Krauß, and V. Fusenig, "Android OS Security: Risks and Limitations," tech. rep., Fraunhofer-Einrichtung für Angewandte und Integrierte Sicherheit (Fraunhofer AISEC), Mai 2012.
<http://www.aisec.fraunhofer.de/content/dam/aisec/de/pdf/tech%20reports/AISEC-TR-2012-001-Android-OS-Security.pdf>.
- [27] T. Blaesing, L. Batyuk, A.-D. Schmidt, S. Camtepe, and S. Albayrak, "An Android Application Sandbox

- system for suspicious software detection,” in *Malicious and Unwanted Software, 2010 5th International Conference on*, pp. 55–62, IEEE, Oktober 2010.
- [28] Aalto University: AppCampus, “Microsoft and Nokia to invest up to 18 million euros in mobile application development program at Aalto University,” 26.03. 2012. <http://appcampus.aalto.fi/about>.
- [29] Google, “Android Developer Device Dashboard - Platform Versions,” 25.04. 2012. <http://developer.android.com/resources/dashboard/platform-versions.html>.
- [30] Valve Corporation, “Steam Hardware & Software Survey: März 2012,” 25.04. 2012. <http://store.steampowered.com/hwsurvey>.
- [31] “Softwarequalität - Prüfung nach DIN 12119 / EN 9241 | TÜV SÜD GRUPPE.” www.tuev-sued.de/ps/apps.
- [32] Arbeitskreis Mobile des eco - Verband der deutschen Internetwirtschaft e.V., “Expertenumfrage Mobile Applications,” 2010. http://mobile.eco.de/files/2011/04/Expertenumfrage_Mobile_Applications_22.pdf.
- [33] Apple, “iTunes Store - Bedingungen,” 13. April 2012. <http://www.apple.com/legal/itunes/de/terms.html>.
- [34] P. J. ZDNet, “Apple app prices rebound,” 13. März 2012. <http://www.zdnet.com.au/apple-app-prices-rebound-339318352.htm>.
- [35] C. Schmidt, *Digitaler Softwarevertrieb für mobile Endgeräte am Beispiel des Apple-App-Store*. GRIN Verlag, 26. Juli 2010.
- [36] V. Goncalves, N. Walravens, and P. Ballon, “„How about an App Store?” Enablers and Constraints in Platform Strategies for Mobile Network Operators,” in *Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR)*, pp. 66–73, IEEE, Juni 2010.
- [37] Statista (über Apple), “Statistik: Kumulierte Anzahl der weltweit heruntergeladenen Anwendungen aus dem Apple App Store in Milliarden (Stand: 03.März 2012).” <http://de.statista.com.eaccess.ub.tum.de/statistik/daten/studie/20149/umfrage/anzahl-der-ge-taetigten-downloads-aus-dem-apple-app-store/>.
- [38] Apple, “Apple Q&A on Location Data,” 27. April 2011. <http://www.apple.com/pr/library/2011/04/27Apple-Q-A-on-Location-Data.html>.
- [39] Statista (über Gartner), “Marktanteile der Betriebssysteme am weltweiten Absatz von Smartphones von 2009 bis 2011.” <http://de.statista.com.eaccess.ub.tum.de/statistik/daten/studie/12885/umfrage/marktanteil-bei-smartphones-nach-betriebssystem-weltweit-seit-2009/>.
- [40] comScore, “Android Captures #2 Ranking Among Smartphone Platforms in EU5,” 13. September 2011. http://www.comscore.com/Press_Events/Press_Releases/2011/9/Android_Captures_number_2_Ranking_Among_Smartphone_Platforms_in_EU5.
- [41] R. Kim, “Amazon’s Android Appstore, not so amazing,” 5. Juli 2011. <http://gigaom.com/2011/07/05/amazon-appstore-not-so-amazing/>.
- [42] Evans Data Corporation, April 2012.
- [43] Nokia, “Nokia builds new custom Ovi Store concept for Orange France and Deutsche Telekom,” 12. Mai 2011. <http://press.nokia.com/2011/05/12/nokia-builds-new-custom-ovi-store-concept-for-orange-france-and-deutsche-telekom/>.
- [44] E. A. (EA), “Electronic Arts Reports Q3 FY12 Financial Results,” 01. Februar 2012. <http://investor.ea.com/releasedetail.cfm?ReleaseID=644995>.
- [45] Ubuntu Wiki, “Softwarecenter,” 2012-03-30. <https://wiki.ubuntu.com/SoftwareCenter>.
- [46] Microsoft, “Discover apps in the Windows Store.” <http://windows.microsoft.com/en-US/windows-8/apps>.
- [47] GetJar, “About GetJar,” 30. März 2012. <http://www.getjar.com/about/>.
- [48] V. Tuunainen, T. Tuunainen, and J. Piispanen, “Mobile Service Platforms: Comparing Nokia OVI and Apple App Store with the IISn Model,” in *Mobile Business (ICMB), 2011 Tenth International Conference on*, pp. 74–83, IEEE, Juni 2011.
- [49] T. Yamakami, “A Three-Dimension Analysis of Driving Factors for Mobile Application Stores: Implications of Open Mobile Business Engineering,” in *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference*, pp. 885–889, IEEE, März 2011.
- [50] K. Kimbler, “App store strategies for service providers,” in *Intelligence in Next Generation Networks (ICIN), 2010*, pp. 1–5, IEEE, Oktober 2010.
- [51] O. Rugnon, M. Escudero, J. Rueda, and S. Shanmugalingam, “Toward Dynamic Business Models on marketplace environments,” in *Intelligence in Next Generation Networks (ICIN)*, pp. 1–6, IEEE, Oktober 2010.
- [52] J. Laugesen and Y. Yuan, “What Factors Contributed to the Success of Apple’s iPhone?,” in *Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR)*, pp. 91–99, Juni 2010. <http://ieeexplore.ieee.org.eaccess.ub.tum.de/stamp/stamp.jsp?tp=&arnumber=5494782>.
- [53] M. Chen and X. Liu, “Predicting popularity of online distributed applications: iTunes app store case analysis,” in *Proceedings of the 2011 iConference, iConference ’11*, (New York and NY and USA), pp. 661–663, ACM, 2011.
- [54] M. Cortimiglia, A. Ghezzi, and F. Renga, “Mobile Applications and Their Delivery Platforms,” *IT Professional*, vol. 13, no. 5, pp. 51–56, 2011.
- [55] M. Cusumano, “Platforms and services: understanding the resurgence of Apple: Combining new consumer devices and Internet platforms with online services and content is proving to be a successful strategy,” *Commun. ACM*, vol. 53, no. 10, pp. 22–24, 2010.

Smart Energy Grids

Konrad Pustka

Betreuer: Andreas Müller

Seminar Innovative Internettechnologien und Mobilkommunikation SS2012

Lehrstuhl Netzarchitekturen und Netzdienste

Fakultät für Informatik, Technische Universität München

Email: konrad.pustka@in.tum.de

KURZFASSUNG

Im Rahmen der Energiewende werden zunehmend erneuerbare Energien für die Stromerzeugung eingesetzt. Durch die dadurch steigende Volatilität der Stromquellen müssen neue Methoden für die Vernetzung, Verteilung und Speicherung des Stromes gefunden werden. Mit Hilfe sog. „Intelligenten Stromnetze“ (Smart Energy Grids) soll der Verbrauch an die Erzeugung angepasst werden. Die Anbindung privater Haushalte an das Smart Energy Grid erfolgt über sog. „Smart Meter“, welche den aktuellen Stromverbrauch an den Stromanbieter übermitteln und in Zukunft auch die Steuerung von Verbrauchern durch den Stromanbieter zulassen sollen. Großer Streitpunkt ist hier noch der Datenschutz und die Privatsphäre der Nutzer. Aber auch das Potenzial, durch Lastverlagerung Stromspitzen zu vermeiden, sowie den allgemeinen Verbrauch zu senken, gilt es noch zu untersuchen. Hier haben private Haushalte verschiedene Möglichkeiten. Einsparungen können allein durch Austausch ineffizienter Geräte und Abschaltung nicht benötigter Verbraucher erreicht werden. Aktuelle Studien untersuchen hierbei den Einfluss von Smart Metern auf das Nutzungsverhalten. Durch zukünftige Technologien zur Stromgewinnung und Speicherung werden sich auf diesem Gebiet weitere Möglichkeiten ergeben.

Schlüsselworte

Smart Energy Grid, Smart Meter, Einsparpotenzial

1. EINLEITUNG

Mit dem auf 2020 festgelegten Ausstieg aus der Atomenergie hat die Bundesregierung Deutschland den ersten Schritt in Richtung nachhaltiger Energieversorgung vollzogen. Der Anteil der erneuerbaren Energien am Gesamtstromverbrauch soll dabei von aktuell 20% (Wert für Deutschland im Jahr 2011) auf mind. 35% gesteigert werden [1]. Mit der Verlagerung hin zu regenerativen

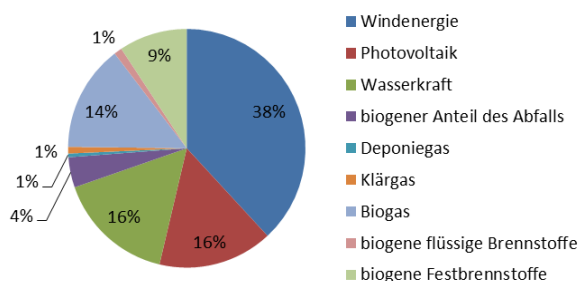


Abbildung 1. Struktur der Stromgewinnung aus erneuerbaren Energien in Deutschland (2011) [4]

Stromerzeugungsmethoden kommt allerdings, neben der technischen Umsetzung, eine weitere große Herausforderung auf die Bundesregierung zu. Stromerzeugung aus fossilen Energieträgern ist, außer von dem Material selbst, unabhängig von natürlichen Einflüssen. Viele der Stromerzeugungsmethoden aus erneuerbaren Energien beruhen allerdings auf Umwelteinflüssen, welche sich durch den Menschen nur schwer oder gar nicht beeinflussen lassen. Betrachtet man beispielsweise ausschließlich den Strom aus erneuerbaren Energien in Deutschland, so ist zu sehen, dass 2011 etwa 70% davon mit Wasserkraft, Windenergie oder Photovoltaik erzeugt wurde (siehe Abbildung 1). Je nach Wetterlage können diese Energie-Quellen stark schwanken. Eine weitere Veränderung des Stromnetzes wird durch die starke Verteilung der Energieerzeuger notwendig sein. Wird im Moment etwa 80% Prozent des Stroms in wenigen großen Kraftwerken erzeugt, so wird sich das im Hinblick auf regenerative Stromerzeuger auf viele kleinere Anbieter verteilen. Diese werden, wie schon beschrieben, zusätzlich auch noch größeren Schwankungen unterlegen sein.

Um diesen neuen Anforderungen gerecht zu werden, ist es notwendig den Aufbau des aktuellen Stromnetzes anzupassen. In Zukunft wird nicht nur die Kapazität eines Stromnetzes von Wichtigkeit sein, sondern vor allem wie der Strom verteilt und gespeichert wird. Zwischen Erzeuger, Speicher, Verbraucher und weiteren Netzkomponenten muss eine bidirektionale Kommunikation stattfinden, um die vorhandenen Ressourcen optimal verteilen zu können. Der Begriff „Smart Energy Grid“ (im Folgenden Smart Grid genannt) beschreibt dabei die Vision eines intelligenten Stromnetzes, welches all diese Faktoren beachtet und selbstständig die bestmögliche Verknüpfung der Netzteilnehmer herstellt. Die „Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE“ verwendet folgende Definition für Smart Grids:

„Der Begriff „Smart Grid“ (Intelligentes Energieversorgungssystem) umfasst die Vernetzung und Steuerung von intelligenten Erzeugern, Speichern, Verbrauchern und Netzbetriebsmitteln in Energieübertragungs- und Energieverteilungsnetzen mit Hilfe von Informations- und Kommunikationstechnik (IKT). Ziel ist auf Basis eines transparenten energie- und kosteneffizienten sowie sicheren und zuverlässigen Systembetriebs die nachhaltige und umweltverträgliche Sicherstellung der Energieversorgung.“ [3]

Viele Strom-Anbieter werben mit diesem Begriff auch für Strom- und Kosteneinsparungen in privaten Haushalten. Inwieweit dies zutrifft und in welchen Bereichen der private Endverbraucher tatsächlich zu einer besseren Lastverteilung beitragen kann, soll in dieser Arbeit dargestellt werden.

doi: 10.2313/NET-2012-08-1_09

Im folgenden Abschnitt 2 wird zunächst der Aufbau eines Smart Grids genauer definiert. Dabei wird sowohl auf die einzelnen Komponenten, als auch auf die Kommunikationswege und Sicherheitsaspekte eingegangen. Abschnitt 3 befasst sich mit den Einsparpotenzialen in privaten Haushalten. Verschiedene Möglichkeiten der optimierten Stromnutzung werden aufgezeigt und auf ihren Nutzen untersucht. Zum Abschluss wird in Abschnitt 4 eine Zusammenfassung über die Ergebnisse dieser Arbeit gegeben.

