Network Architectures
and Services
NET 2013-02-1

FI & IITM
WS 12/13

Proceedings of the Seminars
Future Internet (FI) and
Innovative Internet Technologies and Mobile Communications (IITM)

Winter Semester 12/13

Munich, Germany, 12.10.2012-04.02.2013

Editors

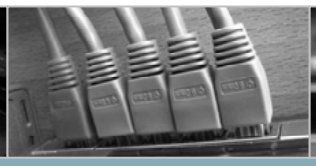
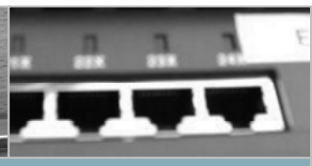
Georg Carle, Corinna Schmitt

Organisation

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

Technische Universität München



Network Architectures
and Services
NET 2013-02-1

FI & IITM
WS 12/13

Proceedings zu den Seminaren
Future Internet (FI) und
Innovative Internettechnologien und Mobilkommunikation (IITM)

Wintersemester 2012/2013

München, 12.10.2012 – 04.02.2013

Editoren: Georg Carle, Corinna Schmitt

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

Technische Universität München

Proceedings of the Seminars
Future Internet (FI) and Innovative Internet Technologies and Mobile Communications (IITM)
Winter Semester 2012/2013

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

Seminar FI & IITM WS 2012/2013
Proceedings zu den Seminaren „Future Internet“ (FI) und „Innovative Internettechnologien und Mobilkommunikation“ (IITM)
München, Germany, 12.10.2012-04.02.2013
Georg Carle, Corinna Schmitt
ISBN3-937201-33-5

ISSN: 1868-2634 (print)
ISSN: 1868-2642 (electronic)
DOI: 10.2313/NET-2013-02-1
Lehrstuhl Netzarchitekturen und Netzdienste (I8) NET 2013-02-1
Series Editor: Georg Carle, Technische Universität München, Germany

Vorwort

Wir präsentieren Ihnen hiermit die Proceedings zu den Seminaren „Future Internet“ (FI) und “Innovative Internettechnologien und Mobilkommunikation” (IITM), die im Wintersemester 2012/2013 an der Fakultät Informatik der Technischen Universität München stattfanden.
Im Seminar FI wurden Beiträge zu unterschiedlichen Fragestellungen aus den Gebieten Internettechnologien und Mobilkommunikation vorgestellt. Die folgenden Themenbereiche wurden abgedeckt:

- Internetzensur: Methoden und deren Beobachtungen
- Monitoring von Internetabschaltungen
- Ubertooth – Bluetooth Monitoring und Injection
- RFID – Überblick
- NFC – Möglichkeiten und Risiken
- CCN – Content-Centric Networking
- Alternativen zu X.509

Im Seminar IITM wurden Vorträge zu verschiedenen Themen im Forschungsbereich Sensorknoten vorgestellt. Die folgenden Themenbereiche wurden abgedeckt:

- Rootkits
- Software Defined Networking mit OpenFlow
- Ressourcen Management Tools
- Einfache PKI
- Was sind “individual-related” Daten?

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, Februar 2013

Georg Carle

Corinna Schmitt

Preface

We are very pleased to present you the interesting program of our main seminars on “Future Internet” (FI) and “Innovative Internet Technologies and Mobil Communication” (IITM) which took place in the winter semester 2012/2013

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:

- Internet Censorship: Methodes and their Observations
- Controlled Internet Outage Monitoring
- Ubertooth – Bluetooth Monitoring and Injection
- Radio-Fequency Identification – Overview
- NFC – Possibilities and Risks
- Content-Centric Networking
- Alternatives to X.509

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

- Rootkits
- Software Defined Networking with OpenFlow
- Resource Management Tools
- Simple PKI
- What is individual-related data?

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, February 2013

Seminarveranstalter

Lehrstuhlinhaber

Georg Carle, Technische Universität München, Germany

Seminarleitung

Corinna Schmitt, Technische Universität München, Germany

Betreuer

Stephan Günther, *Technische Universität München, Wiss. Mitarbeiter I8*
Ralph Holz, *Technische Universität München, Wiss. Mitarbeiter I8*
Heiko Niedermayer, *Technische Universität München, Wiss. Mitarbeiter I8*
Daniel Raumer, *Technische Universität München, Wiss. Mitarbeiter I8*
Johann Schlamp, *Technische Universität München, Wiss. Mitarbeiter I8*
Corinna Schmitt, *Technische Universität München, Wiss. Mitarbeiterin I8*
Lukas Schwaighofer, *Technische Universität München, Wiss. Mitarbeiter I8*
Simon Stauber, *Technische Universität München