2. AUFBAU EINES SMART GRIDS

Das Stromnetz in Deutschland ist in mehrere Ebenen untergliedert: Höchst-, Hoch-, Mittel- und Niederspannungsnetz [2]. Das Höchstspannungsnetz ist für den Transport von Strom über große Strecken zuständig. In diesem Netz speisen Großkraftwerke ein. Kleinere Stromerzeugungsanlagen dagegen speisen in das Hoch- oder Mittelspannungsnetz ein. Dies sind folglich die Netze, die für Smart Grids von Bedeutung sind.

2.1 Teilnehmer

Ein Smart Grid umfasst ein großes Areal an verschiedenen Teilnehmern mit jeweils eigenen Interessen. Ein Überblick über die Verknüpfung der Teilnehmer sowie den Strom- und Datenfluss ist in Abbildung 2 dargestellt.

Auf der einen Seite stehen die Stromerzeuger. Dies können konventionelle Kraftwerke, Biomasse-Verbrennungs-Anlagen, Windparks oder andere auf erneuerbaren Energien basierende Kraftwerke sein. Die Erzeugung muss dabei bestmöglich an den Verbrauch angepasst werden.

Um den überschüssigen Strom „konservieren“ zu können werden Speicherkraftwerke benötigt. Die gängigste Art ist das Pumpspeicherkraftwerk. Wenn Strom verfügbar ist wird dort Wasser in einen höhergelegenen Speicher gepumpt, um später, mit Hilfe des Wasserdrucks, Turbinen und Generatoren antreiben zu können. Der Wirkungsgrad liegt dabei bei 75-80% [5].

Eine große Bedeutung kommt auch den Netzbetreibern zu. Durch die direkte Schnittstelle zu Erzeugern und Verbrauchern können diese den Stromfluss im Netz optimal anpassen. Sie sind das Bindeglied zwischen Strom-Erzeuger und Verbraucher.

Am Ende steht der Verbraucher. Durch die größeren

Schwankungen in der Stromerzeugung muss dieser durch geschickte Steuerung der Endgeräte seinen Verbrauch an das Angebot anpassen. Besonders Lastspitzen, welche im Moment morgen, mittags und abends auftreten gilt es zu vermeiden [6]. Welche Möglichkeiten er dabei hat wird im nächsten Kapitel dieser Arbeit aufgezeigt.

Auf den Verbraucher kommt darüber hinaus noch eine komplett neue Aufgabe zu. Über eigene Stromerzeugungsanlagen (z.B. eine Photovoltaik-Anlage auf dem Hausdach) und Stromspeicher (z.B. einen „Haus-Akku“) kann Strom bei Bedarf in das Netz eingespeist werden. Gerade der Bereich der privaten Stromspeicher wird in Zukunft immer wichtiger werden, eine große Bedeutung wird dabei auch den Elektro-Autos zukommen. Der Verbraucher wird somit vom reinen Strom-Konsumenten auch zum Strom-Produzenten. Er wird deswegen auch als „Prosumer“ bezeichnet. Der „Prosumer“ hat über Strommärkte die Möglichkeit aktiv am Energiehandel teilzunehmen.

2.2 Smart Meter

Eine wichtige Voraussetzung für die neuen Aufgaben des Endverbrauchers ist der „Smart Meter“. Beschrieben wird dadurch ein allgemeiner Zähler für Energie (es besteht keine Beschränkung auf Strom), welcher über eine Kommunikationsschnittstelle den tatsächlichen Verbrauch und die Nutzungszeit auslesbar macht. Je nach Modell sind über den Smart Meter auch weitere Funktionen, wie z.B. die Steuerung elektrischer Geräte, vorhanden. In Deutschland müssen Smart Meter seit Januar 2010 bei Neubauten, größeren Renovierungen und Verbrauchern mit einem Jahresverbrauch von mehr als 6.000 kWh eingebaut werden (vgl. §21c Abs. 1 EnGW [7]). Die Definition eines Smart Meters beschränkt sich dabei allerdings auf die Grundfunktionalität des Anzeigens des Verbrauchs und der Nutzungszeit (vgl. §21d Abs. 1 EnGW [8]). In den meisten Fällen ist es dem Endverbraucher und auch dem Stromanbieter somit nicht möglich den Stromverbrauch über eine Datenschnittstelle abzurufen. Die nächste Generation der Smart Meter soll 2013 eingeführt werden [9].

2.3 Kommunikationsnetze

Um all die bisher beschriebenen Teilnehmer miteinander zu vernetzen ist es notwendig neue Kommunikationsnetze zu

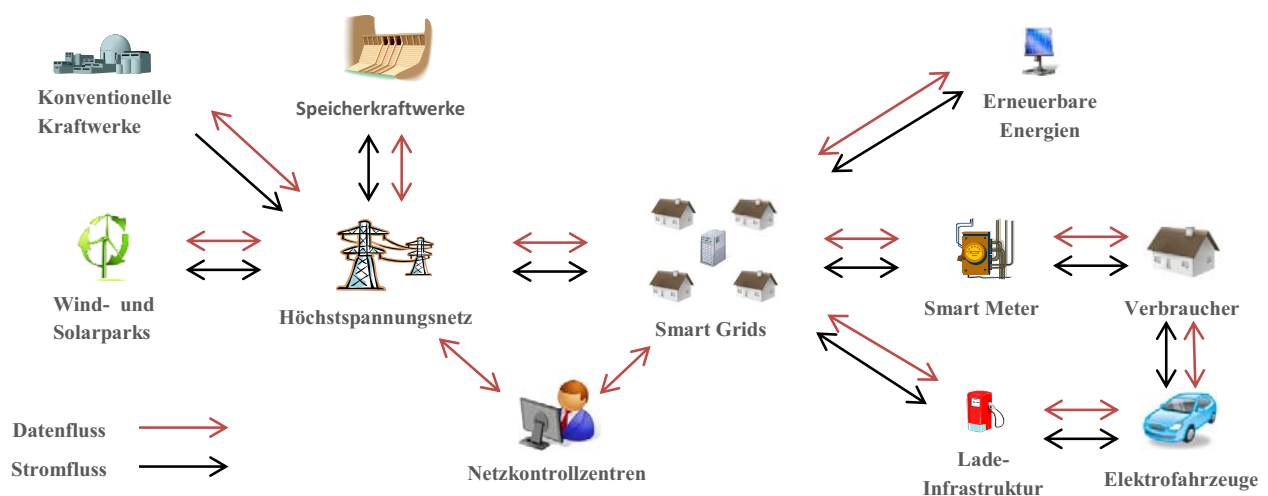


Abbildung 2. Strom- und Datenfluss im Smart Grid [2]

errichten. Für die Kommunikation zwischen Smart Metern und Netzbetreibern sind verschiedene Übertragungswege in Erprobung. Durch die direkte Verbindung des Smart Meters mit der Stromleitung bietet sich das „Power Line Communication“ (PLC) Verfahren an, bei dem die Daten direkt über das Stromnetz gesendet werden. Unklar ist aber noch, ob es den neuen Anforderungen gerecht werden kann [10]. Im Rahmen des E-Energy Förderprogramms werden deutschlandweit in sechs verschiedenen Modellregionen weitere Kommunikationssysteme, wie die Übertragung über das Handynet oder Wide-Area-Networks (WAN) getestet. Welches Verfahren sich durchsetzen wird ist allerdings noch nicht absehbar. Durch die verschiedenen Vor- und Nachteile ist davon auszugehen, dass jedes Verfahren in Abhängigkeit von den örtlichen Gegebenheiten benötigt wird.

Hierfür sind auch protokollübergreifende Standards und Normen notwendig, um die genaue Implementierung möglichst flexibel zu halten. Derzeit wird weltweit an solchen Standards gearbeitet. Deutschlandweit arbeiten Gremien des VDE daran die Standardisierung von Smart Grids voranzutreiben, dies geschieht in enger Zusammenarbeit mit der europäischen Normung und auf internationaler Ebene mit dem IEC. Im Rahmen der „Deutschen Normungsroadmap“ wurden dabei, die in Tabelle 1 dargestellten Kernstandards definiert [3]. Besonders relevant für den Bereich des privaten Verbrauchers sind dabei das Common Interface Model (CIM) sowie der IEC 61850 Standard. Das Common Interface Model teilt sämtliche Übertragungsdaten in Klassen und Attribute auf. Es ist also eine objektorientierte Beschreibung der Messwerte.

Tabelle 1. Kernstandards nach der Deutschen Normungsroadmap[3]

| Kernstandards | Thema |
|-----------------|---|
| IEC 62357 | Seamless Integration Reference Architecture (SIA) |
| IEC 61970/61968 | Common Interface Model (CIM) |
| IEC 61850 | Substation Automation, Distributed Energy Resources |
| IEC 62351 | Security |

Im Standard „IEC 91850“ ist die Verwendung von Ethernet-Kommunikation und Internetprotokollen festgeschrieben. Des Weiteren enthält er Angaben über eine normierte Geräte- und Systembeschreibungssprache.

2.4 Sicherheitsaspekte

Beim Austausch von Informationen über Datennetze besteht immer die Gefahr des Auslesens oder Manipulierens der Daten durch Dritte. In IEC 62351 wurden deswegen schon frühzeitig Vorgaben für die Verschlüsselung, Authentifizierung, sowie die Erkennung von Manipulationen gemacht.

Ein weiterer häufig diskutierter Punkt, betreffend Smart Meter, ist die Privatsphäre der Endnutzer. Kennt der Netzbetreiber den genauen Stromverbrauch eines Kunden, so kann er über das Lastprofil des Haushalts genaue Angaben über die Gewohnheiten der Bewohner machen. Wann aufgestanden, geduscht und gefrühstückt wird, lässt sich einfach ablesen. Forscher der Fachhochschule Münster haben diese Problematik genauer untersucht und konnten anhand der von einem Smart Meter

übertragenen Daten sogar feststellen, welches Fernsehprogramm der Kunde gerade ansieht [11]. Dieses Ergebnis gilt es allerdings richtig einzuordnen, denn in dem Versuch wurde ein sekundliches Übertragungsintervall gewählt. Solch eine Granularität wird mit hoher Wahrscheinlichkeit nicht notwendig, wenn nicht sogar nicht umzusetzen sein. In welchem Intervall jedoch letztendlich übertragen wird, ist noch offen. Im Moment wird in den meisten Fällen eine Viertelstunde gewählt.

Um die Privatsphäre weiter zu schützen, gibt es verschiedene Ansätze [18]. Zum einen könnte man den aktuellen Verbrauch „verschleiern“. Hierbei kommt ein Akku zum Einsatz, der zu Stoßzeiten Strom abgibt und bei allgemein niedrigem Verbrauch wieder aufgeladen wird. Hierdurch werden nicht nur Stromspitzen vermieden, der reale Verbrauch durch Verbraucher im Haushalt wird so auch „überdeckt“ und Rückschlüsse auf einzelne Geräte können nur noch sehr schwer getroffen werden.

Ein weiterer Ansatz ist das Übertragen rein statistischer Daten. Der Stromverbraucher würde so nicht mehr den Verbrauch eines einzelnen Haushalts, sondern nur noch den Gesamtverbrauch einer Gruppe von Häusern übermitteln bekommen. Dies würde ausreichen, um den Stromfluss entsprechend anpassen zu können, ohne dass genaue Details über einzelne Haushalte verfügbar wären. Die Abrechnung der Stromkosten würde über einen getrennten Weg, anhand über den Monat aggregierter Daten, stattfinden.

3. EINSPARPOTENZIALE

Mit dem Wissen über Smart Grids, welche Ziele verfolgt werden, und wie die Struktur um einen Haushalt mit Smart Meter aussieht, soll nun das Einsparpotenzial durch Smart Grids untersucht werden. Grundsätzlich unterscheidet man zwischen investiven und nicht-investiven Maßnahmen. Investive Maßnahmen beschreiben dabei bauliche Veränderungen (z.B. durch Einbau besser gedämmter Fenster oder einer effizienteren Heizung), welche meist mit einem größeren finanziellen Aufwand verbunden sind. Nicht-investive Maßnahmen dagegen sind Einsparungen, welche hauptsächlich durch angepasstes Nutzungsverhalten erreicht werden. Aufgrund der größeren Bedeutung für Smart Grids werden im folgenden Kapitel allein die nicht-investiven Maßnahmen betrachtet.

Dabei wird zunächst ein Überblick über die Verbraucher im Haushalt gegeben, um anschließend die verschiedenen Möglichkeiten des Stromsparens und der Kostensenkung zu überprüfen.

3.1 Verbraucher im Haushalt

In einem modernen Haushalt befindet sich eine Vielzahl an elektrischen Verbrauchern. Um Einsparpotenziale bestimmen zu können muss zunächst der Gesamtverbrauch der einzelnen Geräte betrachtet werden. Tabelle 2 zeigt eine Auflistung ausgewählter Geräte im Haushalt, mit durchschnittlichem Jahresverbrauch [12][13]. Die jährlichen Kosten wurden anhand eines Strompreises von 22 Cent/kWh geschätzt.

In der Tabelle wird deutlich, dass Geräte wie Waschmaschine, Trockner, Spülmaschine, etc. (sog. „Weiße Ware“) absolut gesehen den höchsten Verbrauch haben. Aber auch die weniger „hungrigen“ Verbraucher wie einfache Lampen tragen in der

| Gerät | Leist. [W] | Std. [h] | El. Arbeit [kWh] | Kosten [€] |
|---------------------------------------|---------------|-------------|------------------------|---------------|
| Waschmaschine | 450 | 300 | 135 | 29,70 |
| Trockner | 1410 | 158 | 223 | 49,06 |
| Spülmaschine | 550 | 436 | 240 | 52,80 |
| Kühlschrank | 140 | 2920 | 409 | 89,98 |
| Gefrierschrank | 142 | 2920 | 415 | 91,30 |
| Hi-Fi-Anlage mit Plasma-TV | 429 | 730 | 313,2 | 68,90 |
| Deckenfluter 300W Halogen | 210 | 365 | 76,7 | 16,86 |
| Lampenzeile mit 3 x 20W Halogen | 51 | 1460 | 74,5 | 16,38 |
| Nachtlicht 15W Glühlampe | 15 | 4380 | 65,7 | 14,45 |
| PC-System | 122 | 365 | 44,5 | 9,80 |

Tabelle 2. Jährlicher Verbrauch und Kosten ausgewählter Haushaltsgeräte [12][13]

Summe einen großen Teil zu den jährlichen Kosten eines Haushalts bei.

3.2 Einsparung durch Austausch ineffizienter Geräte

Eine einfache Möglichkeit Strom zu sparen besteht darin bestehende Geräte durch Geräte mit äquivalentem Nutzen, aber höherer Energieeffizienz auszutauschen. Ein Beispiel aus der oben genannten Liste ist der 300 W Halogen Deckenfluter. Der jährliche Verbrauch von 76,7 kWh ließe sich durch den Austausch mit einer vergleichbar hellen, 24 W ESL Energiesparlampe, auf fast ein Zehntel (8,8 kWh) reduzieren [13].

Bei vielen Geräten ist aber nicht nur der Stromverbrauch in angeschaltetem Zustand von Bedeutung, sondern auch der Verbrauch im Standby. Allein manch eine ältere Kompakt-Hi-Fi-Anlage hat im Standby immer noch eine Leistung von 10 W. Über das Jahr gesehen summiert sich das zu einem Verbrauch von 84 kWh auf, das entspricht nach obiger Kalkulation 18,47 € [13].

Heutige Geräte erlauben solch hohe Standby-Verbrauchswerte nicht mehr. Laut der EU Richtlinie 2009/125/EG Lot 6 dürfen neue Geräte seit 7.1.2010 im Standby-Betrieb nur noch 1 W verbrauchen. Zeigt das Gerät auch im Standby einen Status an (z.B. Uhrzeit oder Timer), so erhöht sich der Wert auf 2 W. Diese Werte werden ab dem 7.1.2013 auf 0,5 W bzw. 1 W reduziert.

Allein durch den Austausch älterer Geräte lässt sich also schon heute viel Strom und Geld sparen. Laut einer Studie des Bund Naturschutz Bayern liegt das Einsparpotenzial dadurch pro Haushalt bei durchschnittlich 36 % [14]. Details zu dieser Studie sind allerdings nicht bekannt.

Dieser Trend zu immer effizienteren Geräten wird auch in Zukunft weitergehen. Aktuelle Prognosen gehen davon aus, dass die Energie-Effizienz von elektrischen Geräten bis 2025 um 20 % steigen wird [12].

3.3 Einsparung durch Abschaltung nicht benötigter Verbraucher

Eine weitere einfache Möglichkeit des Stromsparens besteht darin, ungenutzte Verbraucher abzuschalten. Zum einen können dies Geräte sein, welche auch im Standby noch eine hohe Leistungsaufnahme haben. Wie im vorherigen Abschnitt beschrieben sind diese allerdings schon vom Markt verschwunden und auch, sich noch in Benutzung befindende Geräte, werden früher oder später ausgemustert werden.

Aber auch Lichter und Fernseher werden gerne angelassen, wenn man den Raum verlässt. Die schon im vorherigen Abschnitt genannte Studie des Bund Naturschutz geht hierbei von einem Sparpotenzial von 4 % aus [14].

3.4 Einsparung durch Feedback

Smart Meter können die bisher genannten Arten des Stromsparens zwar nicht direkt beeinflussen, aber eine Studie des Intelliekon Projekts im Rahmen des E-Energy Programms hat gezeigt, dass allein durch das Konfrontieren der Nutzer mit dem Thema „Energiesparen“ und der Hilfe durch Verbrauchsanalyse und Energiespartipps, eine Einsparung von 3,7 % stattgefunden hat [15]. Untersucht wurden über 2000 Haushalte in Deutschland und Österreich. Die Verbrauchs-Anzeige durch einen Smart Meter gibt dabei den Anreiz sich mit dem Thema auseinander zu setzen und ermöglicht das selbstständige Evaluieren der getroffenen Maßnahmen. Dieses Feedback wurde im Intelliekon-Projekt über ein Web-Portal, sowie monatliche, schriftliche Verbrauchsinformationen gegeben. Die Testpersonen favorisierten dabei das Web-Portal. Inwieweit die Einsparung durch Abschalten oder durch Austauschen von Geräten zustande kam wurde jedoch nicht untersucht. Im Vergleich zu der Potenzialanalyse durch den Bund Naturschutz sind diese Ergebnisse aber ernüchternd. Ältere Studien zwischen 2004 und 2010 kamen zu dem Ergebnis, dass durch das Anzeigen von Strom-Feedback Einsparungen zwischen 2 % und 10 % möglich sind [15]. Fasst man die Ergebnisse dieser Studien zusammen, so lassen sich drei Eigenschaften eines erfolgreichen Feedback-Systems feststellen, welche besonders gut durch Smart Metering unterstützt werden können. Dies sind [19]:

- Feedback basiert auf derzeitigem Verbrauch
- Feedback wird regelmäßig und häufig gegeben
- Interaktivität des Feedback-Systems

3.5 Einsparung durch Lastverteilung

Mit der steigenden Abhängigkeit von erneuerbaren Energien wird es notwendig werden Strom genau dann zu nutzen, wenn er verfügbar ist. Die Netzbetreiber erhoffen sich hier, auch in Privathaushalten Lastverschiebungen realisieren zu können. Das Verteilen des Stromes anhand der aktuellen Verfügbarkeit wird auch Demand Side Management (DSM) genannt.

Eine der wichtigsten Fragen hierbei ist inwiefern der Nutzer bereit ist seine elektrischen Verbraucher in einem Haushalt dem Stromangebot anzupassen. Ist bei dieser Verlagerung mit größerem Komfortverlust zu rechnen, so ist es unwahrscheinlich, dass sie angenommen wird. Es ist schließlich schwer vorstellbar, dass ein Mensch sein Abendessen um zwei Stunden verschiebt, oder das Licht zunächst auslässt, nur um Kosten zu sparen. Bei anderen Geräten, wie z.B. der Spülmaschine oder der Waschmaschine scheint variable Benutzung wahrscheinlicher. Wie groß das Potenzial der Lastverteilung ist soll im folgenden

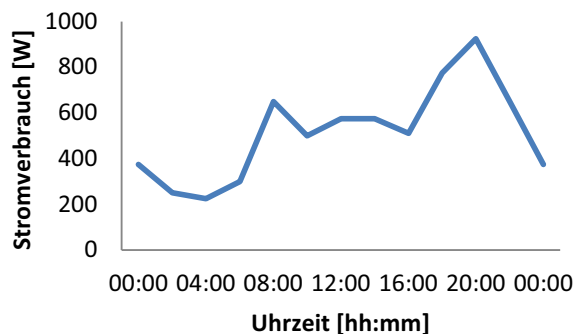


Abbildung 3. Durchschnittlicher Lastverlauf in der Modellstadt Mannheim [6]

Abschnitt anhand von bisher durchgeführten Pilotprojekten erläutert werden.

Allein, den Nutzer zu informieren, wann er mehr und wann weniger Strom nutzen sollte, reicht dafür nicht aus. Erste Ergebnisse in der Modellstadt Mannheim (moma) haben gezeigt, dass die Hauptmotivation der Haushalte, das Verhalten zu ändern, in finanziellen Vorteilen liegt [6]. Abbildung 3 zeigt die durchschnittliche Lastverteilung über einen Monat, wie sie im moma Projekt gemessen wurde. Lastspitzen sind morgens, mittags sowie abends zu erkennen. Um diese Spitzen zu vermeiden wurden in den Modellregionen des E-Energy Programms Anreize durch variable Tarife geschaffen. Teil des Intelliekon Projekts war ein sogenannter „Happy Hour“ Tarif, bei dem der Strompreis zwischen 10 und 18 Uhr von 16,8 ct / kWh auf 30,2 ct / kWh erhöht wurde. Über ein Webportal konnten die Nutzer ihren Stromverbrauch in den einzelnen Tarifzonen nachvollziehen. Die Auswertung des Tests ergab, dass durch diesen Tarif 2 % der Gesamtlast in einen Niedertarif-Bereich verschoben wurde. Ein Nebeneffekt war eine erhöhte Stromersparnis im Vergleich zu den Nutzern ohne variablen Tarif. So wurde der Gesamtverbrauch um 6 % gesenkt, immerhin 2,3 % mehr als bei den Nutzern eines konstanten Tarifs. Eine genaue Erklärung hierfür konnte noch nicht gegeben werden. Es wird vermutet, dass das tägliche Auseinandersetzen mit dem Strompreis dieses verbesserte Sparverhalten bewirkt.

Auch das moma Projekt untersucht das Verhalten der Nutzer im Kontext variabler Stromtarife. In diesem Projekt werden die Stromtarife genau den bekannten Stromspitzen angepasst. Niedertarif-Zeiten sind zwischen 21 und 5 Uhr, 9 und 11 Uhr, sowie 14 und 16 Uhr angesetzt. Die genauen Zeiten können jeweils einen Tag im Voraus ebenfalls über ein Webportal abgerufen werden. Zwischenergebnisse konnten hier eine Lastverschiebung von 8 – 10 % feststellen. Der Gesamtverbrauch wurde allerdings nur um 2 % reduziert. Eine grafische Darstellung der Lasten mit und ohne variablen Stromtarif ist in Abbildung 4 dargestellt.

Vergleicht man die unterschiedlichen Ergebnisse der beiden Projekte miteinander, so muss beachtet werden, dass im moma Projekt eine feinere Abstufung der Tarife stattgefunden hat, was den Teilnehmern bessere „Ausweichmöglichkeiten“ gegeben hat. Besonders der abendliche Stromverbrauch lässt sich so weiter in Richtung Mitternacht verschieben, was wie gesehen, eine deutliche Abstumpfung der größten Lastspitze zwischen 19 und 21 Uhr zur Folge hat.

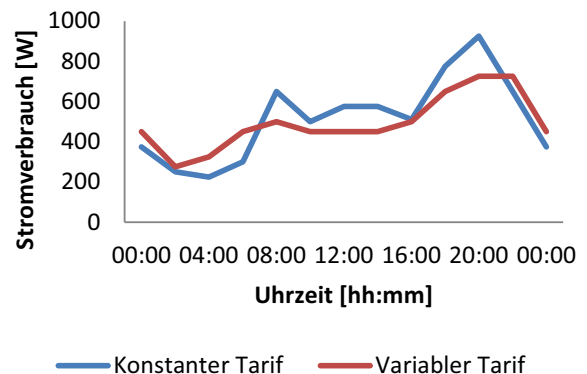


Abbildung 4. Lastverlauf in der Modellstadt Mannheim in Abhängigkeit des Stromtarifs[6]

In diesen beiden Pilotversuchen mussten die Teilnehmer ihre Haushaltsgeräte selbstständig an- und abschalten. Die nächste Stufe der Lastverschiebung, wie sie in Smart Grids vorgesehen wird, ist das Steuern von „Weißer Ware“ durch den Netzbetreiber. Pilotversuche hierzu sind noch im Gange, so wird dies beispielsweise in der letzten Phase des moma Projekts getestet. Eine Studie zur Akzeptanz solcher Geräte in der Bevölkerung ergab, dass in Abhängigkeit des Funktion (Waschmaschine, Trockner, etc.) etwa 70 – 80 % der Testpersonen sich vorstellen können eine Automatisierung zu benutzen [16]. Die Studie basiert allerdings allein auf Umfragen und die Teilnehmer waren gebildete Menschen, welche schon ein starkes Bewusstsein für Umweltschutz haben. Wie hoch die Akzeptanz in der breiten Bevölkerung ausfallen wird ist noch offen. Einen elektrischen Boiler je nach Stromangebot zu unterschiedlichen Zeiten aufzuheizen ist noch leicht vorstellbar, aber schon eine Waschmaschine nachts laufen zu lassen, ist nicht jedermanns Sache. Einerseits stellt die Waschmaschine eine Lärmbelästigung dar, andererseits würde die Wäsche nach einem nächtlichen Waschgang eventuell mehrere Stunden lang feucht in der Waschmaschine liegen bleiben. Hier sind die Hersteller von Haushaltsgeräten gefragt, um diesen Bereich mit neuen Innovationen abzudecken.

Eine weitere technische Herausforderung ist die Kommunikation zwischen Geräten und Smart Meter. Viele Hersteller und Stromanbieter verwenden ihre eigenen Protokolle. Hier muss ein einheitliches Interface geschaffen werden, um die Vernetzung zu ermöglichen.

Eine Entwicklung in diese Richtung ist der EEBus [20]. Er wurde ebenfalls im Rahmen des E-Energy Programms entwickelt und beschreibt eine Technologie um die verschiedenen Geräte im Haushalt miteinander zu verknüpfen und eine einheitliche Schnittstelle zum Netzbetreiber zur Verfügung zu stellen.

Ein ähnliches Projekt ist das, ehemals EU geförderte, Hydra-Projekt. Entwickelt wird, mittlerweile unter dem Namen „LinkSmart“, ebenfalls eine Middleware zur Verknüpfung netzwerkbasierter Systeme. Die verschiedenen Geräte lassen sich dabei über eine Web-Service Schnittstelle ansprechen. Eine erste Open-Source Referenz-Implementierung ist angekündigt worden.

3.6 Ausblick

Die bisherigen Projekte sind immer von dem aktuellen Stand der Technik ausgegangen. In Zukunft werden sich im Bereich der

Stromerzeugung und -speicherung jedoch weitere Möglichkeiten aufzun.

Im Falle der Erzeugung wird zurzeit in viele Richtungen geforscht, um auch den privaten Haushalt einzubinden. Durch Photovoltaik-Anlagen wird auch heute schon von vielen Privatleuten eingespeist, aber auch hauseigene Generatoren können hier eine Rolle spielen. Ein Beispiel hierfür ist das „LichtBlick-ZuhauseKraftwerk“ [17]. LichtBlick verwendet hier einen, beim Kunden installierten, VW Gas-Verbrennungsmotor. Dieser kann, von außen gesteuert, bei Bedarf Strom erzeugen und in das Netz einspeisen. Die dabei entstehende Wärme wird zum Aufheizen des häuslichen Wassers genutzt.

Größere Bedeutung wird allerdings die Speicherung von Energie haben. Um die Schwankungen von Wind- und Sonnenenergie ausgleichen zu können muss jede Möglichkeit, den im Überschuss produzierten Strom zu speichern genutzt werden.

Eine bedeutende Rolle kommt dabei dem Elektroauto zu. Die erzeugte Energie wird so nicht nur CO₂-effizient genutzt, im Hinblick auf Smart Grids ergibt sich dadurch auch eine neue Art von Energiespeicher. So können die Autos bei Stromüberfluss geladen werden und bei Strommangel auch wieder in das Netz einspeisen. Entsprechende „Parkplätze“ hierfür sind in der Test- und Entwicklungsphase, aber auch die Batterie-Technik muss für diesen Zweck noch weiter entwickelt werden.

Beispiele für weitere Ideen sind hauseigene Pumpspeicherkraftwerke oder Akkumulatoren. Auch Nachtspeicherheizungen werden im Rahmen der Energiespeicherung an Bedeutung gewinnen.

4. ZUSAMMENFASSUNG

Der beschlossene Weg der Energiewende bringt einige neue Herausforderungen mit sich. Auch die Informations- und Kommunikationstechnologie (IKT) wird dabei eine große Rolle spielen. Smart Grids bringen verschiedene Teilnehmer mit unterschiedlichen Interessen zusammen. Internationale und nationale Gremien sind hier gerade dabei geeignete Standards und Normen zu entwerfen. Viele der geplanten Konzepte sind gerade erst in der Test- und Entwicklungsphase, so dass hier noch ein gemeinsamer Nenner gefunden werden muss.

Mit dem verpflichtenden Einbau von Smart Metern in Neubauten und bei größeren Renovierungen hat die Bundesregierung Deutschland schon einen ersten Schritt zur Anbindung der privaten Haushalte an das Smart Grid gemacht, allerdings sind die aktuellen Mindestanforderungen an ein Smart Meter noch zu gering um auch für die Zukunft gerüstet zu sein.

Diskutiert wird auch noch der Schutz der Privatsphäre der Nutzer, da über ein Smart Meter Daten gesendet werden, welche sich zum Anlegen eines genauen Nutzerprofils eignen. So lässt sich im extremsten Fall über die Stromnutzung sagen, welches Fernsehprogramm der Nutzer gerade ansieht.

Das Einsparpotenzial durch Smart Grids bezüglich Stromverbrauch und -kosten lässt sich in mehrere Teilbereiche aufteilen. Austausch von „Stromfressern“ durch neuere stromsparende Geräte, sowie einfaches Abschalten nicht genutzter Verbraucher kann zwar nach dem Bund Naturschutz sehr große Einsparungen bringen, Feldtests im Rahmen des E-Energy Projekts beobachteten allerdings nur Stromeinsparungen zwischen 2 und 4 %. Auch eine Einsparung von nur 3,7 % pro Haushalt würde allerdings für den deutschen Stromverbrauch eine

Reduzierung um 5 TWh pro Jahr bedeuten. Dies entspricht etwa 1 Mrd. Euro.

Im Hinblick auf volatile Energiequellen, wie Wind und Sonne, wird auch der private Haushalt seinen Verbrauch an das aktuelle Angebot anpassen müssen. Anreize für die Lastverschiebung müssen durch flexible Tarife gegeben werden. Aktuelle Pilottests, bei denen über ein Webportal die Stromkosten des kommenden Tages angezeigt wurden, konnten dadurch Lastverschiebungen zwischen 2 – 10 % feststellen. Eine weitere Möglichkeit diese Verschiebung zu unterstützen, ist die Steuerung von „Weißer Ware“ (z.B. Geschirrspüler, Waschmaschine, etc.) durch den Netzbetreiber. Pilotprojekte hierzu sind noch im Gange, so dass sich noch keine bestätigten Aussagen dazu machen lassen.

Insgesamt lässt sich feststellen, dass durch Smart Grids im Bereich privater Haushalte keine Wunder zu erwarten sind, die bisher getesteten Methoden aber schon einen guten Beitrag zu der Energiewende beisteuern können.

Dennoch wird, um den Umschwung auf regenerative Energien schaffen zu können noch viel Forschung, vor allem im Bereich der Stromspeicherung notwendig sein.

5. REFERENZEN

- [1] „Energiekonzept für umweltschonende, zuverlässige und bezahlbare Energieversorgung“, BMWi, BMU, September 2010
- [2] R. Höfer-Zygan, E. Oswald, M. Heidrich, „Smart Grid Communications 2020“, 2011
- [3] „Deutsche Normungsroadmap E-Energy / Smart Grid“, VDE, DKE, Mai 2010
- [4] „Erneuerbare Energien in Deutschland, Das Wichtigste im Jahr 2011 auf einen Blick“, BMU, März 2012
- [5] M. Popp, „Speicherbedarf bei einer Stromversorgung mit erneuerbaren Energien“, Springer-Verlag, Berlin, Heidelberg, 2010
- [6] „Ergebnisse der Abschlussbefragung im Praxistest 2“, Modellstadt-Mannheim, März 2012
- [7] „§21c EnWG“, Juris, 2005
- [8] „§21d EnWG“, Juris, 2005
- [9] „Intelligente Zähler“, dena, Dezember 2011
- [10] M. Bauer, M. Sigle, K. Dostert, „Evaluation von PLC-Übertragungssystemen für Smart Metering“, Oldenbourg Wissenschaftsverlag, Oktober 2010
- [11] U. Greveler, B. Justus, D. Löhr, „Hintergrund und experimentelle Ergebnisse zum Thema Smart Meter und Datenschutz“, Technischer Report, September 2011
- [12] D. Seebach, C. Timpe, D. Bauknecht, „Costs and Benefits of Smart Appliances in Europe“, Öko-Institut e.V., September 2009
- [13] E. Ahlers, „Schalt mal ab“, Zeitschriftenartikel, c't Magazin, Heft 11 / 2011, Heise Zeitschriften Verlag
- [14] „Stromsparen in Bayern: minus 40 Prozent, minus 30 Milliarden Kilowattstunden, minus 7 Milliarden Euro Kosten“, Pressemitteilung, Bund Naturschutz Bayern, März 2012
- [15] „Nachhaltiger Energiekonsum von Haushalten durch intelligente Zähler-, Kommunikations- und Tarifsysteme,

Ergebnisbericht – November 2011”, Fraunhofer-Institut für Solare Energiesysteme, November 2011

[16] W. Mert, J. Suschek-Berger, W. Tritthart, „Consumer acceptance of smart appliances“, Report, Dezember 2008

[17] LichtBlickAG, „LichtBlick-ZuhauseKraftwerk“, http://www.lichtblick.de/h/ZuhauseKraftwerk_310.php, Abgerufen am 20. Juli 2012

[18] Georgios Kalogridis, „Smart Grid Privacy Protection by Design“, April 2011

[19] Corinna Fischer, „Influencing Electricity Consumption via Consumer Feedback. A Review of Experience“, June 2007

[20] Wolfgang Dorst, Til Landwehrmann, „EEBus: Whitepaper“, Mai 2011

The Evolution of Avionics Networks From ARINC 429 to AFDX

Christian M. Fuchs

Advisors: Stefan Schnee, Alexander Klein

Seminar Aerospace Networks SS2012

Chair for Network Architectures and Services

Faculty of Informatics, Technical University of Munich

Email: christian.fuchs@tum.de

ABSTRACT

Signaling and inter-system communication in avionics have been crucial topics ever since electronic devices were first used in aerospace systems. To deal with the challenges introduced by the widespread use of general purpose computing in commercial avionics, standards like ARINC 419 and later on 429 were published and adopted by the industry. While in industrial use, 429 has been adapted and extended very little since the standard was formulated in the late 1970s. 429 today cannot meet challenges and new requirements generated by the use of Integrated Modular Avionics and flexible system design. AFDX combines proven safety and availability functionality with modern Ethernet technology to be able to handle today's requirements. This paper outlines two of the most important avionics network architectures and aims at depicting the evolution of networking concepts and requirements over the course of the past 30 years. It mainly focuses on ARINC 429 and AFDX, the most prominent current and past standards, but also covers two other interesting past protocols.

Keywords

AFDX, ARINC 664, ARINC 429, Ethernet, MIL-STD-1553, avionics, fault tolerance, security, safety

1. INTRODUCTION

Signaling and inter-system communication in avionics have been a crucial topic ever since electronic devices were first used in aircraft. Initially, simple sensory feedback and components like radar and engines needed to be interconnected with cockpit controls. As time progressed, more and more systems which produce and consume data were introduced in avionics, at some point becoming crucial for even the most essential tasks, such as steering and later fly-by-wire. To deal with these challenges in commercial avionics, standards like ARINC 419 (and later on 429) were drafted and adopted not just by individual corporations, but collectively by almost the entire industry [1, 2].

Today, ARINC 429 can be found in most active and retired aircraft series. While it is well-established in the industry, it has been adapted and extended little since the initial specifications were formulated in the late 1970s. In contrast to avionics standards, multiple technological revolutions have happened in the computer industry at a fast pace. Networking of computers aboard aircraft may have been unthinkable in 1970, while modern aircraft without any networked com-

puters are very uncommon. Legacy avionics communication standards still reflect past views on computing [1, 3].

Ultimately, a modern networking architecture for avionics use should offer a maximum of safety, redundancy and security, as well as apply failsafe defaults. The resulting infrastructure should be efficiently maintainable, flexible and offer a solid foundation for software development. More recent standards reflect these demands, though few saw broader use across the industry [4].

In contrast to the Internet, security and cost efficiency are not the key objectives in avionics; rather safety is. However, most modern networking standards are aimed at achieving traditional PC-world security objectives and only indirectly address safety requirements (by fulfilling traditional security objectives) [5, 6].

In ARINC 664 Part 7, also referred to as AFDX, standard Ethernet technology is extended and design objectives are built around safety.

Two of the most important network architectures in the avionics industry are outlined in this paper, and we aim at depicting the evolution of networking concepts and requirements over the course of the past 30 years. It mainly is focused on the most prominent current and past standards, ARINC 429 and 664, but also covers two other significant standards (MIL-STD-1553 and ARINC 629). These standards introduced important features into aerospace networking design and are used as intermediate steps in this paper even though AFDX evolved independently.

In this paper, a deeper understanding of Ethernet is assumed; the reader should be familiar with redundancy and failover concepts, as well as information-security. The OSI layer model is used throughout this paper, even though it is not used within the cited avionics standards. When referring to layer 2 (L2) frames, Ethernet or AFDX frames at the data link layer are meant, while L3 and L4 refer to data structures used in the respective protocol at the network and transport layers.

Within the next section, the most widespread standard, ARINC 429, is explained in detail. In Section 3, the transition from federated network architectures, such as 429 to modern *Integrated Modular Avionics*, is depicted. Then, an analy-

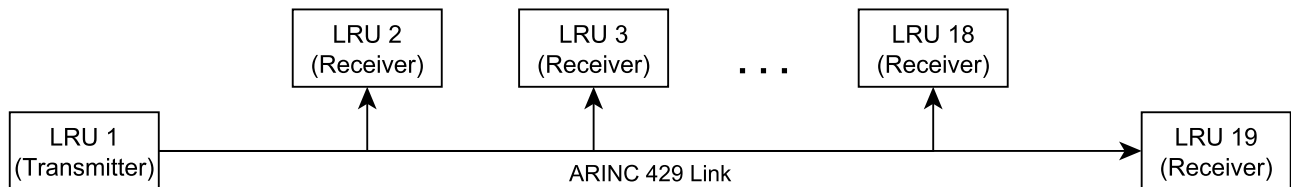


Figure 1: An ARINC 429 layout with just one transmitting LRU and up to 19 recipient. [4]

sis of the reference operating system proposed in ARINC 653 for use with integrated architectures is conducted. In Section 4, ARINC 629 and Mil-Std-1553, two more recent networking standards are briefly introduced. Section 5 is focused on the networking standard AFDX. The emphasis is on the enhancements to Ethernet needed to comply with the requirements of avionics applications. The final chapter is dedicated to summarizing the advantages and disadvantages of the main two named architectures.

2. ARINC 429

In the following section, the ARINC standard 429 will be depicted in detail. We will especially focus on its architectural principles, history and capabilities. Subsequently, the limitations imposed on networks design will be outlined.

2.1 Basic Topology

ARINC 429 (in full, *Mark 33 Digital Information Transfer System*), implements serial line communication and was one of the first standards specifically targeted at avionics applications. ARINC 429's predecessor, the commercial standard 419 [2], was first published in 1966, with 429 being based on (from a 1978 viewpoint) more recent technology [1].

429 specifies direct wiring of LRUs using a serial *twisted shielded pair*-based interface that can connect peers which are up to about 90 meters apart. Unlike in modern networking protocols, this is a signaling standard; thus the sender always sends to the line, and the recipients always read from it. If no data is available for sending, the line is set to zero-voltage. Even though *twisted shielded pair* cabling is used, all lines are simplex connections populated by one single sending station and multiple recipients (up to 19), as shown in Figure 1 [1].

The bus, also referred to as a *multi-drop bus*, can operate at *low* or *high speed*. *Low speed* uses a variable clock rate and a nominal throughput of 12-14 kbps, while the *high speed* mode requires a fixed clock rate and allows 100 kbps.

In ARINC 429 terminology, each chunk of data transmitted over a link is called a *word*; the word format is defined in the next section. Two types of words exist: data words and message control words. Messages consist of multiple words of data, which are then called records [4].

Link control messages are used in a similar way as in modern networking stacks. If a listening LRU is ready to receive data, a *Request to send* message will be transmitted via its dedicated sending link. *Clear to send* is used for the opposite. *data follows*, *data received OK*, *data received not OK* and *synchronization lost* can be used by the sender/recipient for further interaction. Each *message* is started by the message control word *data follows*, followed by up to 126 data words [1].

In contrast to today's networking standards, as 429 defines a unidirectional and simplex bus, recipient LRUs may not send messages to the bus they are listening to. This includes messages related to link control.

A station may be attached to multiple buses and operates as either sender or recipient, thus hierarchical layouts are possible. However, bidirectional information exchange between systems is at least required for message control and acknowledgment [1]. In this case, recipient LRUs respond via a secondary interface, on which the positions of sender and recipient are interchanged. As only one single station participating in an ARINC 429 link can occupy the sender role, one back-channel for each LRU is required [4].

2.2 Word Format and Data Types

Multiple data encoding formats which are targeted at different usage scenarios in avionics are defined: *Binary*, *Binary Coded Decimal* (BCD, see Figure 3), *Discrete Data*, *File transfer using the ISO 646 character set*, and *Maintenance Data and Acknowledgment* [1].

The structure of ARINC 429 words does not conform to modern communication standards; it is non-byte-aligned, but optimized for keeping down latency.

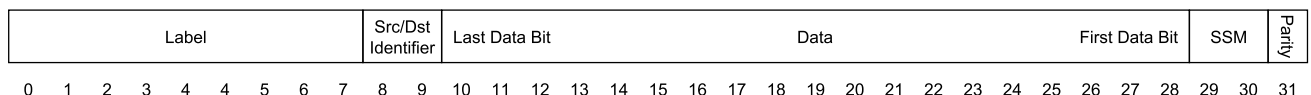


Figure 2: The data format of an ARINC 429 word. [1]

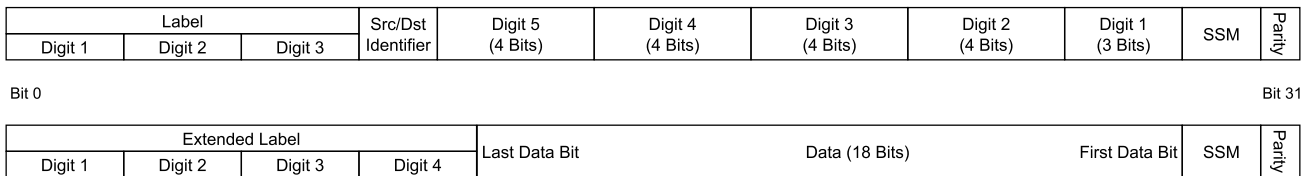


Figure 3: Two 429 words in *binary coded decimal* format(above) and *binary* format. The binary word has an extended label (4 digits/3 bits each), whereas the BCD word in this example uses a regular 3-digit label. [1]

Common fields in all word formats are:

- *label* (8 bit),
- *source* and *destination identifiers* (optional, 2 bit),
- the *sign/status matrix* (2 bit), and
- *data* (19 bit) field.

Words are terminated by a single parity bit. The total size of all words is 32 bits as depicted in Figure 2.

The small message size results in very low latency, minimizing delays during processing and guaranteeing timing, as no transport side queuing or rescheduling of traffic can happen. The message size is one of the cornerstones for achieving safety, resilience and reliability [1]. No other communication standard currently in use in avionics offers the same level of responsiveness, which makes it an interesting choice for applications where a delay of a few milliseconds may be too much [7].

Depending on the chosen data format, the *Sign/Status Matrix* flags have predefined meaning; for example, if the BCD format is used, setting both SSM bits to zero means *North, East, Right, To* (directional), *Above* or *Plus*. Other bit-patterns and data encoding variants have different predefined meanings, which LRUs are supposed to support to retain compatibility [1].

A word’s label is used as frame header, holding information on the encoding format and three octal numbers, which can be used by LRUs to select matching messages upon receipt. The label may be extended by three bits, which then serve as fourth label digit. Label digit codes are predefined in the standard and have fixed meanings, too. The label is sent high-order-bit first, whereas the rest of the message is sent least-significant-bit first [1].

2.3 Limitations

Due to the simplistic layout of 429 links, each individual connection is a physical cable, which allows for easy testing, as either a LRU or the line itself may be faulty. This also poses a severe challenge when designing systems with dense interlinking.

As depicted in Figure 4, even an environment with few stations present may become very complex once a certain degree of interaction is needed [1]. In modern commercial aircraft, interlinking between a multitude of systems as well as extreme cabling overhead may occur, imposing severe limitations on network design as well as impacting the overall weight of such a craft [7].

Bit-error correction via symmetric or cryptographic checksum algorithms is not designed to happen within ARINC 429 but instead needs to be implemented at application level. In fact, all data processing must be handled by each LRU’s software directly. There is no uniform, device-independent 429 software stack [1].

Overall, custom proprietary (and thus, expensive) hardware is required to implement an ARINC 429 setup, which is common in the aerospace industry. Almost no consumer-off-the-shelf hardware is available, with the exception of cabling. However, it should be noted, separate aviation-standards apply to cabling. Software development is problematic too, as no modern day networking protocols can be used, and development is done for highly specialized hardware. Retrofitting of older aircraft with new technology may thus be costly.

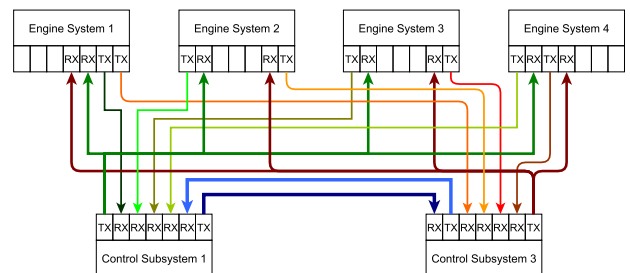


Figure 4: An ARINC 429 network containing just two control subsystems and four data-consuming components. [7]

3. INTEGRATED MODULAR AVIONICS

As has been outlined in the previous section, federated architectures severely restrict scalability and flexibility of computerized environments with dense interlinking. Thus,we will take a look at the modern day alternative to federated avionics in this section. First Integrated Modular Avionics will be introduced, followed by a brief analysis of ARINC 653.

3.1 Towards Integrated Modular Avionics

ARINC 429 allows the implementation of federated network architectures. It does not distinguish between hardware and software, but rather between devices specially built for a single purpose (e.g. polling a sensor, transmitting radar signals, issuing steering commands, etc). LRUs are thus unique to a certain degree and also need to be certified as a whole. As there is no distinction between hardware and software,

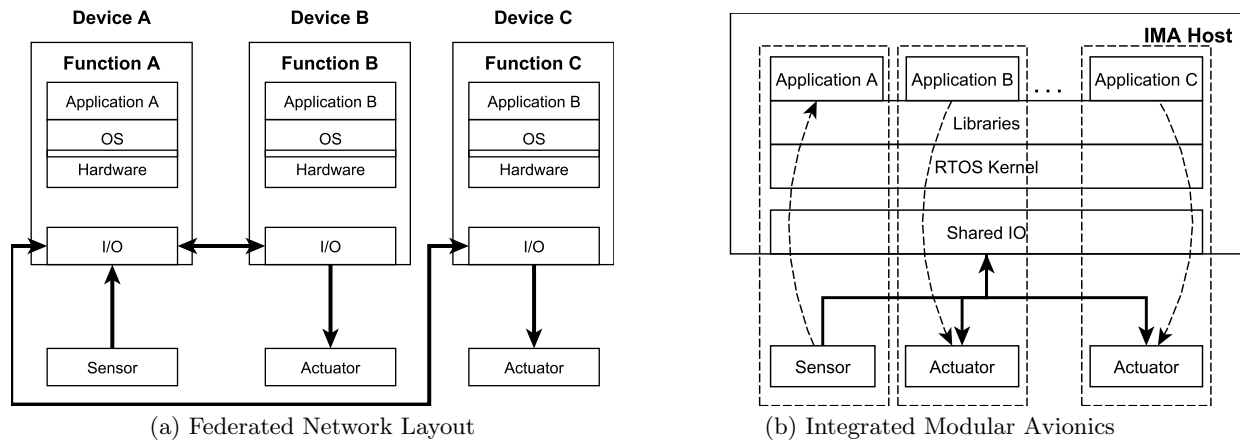


Figure 5: A comparison of network architectures. [8, 9]

re-certification needs to take place even if only software was updated or replaced [10].

As software may require additional system resources in the future (more RAM, disk space, CPU-power, ...), devices need to be constructed with appropriate reserves. Components in federated networks (usually) cannot share resources and reserves need to be defined on a per-system level, network-wide resulting in a high degree of idle capability, extra weight and extra cost. The overall result is a proprietary network with potentially extreme cabling overhead containing a multitude of different line replaceable units or modules fulfilling unique roles [11].

Due to the described limitations, the aerospace industry has begun to move away from federated architectures in favor of *Integrated Modular Avionics* (IMA) [9, 12]. Most importantly, IMA does discriminate between software and hardware and defines an abstraction layer between physical hardware and software-implemented functionality. However, IMA still allows the use of monolithic, unique LRUs, but encourages the homogenization of on-board hardware [10].

Multiple functional packages implemented by different software may be executed on a single IMA-host, as depicted in Figure 5. As such, applications must be isolated from each other, I/O ports and system resources need to be provided to the correct applications. As resources on an IMA-host are shared and available to all applications, reserves can be calculated collectively for all run applications [13].

3.2 ARINC 653

As described in later sections, timing and timing constraints are critical in avionics and networks design. These restrictions cannot be met by regular operating systems; hence, the use of a real time operating system (RTOS) is mandatory in IMA [12]. A reference implementation of such an operating system is specified in the ARINC standard 653, including a matching API and communication concepts for use on an IMA-host. The standard is categorized in different parts, covering essential and extra services that can or must be provided by the OS, as well as guidelines for testing [13].

In contrast to the operating systems used in most of the computer industry, 653's API *APEX* allows applications to remain completely independent of the underlying system's architecture, as shown in Figure 7. It grants applications access to hardware, allows file handling through a dedicated core service and permits access to the virtual networks backplane. Applications are assigned queuing ports, which are FIFO buffers for packets, and sampling ports, which are single-packet buffers to be overwritten upon each iteration. In conjunction with *APEX*, communication ports allow time-deterministic networking [13].

In 653, the core components of the RTOS are defined; many components are identical or very similar to those in use in common non-avionics real time operating systems and will thus not be described further. Others, like partition management and the *health monitor*, are uncommon. The *health monitor* provides hardware, software and network status information to the OS as well as the partitions. Thus, both the RTOS' core services as well as applications must support health monitoring functions [11].

Feedback on faults related to system services allows the *health monitor* to initiate countermeasures in case of failure (e.g. memory relocation, starting of service routines, etc). In case application partitions receive relevant information on issues, they can also take countermeasures on their own by selecting a different path to a sensor in the network or switching to fail-over functionality [11].

Health monitoring requires classification and categorization based on information provided by the application's author or manufacturer of the individual device. Thus, errors relevant for health monitoring are handled and also detected at different levels on an IMA host. Information on all of these error handling strategies are collected in recovery strategy tables at a per-partition level and at the system level. The tables themselves are to be maintained by the aircraft designer.

Isolation was one of the key design objectives in 653 and IMA, as they must allow side-effect-free executions of ap-

plications. Applications only need to properly make use of *APEX* and can be completely isolated from each other or can be permitted IPC [11]. *APEX* and the underlying system libraries are usually considered as one single functional block. Each application is run inside an isolated partition¹. The partition management services of the RTOS are responsible for assigning priorities and timing constraints for the individual application partitions [13].

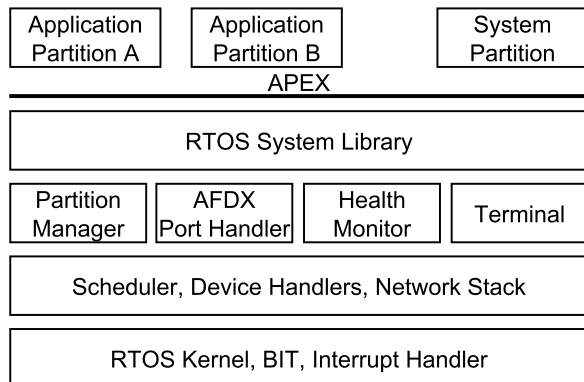


Figure 7: The functional layout of an operating system as specified in ARINC 653. [13, 14]

Therefore, if ARINC 653 is implemented, hardware and software can be certified incrementally. If software is replaced or updated, only the individual application or the OS is required to be re-certified. The underlying hardware remains unaffected and does not require additional steps to be taken beyond contingency planning and mitigating the need for possible extra systems resources [12, 13].

Strict guidelines are imposed on application design as well. In 653, among other guidelines, the way application software and core services of the OS are supposed to store and handle files is specified (i.e. configuration files are to be stored as XML files). *APEX* compliant applications can either run persistently or transiently, and thus can be started, stopped or moved to other IMA-hosts if it is permitted by the systems and networks configuration. IMA thus requires abstraction

¹653 partitioning is not to be confused with hardware partitioning commonly in use in commercial high performance computing.

of networking hardware to satisfy software requirements.

4. LEGACY NETWORKING STANDARDS

Strictly federated ARINC 429 networks do not offer the logical abstraction required for deploying integrated modular avionics; a shared medium and logical abstraction of interfaces is needed. In the following section, legacy network architectures offering such abstractions will be described briefly. First we will take a look at ARINC 629; while it is used rarely in avionics, it introduced a very important concept for gaining determinism in a shared medium network. Then, we will investigate into one of the most widely deployed military networking standards, Mil-Std-1553b.

4.1 ARINC 629

ARINC 629, also known as *Digital Autonomous Terminal Access Communication*, was developed jointly by Boeing and NASA to overcome limitations imposed by 429, and was later handed over to ARINC. Though more modern than previous standards, it never saw wider use in the industry. It was almost exclusively used in the Boeing 777 and even this aircraft contained additional 429 backup infrastructure. According to 629, the deployment of a triplex-bus layout, similar to the one specified by 10BASE2/10BASE5, was foreseen. It also implements CSMA/CD as used in Ethernet [16]; thus, collisions may occur on the bus. The original standard defined a clock-speed of 2 MHz, which can be raised as technology advances, and the bus was built upon twisted pair cabling from the very beginning. Like ARINC 429 it uses 32 bit data words, but the data element is 2-byte-aligned [4].

While 429-based architectures support only direct peer-to-peer communication, 629 mainly uses directed unicast and allows broadcast on a shared medium. It was an early attempt to increase flexibility and reduce cabling effort. Flexibility and the added complexity resulted in increased signal latency and reduced determinism, while the concept in general still lacks support for COTS-hardware. Thus, bus access by multiple stations in a 629 network may happen in random order, depending on which system sends first.

Parts of the basic protocol in 629 have found their way into modern standards, especially the use of predefined gaps between transmissions to prevent stall or packet loss to add determinism. As depicted in Figure 6, the *Synchronization Gap* is a random per-frame interval, whereas the terminal gap is distinct for each terminal.

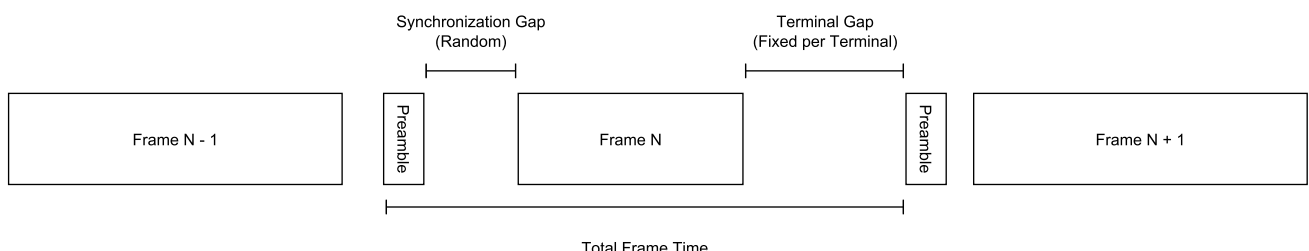


Figure 6: Frame-timing as specified in ARINC 629. [4]

4.2 MIL-STD-1553b

MIL-STD-1553b, the *Aircraft Internal Time Division Command/Response Multiplex Databus*, is widely used in military aircraft (e.g. Airbus's A400M) and even the International Space Station [13]. It is an interesting approach that also helped bridge the gap between basic signaling and modern networking standards like AFDX. The standard was evolved from the initial legacy data rate of 1 Mbps to the *extended* and *hyper* variants, which use newer hardware to offer 120 and 200 Mbps respectively [7]. In contrast to other bus based standards, MIL-STD-1553b specifies a logical star on top of the physical bus topology. This topology is called *robust physical layer* and uses *triaxial cabling* [15].

In the standard the role of a *bus controller* is defined. This device is responsible for initiating all communication between *subsystems* (peers) on the bus through a command-response protocol. In case the *bus controller* fails, another *remote terminal* can take over this role for the time being. To ensure fail-over, multiple redundant bus instances operate in parallel (see Figure 8), while each bus still only allows half-duplex communication [3].

Subsystems sending data via the bus are not directly connected, but instead use individual *remote terminals* to access the network. All communication on the bus is supervised by the *bus monitor*, which may also perform logging of parts or all the communication. Usually, transmissions are unicast, and are thus only exchanged between two remote-terminals. Broadcast is supported by design but discouraged [3].

In 1553, data words are just 20 bits long, but there is less protocol overhead than in other standards, as all communication is directed by the *bus controller*, and peers only execute commands they were given (e.g. read from bus, send to terminal).

To save on bit-space, three word formats exist:

- command word (sent by the *bus controller*),
- status word (response from LRUs to *bus controller*),
- and data word format.

While the standard is still incompatible with modern day networking protocols and does not adhere to the OSI layer model, it allows an LRU to emulate logical links, which is required for integrated system architectures [17].

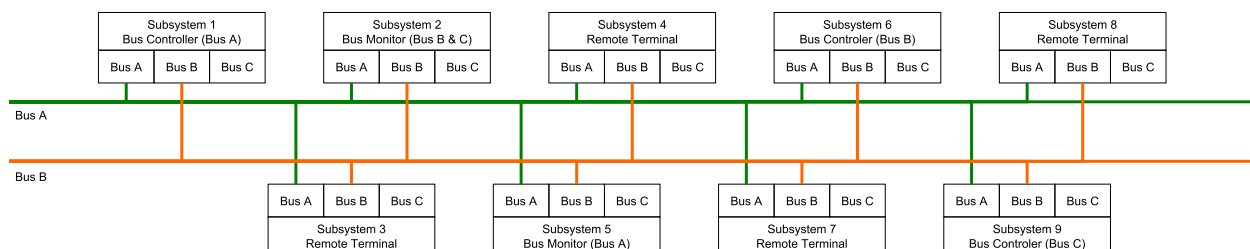


Figure 8: A single redundant MIL-STD-1553 bus network with hardware and device roles predefined for provisioning a second fail-over bus C. [3, 15]

5. AVIONICS FULL DUPLEX PACKET EXCHANGE

In this final section, we will take an in depth look at *Avionics Full-Duplex Ethernet switching* (AFDX). We will analyze how it was designed to extend standard Ethernet to meet today's requirements in an aerospace environment. Afterwards, the key elements of an AFDX network and the changes necessary to upper layer protocols will be described.

5.1 A brief history of ARINC 664

As an evolved standard, 429 had many limitations, but it is a proven and commonly used protocol. As time progressed and technology advanced, more bandwidth, more flexible topologies and new challenges like *Integrated Modular Avionics* (IMA, see Section 3) [13, 17] emerged and were beyond ARINC 429's capabilities [12, 18].

ARINC 664 (Part VII) was initially developed by the EADS Airbus division as *Avionics Full-Duplex Ethernet switching* (AFDX). Though previous aircraft already deployed fully electronic fly-by-wire systems, wiring using previous standards could no longer meet the requirements of modern day state-of-the-art aircraft. In the case of AFDX, the Airbus A380 prompted for a new technological base to be implemented; thus, AFDX was created. Later on, Airbus' AFDX was transformed into the actual ARINC standard [19]. Figure 9 shows a simple AFDX-based Network.

5.2 From Ethernet to AFDX

5.2.1 Architectural Changes

Ethernet has been in use for decades outside of the aerospace industry and proved to be a robust, inexpensive, extensible and flexible technology. However, it cannot offer essential functionality required for high availability and reliability. Thus, it is not directly suitable for avionics. 664 offers modern day transfer rates, while building on top of the previously much-loathed Ethernet standard 802.3 [16]. AFDX inherits parts of the MIL-STD-1553 terminology and overall setup. Devices transmitting data via the network are called *subsystems*, which are attached to the network via *end systems*. The full-duplex network itself is called *AFDX Interconnect*; in Ethernet terms, this includes all passive, physical parts of the network, but not switches and other active devices [7].

The most prominent hindrance for using Ethernet networking in avionics is Ethernet's non-determinism. For regular computer networks, packet loss and timeouts are a common

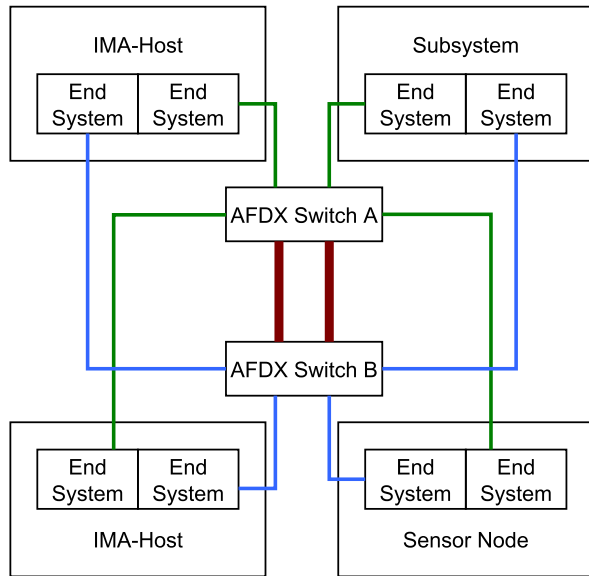


Figure 9: An example of an AFDX based network. Each subsystem is attached physically to the network by two *end systems*. [19]

issue. Upper layers, such as a station's operating system or applications, are supposed to handle these issues by design. If a message is lost or corrupted during transmission, it will simply be resent or its loss fully mitigated. When sending data on a non micro-segmented network, collisions may occur in each segment, forcing all stations involved in the collision to resend. Transmission of packets is retried after a random time interval by whichever station starts first. Again, a collision may occur which may lead to next to indefinite repeating, and this may subsequently result in a jammed bus [19].

Another variable factor of Ethernet networking and subsequently ARINC 664, are switches/bridges. While they add flexibility to networking, additional non-determinism is introduced, as frames may be reordered or manipulated in transit. Switches offer micro-segmentation of network segments, but in turn also increase the number of hops a frame takes from source to destination. Thus, latency is increased and timing behavior may vary if frames move along multiple

paths [19]. In highly congested setups, switches may even drop packets on purpose if buffer limits have been reached².

In Ethernet, collisions are handled via CSMA/CD, but upper layers may encounter packet loss. There, protocols (e.g. TCP, SCTP, etc) in the operating system's network stack have to deal with packet loss [13]. However, this is not a viable solution in safety-critical environments. Certain applications require bandwidth guarantees, while others may demand timing behavior to remain within strict boundaries. Neither can be offered by Ethernet. No *hard* quality of service guarantees are available in vanilla Ethernet, and soft scheduling is only offered through protocol extensions such as Ethernet-QOS IEEE 802.1p. The same applies to bandwidth allocation, which can not be guaranteed in Ethernet on a per-flow level, but is implemented using various different algorithms. While there are several proprietary approaches for making Ethernet usable in real-time environments, none of these standards is directly usable in avionics [20, 21]. Thus, the new standard required determinism to make it usable in avionics [19].

5.2.2 Virtual Links

Ethernet is independent of physical connections and allows logical endpoints to be defined. Multiple physical or virtual devices may thus share one link, supporting virtual subsystems or virtual machines in IMA [12, 13, 18]. Multiple applications or devices may require different timing characteristics or a fixed minimal amount of bandwidth [19].

Virtual point-to-point connections implement the same concept as used in ARINC 429. In contrast to 429, they do not exist physically, but as logical links. They are implemented as *Virtual Links* (VL) on top of the AFDX Ethernet layer. An example of virtual channels is given in Figure 10. To a certain degree, VLs are quite similar to VLAN tagging as defined in IEEE 802.1Q [22], but offer additional information in addition to network isolation. Each virtual channel has three properties besides its channel ID: the Bandwidth Allocation Gap, the maximum L2 frame size, called LMAX or Smax, and a bandwidth limit [4].

LMIN and LMAX are used to set a predefined smallest and largest common Ethernet frame size along the path a packet

²In a properly laid out AFDX network, buffer overruns should never actually occur. The network parameters are configured based on values calculated during the planning phase of an aircraft using a mathematical framework.

| ASN Channels | Sensor Rate Groups (BAG value) | | | |
|------------------------|--------------------------------|-----------------|-----------------|------------------|
| | 1 msec | 8 msec | 32 msec | 128 msec |
| Sensor Pod 1 (VL ID#) | 0 - 7 | 8 - 15 | 16 - 23 | 24 - 31 |
| Sensor Pod 2 (VL ID#) | 32 - 39 | 40 - 47 | 48 - 55 | 56 - 63 |
| Sensor Pod 3 (VL ID#) | 64 - 71 | 72 - 79 | 80 - 87 | 88 - 95 |
| Sensor Pod 4 (VL ID#) | 96 - 103 | 104 - 111 | 112 - 119 | 120 - 127 |
| Data Size | 64 bytes | 128 bytes | 256 bytes | 512 bytes |
| Busy Time/Channel | 0.84 μ sec | 1.48 μ sec | 2.76 μ sec | 5.32 μ sec |
| Total Time | 26.88 μ sec | 47.36 μ sec | 88.32 μ sec | 170.24 μ sec |
| Netto Bandwidth | 21.504% | 4.736% | 2.208% | 1.064% |

Table 1: Impact of the *bandwidth allocation gap* on virtual link performance. [13, p. 11]

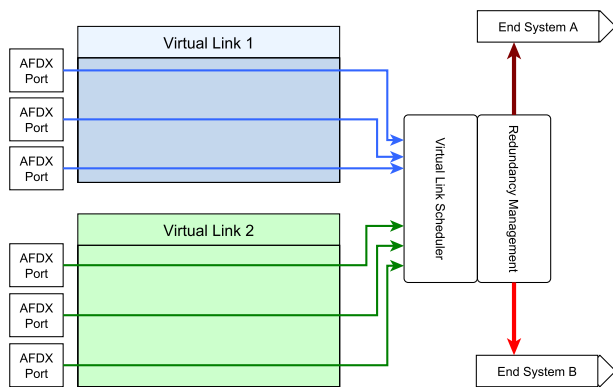


Figure 10: AFDX ports are bound to *virtual links*, characteristics specified by ports as well as VLS result in the actual VL parameters. [4]

may take, removing the need for IP-Packet fragmentation and similar mechanisms, thus removing yet another source of non-determinism from Ethernet [4].

Once the buffers present in switches or end stations are filled up, physical links can become congested and packets would be lost or delayed in transition. In safety-critical environments this is unacceptable; thus, the so called *Bandwidth Allocation Gap* (BAG) was defined. The BAG is defined in software for each VL, setting the delay (1-128 milliseconds) between sending the VL's frame and the next. In congested networking scenarios, frames will be sent within the time-window given by the BAG [19].

By setting LMAX and a reasonable BAG, bandwidth is segmented and can thus be guaranteed for each VL individually. The downside of introducing timesharing is a considerable loss of bandwidth if multiple links with different properties are defined. Table 1 reflects this loss of throughput depending on the network's parameters. The data was taken from benchmarking conducted by NASA in [13, p. 11] based on a 100 Mbps ancillary sensor network configuration. Even if little data is sent, one single VL with high BAG (long wait-time between frames) and low LMAX (small supported frame size) can drastically reduce overall network throughput [7].

Virtual Links are designated using so called *Virtual Link Identifiers* (VLID). The VLID replaces MAC-address based delivery, occupying the bits normally used for the destination MAC. To retain compatibility with Ethernet, AFDX splits the destination-MAC field into multiple parts: the initial bits are set to reflect a locally administered MAC-address (site-local), the final 16 bits store the VLID [19].

Only one *subsystem* may send data using a given VLID, thus *Virtual Links* are again unidirectional. As in ARINC 429, a *subsystem* can assume different roles in multiple VLs using different ports (see below), and multiple recipients may participate in a *Virtual Link*. Subsystems are not explicitly addressed, as in common Ethernet where MAC addresses are used, but the meaning of a *Virtual Links* identifier is defined

and enforced by the network configuration [23]. However, parts of the original MAC address data area are designated user specifiable.

To make use of AFDX's capabilities, traditional socket programming is insufficient. Thus, a layer of abstraction was added: the communication ports. Communication ports - that is *sampling* and *queuing ports* - are accessed through a dedicated networking API (see ARINC 653) by a *subsystem*. Ports are assigned to *Virtual Links* and used to exchange data between *subsystems*; *end systems* deliver messages to one of the *subsystem's* ports, multiple ports at a *subsystem* can be members of a VL, but each port may only be attached to a single VL.

Sampling ports have dedicated buffer-spaces in which one single message can be read and stored. If a new message arrives, previous data will be overwritten. A *queuing port's* buffer may contain up to a fixed number of messages that are stored in a FIFO queue; upon reading the oldest message, it is removed from the queue. Handler services for communication ports need to be provided according to the ARINC 653 specifications [12]. BAGs and LMAX of a VL should be set accordingly to the collective requirements of all ports participating in a link [19].

Scheduling is performed on a per-port and per-link level at each *end system* independently based on the previously named properties. End systems ensure *Virtual Links* do not exceed their bandwidth limits, when multiplexing transmissions from different ports and *Virtual Links*. Also, jitter must be kept within fixed boundaries (specified by the standard), as frame-transmissions are offset by jitter within the BAG [19]. *Virtual Links* are scheduled before redundancy is applied to transmissions.

Subsequently, data passing through a Virtual Link will always take the same path in an active AFDX network, as they are predefined on the AFDX-switches along the route. As reconfiguration of AFDX devices does not happen at runtime, this path will persist until the relevant devices in the network are reinitialized. This implies timing behavior at runtime will remain constant for each frame (as defined by the BAG), even if individual devices along the path fail. Single failures will have no impact on the overall setup due to the provided redundancy. More on AFDX-switching will be discussed in Section 5.2.4

5.2.3 Redundancy

High availability environments also require redundancy on the bus as well as within stations. Again, Ethernet does not offer any sort of fail-over by default, however, optional link aggregation as defined in IEEE 802.1AX [24] can offer such functionality. 664 by design specifies sophisticated redundancy concepts for end stations as well as cabling by providing two dedicated networks (network A and B). After scheduling of Ethernet frames, redundancy is introduced. Each AFDX *subsystem* has two interfaces called *end systems*. Redundancy is added transparently by sending each frame via both *end systems*, applying the frame sequence number (see Figure 10). Assuming no transmission errors occurred, one duplicate will arrive at the destination for each frame transmitted [19].

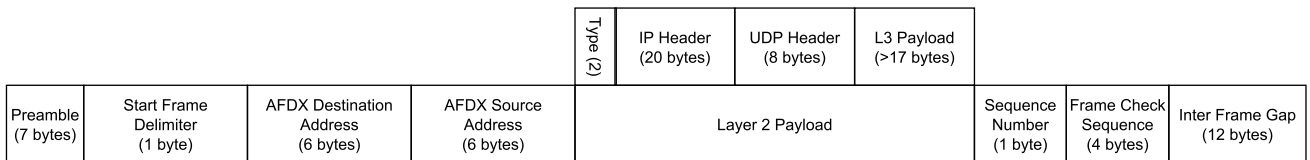


Figure 11: The layout of an ARINC 664 frame is almost identical to an standard Ethernet frame and can be transported via COTS Ethernet hardware. [19]

Duplicate frames must be detected and discarded. To do so, a single byte sequence counter was appended to each OSI layer 3 packet, turning the resulting Ethernet frame into an AFDX Frame, as depicted in Figure 11. The AFDX sequence number is added as a trailer to the Layer 2 payload. For each packet sent via a *Virtual Link*, the counter is incremented. The first frame with a valid checksum to arrive at a *subsystem* is used for processing, subsequent frames with a duplicate sequence number can then be identified and discarded [4].

In case of a failure on the *AFDX Interconnect*, software and switches can be made aware of this issue and may divert traffic along other, predefined fail-over paths within the same network without having to rely solely on the independent network redundancy [11].

AFDX currently supports line speeds of up to 1 Gbps (see Table 2 created by [7]), but may support faster transports as technology advances. End systems are connected to switches directly and use full-duplex links [4]. As defined in the standard and due to the use of fully-switched Ethernet, only two stations exist in each AFDX Layer 2 segment: an *end system* and a switch, two switches, or two *end systems*. Collision domains are thus minimized. Subsequently frame collisions that need to be resolved via CSMA/CD on a single segment essentially should not happen [25].

5.2.4 AFDX Switches

Most features AFDX consists of can also be implemented using regular Ethernet hardware, if special AFDX-stack implementations are run. While purely software-based implementations exist [23], these solutions can not guarantee determinism. They can not keep jitter within boundaries imposed by AFDX and are useful for basic interoperability testing only.

To achieve determinism, specialized hardware to enforce the *Virtual Link* rules, which are based on the VL parameters introduced by ARINC 664 is needed. AFDX switches fill this role and enforce latency, bandwidth constraints for VLs and provide a dependable, fixed configuration. This configuration is read at bootup and remains constant at run time to avoid fluctuations in the network's topology and provide uniform timing behavior.

For integrity reasons, store-and-forward circuit switching is used when relaying packets, in contrast to most modern day high-speed Ethernet switches, which perform cut-through switching [19]. The configuration for all *Virtual Links* (LMIN, LMAX, BAG, priority) and switch param-

eters should be set according to a one of the mathematical proofing models in use today [26, 27].

By fixing network parameters at boot-up, changes at run-time are prevented and the network retains constant timing properties and a static layout throughout operation. Non-fault generated deviations off default settings may not happen and are taken into account when calculating global parameters mathematically [27]. Switches isolate *Virtual Links* from each other and perform scheduling for passing-through frames based on their VLID [4]. Other parameters specified in switch and system configuration include priority, LMIN (equivalent to LMAX) and jitter for *Virtual Link*. Ports have a fixed maximum delay and buffer-size [19].

Certification of components for use in avionics environments requires provable properties and usually results in a worst-case but congestion-free setup. Network Calculus [28] is widely used, but alternative approaches such as the trajectory based models or model-checking would be viable alternatives, too; common to all of them is a resulting formal proof of the network's correct functionality [26, 27].

5.2.5 Impact On OSI-Layer 3 and Above

AFDX adheres to the OSI layer model and is based on common protocols from the Internet-world. Subsequently, familiar protocols like IP, UDP and IP-multicast are used. Alien networking environments, such as ARINC 429 links, can be transported within a *Virtual Link* transparently to the individual applications, thereby reducing development effort. In fact, virtually any previous network standard which does not exceed ARINC 664 in capabilities can be implemented on top of it [19].

At Layer 3, the IPv4 protocol is deployed, though the fields usually used for source and destination IP-addresses have been reassigned, as depicted in Figure 12. The top packet-version shows an IP packet being directed to an individual system using the VLID, while the bottom packet uses multicast-addressing. The 32 bits of the source IP address field are separated into:

- the single bit *class identifier*,
- 7 bit *private address*,
- user-defined 16 bit ID,
- as well as an 8 bit *partition identifier*.

The *partition identifier* is used to address virtual *subsystems* in a virtualized IMA environment [12, 18].

The Destination IP is either used to designate a multicast

| | | | | | | | |
|---------------------|--------------------------|--------------------------------------|-------------------|-----------------------------|---------------------------|--------------------------|---------------------------|
| | Class Prefix (4 bits) | Constant Field (12 bits) | VLID (2 bytes) | AFDX Prefix (1 + 7 bits) | User Defined (2 bytes) | Partition ID (1 byte) | L3 Payload (>17 bytes) |
| Unchanged IP Header | IP Destination Address | | | IP Source Address | | UDP Header (8 bytes) | AFDX Payload |
| | Class Prefix (4 bits) | IP Multicast Identifier (28 bits) | | AFDX Prefix (1 + 7 bits) | User Defined (2 bytes) | Partition ID (1 byte) | L3 Payload (>17 bytes) |

Figure 12: Similar to AFDX’s frame format on OSI-layer 2, the structure of the IPv4 header remains unchanged. [19]

IP address, or contains a field of 16 bits prefixed to the VLID. The first 16 bits contain a fixed number (specified by the standard), while the second part contains the VLID, if direct IP-addressing and IMA is used [19].

Due to the guarantees provided by AFDX, certain features usually introduced at higher OSI layers (e.g. packet-loss handling and reordering of packets) are already implemented by the underlying L2/3-networking structure. In commercial networking, protocols such as TCP or SCTP are used to provide this functionality. In AFDX, transmission control and integrity is already provided at the lower layers, thus, UDP was chosen to be the default protocol in AFDX [7].

AFDX-Ports are mapped directly at UDP’s source and destination port fields. AFDX-flows are identified by using a combination of the following parameters:

- destination MAC address (containing the VLID),
- source and destination IP address,
- source and destination UDP port,

Due to architectural restrictions, the minimum payload size for packets transmitted inside a AFDX-L3 packet is 144 bits. If an UDP packet’s length drops below this limit, padding is added at the end of the L4 packet [19].

The standard also defines monitoring to be performed via SNMP, and intra-component data transfer through TFTP. Payload transferred inside the L4-structure usually has no fixed predetermined meaning, in contrast to earlier standards. However, ARINC 664 defines a number of common data structures, such as floating point number formats and booleans. These do have no direct impact on network payload, but offer common ground for software development [19].

6. CONCLUSIONS

ARINC 429 was developed at a time when the use of inter-connected, programmable *subsystems* aboard aircraft was simply not feasible due to aspects such as size, energy consumption, fragility and hardware cost. 429 solely treats data transfer between systems at a per-device level, interconnecting systems on a pin level. Though it has advantages over more modern standards, it clearly had reached its limits once multipurpose computers are interconnected. However, 429 will most likely not simply vanish; it will still be used in scenarios where simple signaling is sufficient, and in latency critical scenarios. It is a proven and extremely reliable technology and thus is also used as fall-back network for the AFDX network, e.g. in the Airbus A380.

Most of 429’s limitations have been identified decades ago, but no uniform standard had been adopted by the aerospace industry. Other standards introduced more modern, faster or more flexible networking models, in contrast to basic signaling as in 429. However, known attempts are either in use exclusively in individual aircraft models or are applied only internally by manufacturers (629, *ASCB*, *CSCB*, *EFabus* [4]) offering little or no compatibility to the other implementations. Still, these standards introduced approaches which can be found in an altered form in AFDX [7].

AFDX combines proven safety and availability functionality with modern technology to be able to handle today’s requirements. It adheres to the OSI-layer-model and outlines a compatible stack architecture, while allowing to emulate previous communication standards on top. Besides, the Internet Protocols Suite (IP/UDP) and Ethernet are used and only slight alterations to the individual data structures are applied, which lowers the bar for designing hardware and developing software in avionics considerably.

| | ARINC 429 | ARINC 629 | Mil-Std-1553 | ARINC 664 (at 100 Mbps) |
|---------------------------|----------------|-------------|-----------------|-------------------------|
| Topology (logical) | Bus | Bus | Bus (Star) | Star |
| Duplex | Simplex | Half-Duplex | Half-Duplex | Full-Duplex |
| Medium | Dedicated | Shared | Shared | Shared |
| Speed | 100 kHz | 2 MHz | 1 MHz | 100 MHz |
| Bandwidth | 2778 words/sec | Variable | 46000 words/sec | 3,000,000+ frames/sec |
| Latency | Fixed | Bounded | Variable | Bounded |
| QoS | 100% | None | None | Configurable |

Table 2: A capability comparison of ARINC 429, 629, Mil-Std-1553 and AFDX. [7]

For certain parts of an AFDX network, COTS hardware can be used in conjunction with matching software, though AFDX hardware implementations must be used to retain determinism. Still, by adding standard Ethernet hardware in conjunction with an AFDX-stack implementation in the operating system, non-AFDX hardware could be used without further alterations [19, 23].

Changes to the overall network layout do not negatively impact individual *Virtual Links* or ports of the individual end- and *subsystems*, due to the added abstraction [12, 18]. Diagnosing issues within an AFDX network requires highly skilled personnel, in contrast to 429. Still, hardware swapping in a 664 network can be done with little effort whereas fixing a line running through an entire aircraft due to a fault may require considerable effort [7]. ARINC 664 supporting devices may also support virtualization and hardware partitioning, as virtual/logical devices as specified in IMA/ARINC 653 can be used [29]. 429 will never be able to support IMA architectures, let alone non-physical hardware [12, 13, 18].

AFDX implementations still remain relatively conservative, using a proven and mature technology base instead of state-of-the-art hardware [14]. For example, the Airbus A380's and A350's networks are based on copper-cabling, while optical cabling has become the de-facto standard for high-speed interconnection on backbones in corporate and carrier backbones. However, ARINC 664 architectures can integrate future technology seamlessly. Future AFDX implementations like the one used in Boeing's 787 will use fiber-optics.

664's flexibility in complex setups makes it an obvious solution to the cabling overhead and complexity issues of 429. While offering a comparable level of reliability as Mil-Std-664 in civilian application, its architecture is a considerably more adaptive and can benefit seamlessly from the parallel evolution of the Ethernet standards family. Also line speeds previously unachievable in commercial avionics are now attainable and require far less development effort for both hardware and software, as the technological base still remains to be Ethernet.

In the long run, the technological layering will minimize the need to review AFDX to incorporate new features, as those can be introduced via the underlay standards. AFDX is far more advanced, modern and powerful than previous standards, yet retains its flexibility. In the future, we will most likely observe an acceleration of the introduction of this new networking standard into commercial avionics.

7. REFERENCES

- [1] P. Frodyma and B. Waldmann. *ARINC 429 Specification Tutorial*. AIM GmbH, 2.1 edition, 2010.
- [2] Working Group. ARINC 615, P2: ARINC Airborne Computer Data Loader. Technical Report 1, Aronautical Radio, INC, June 1991.
- [3] P. Frodyma, J. Furgerson, and B. Waldmann. *MIL-STD-1553 Specification Tutorial*. AIM GmbH, 2.3 edition, 2010.
- [4] L. Buckwalter. *Avionics Databases, Third Edition*. Avionics Communications Incorporated, 2008.
- [5] Miguel A. Sánchez-Puebla and Jesús Carretero. A new approach for distributed computing in avionics systems. In *Proceedings of the 1st international symposium on Information and communication technologies*, ISICT 2003, pages 579 – 584. Trinity College Dublin, 2003.
- [6] N. Thanthry and R. Pendse. Aviation data networks: security issues and network architecture. *Aerospace and Electronic Systems Magazine, IEEE*, 20(6):3 – 8, June 2005.
- [7] T. Schuster and D. Verma. Networking concepts comparison for avionics architecture. In *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th*, pages 1.D.1–1 – 1.D.1–11, October 2008.
- [8] J. Craveiro, J. Rufino, C. Almeida, R. Covelo, and P. Venda. Embedded Linux in a partitioned architecture for aerospace applications. In *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*, pages 132 – 138, May 2009.
- [9] M.J. Morgan. Integrated modular avionics for next-generation commercial airplanes. In *Aerospace and Electronics Conference, 1991. NAECON 1991., Proceedings of the IEEE 1991 National*, pages 43 – 49 vol.1, May 1991.
- [10] C.B. Watkins and R. Walter. Transitioning from federated avionics architectures to Integrated Modular Avionics. In *Digital Avionics Systems Conference, 2007. DASC 2007. IEEE/AIAA 26th*, pages 2.A.1–1 – 2.A.1–10, October 2007.
- [11] P.J. Prisaznuk. ARINC 653 role in Integrated Modular Avionics (IMA). In *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th*, pages 1.E.5–1 – 1.E.5–10, October 2008.
- [12] INC Aronautical Radio. *ARINC 653, P1-3: Avionics Application Software Interface*. ARINC Specification 653 Parts 1-3, November 2010.
- [13] Richard L. Alena, John P. Ossenfort Iv, Kenneth I. Laws, and Andre Goforth. Communications for Integrated Modular Avionics. In *IEEE Aerospace Conference 2007*, pages 1 – 18, March 2007.
- [14] R. Ramaker, W. Krug, and W. Phebus. Application of a civil Integrated Modular Architecture to military transport aircraft. In *Digital Avionics Systems Conference, 2007. DASC 2007. IEEE/AIAA 26th*, pages 2.A.4–1 – 2.A.4–10, October 2007.
- [15] Michael Hegarty. A Robust Physical Layer for aircraft data networks based on MIL-STD-1553. In *SAE 2011 AeroTech Congress and Exhibition*, pages 1394 – 1401, October 2011.
- [16] Working Group. 802.3-2008 IEEE Standard for Ethernet, 2008.
- [17] Gitsuzo B. S. Tagawa and Marcelo Lopes de Oliveira e Souza. An overview of the Intergrated Modular Avionics (IMA) concept. In *DINCON 2011*, pages 277 – 280. Conferência Brasileira de Dinâmica, Controle e Aplicações, September 2011.

- [18] John Rushby. Partitioning in avionics architectures: Requirements, mechanisms, and assurance. Final Report DOT/FAA/AR-99/58, NASA Langley Research Center and U.S. Federal Aviation Administration (US DOT), March 2000.
- [19] INC Aronautical Radio. *ARINC 664, P7: Avionics Full Duplex Switched Ethernet (AFDX) Network*. ARINC Specification 664 Part 7, June 2005.
- [20] M. Felsler. Real-Time Ethernet - Industry Prospective. *Proceedings of the IEEE*, 93(6):1118 – 1129, June 2005.
- [21] J.-D. Decotignie. Ethernet-Based Real-Time and Industrial Communications. *Proceedings of the IEEE*, 93(6):1102 – 1117, June 2005.
- [22] L A N Man and Standards Committee. *802.1Q-2005 IEEE Standard for Local and metropolitan area networks: Virtual Bridged Local Area Networks*, volume 2005. IEEE, 2005.
- [23] Emre Erdinc. Soft AFDX end system implementation with standard PC and Ethernet card. Master’s thesis, Graduate School of Natural and Applied Sciences of the Middle East Technical University, 2010.
- [24] Working Group. 802.1AX-2008 IEEE Standard for Local and Metropolitan Area Networks - Link Aggregation, 2008.
- [25] Todd Lammle. *CCNA: Cisco certified network associate study guide*. Sybex, San Francisco, 2004.
- [26] M. Boyer, N. Navet, and M. Fumey. Experimental assessment of timing verification techniques for AFDX. Technical report, Realtime-at-work, ONERA, Thales Avionics, 2012.
- [27] Ananda Basu, Saddek Bensalem, and Marius Bozga. Verification of an AFDX Infrastructure Using Simulations and Probabilities. In *Runtime Verification*, volume 6418 of *Lecture Notes in Computer Science*, pages 330 – 344. Springer Berlin / Heidelberg, 2010.
- [28] Jean-Yves Le Boudec and Patrick Thiran. *Network calculus: a theory of deterministic queuing systems for the internet*. Springer-Verlag, Berlin, Heidelberg, 2001.
- [29] D. Kleidermacher and M. Wolf. MILS virtualization for Integrated Modular Avionics. In *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th*, pages 1.C.3-1 – 1.C.3-8, October 2008.

ISBN 3-937201-27-0
DOI 10.2313/NET-2012-08-1

ISSN 1868-2634 (print)
ISSN 1868-2642 (electronic)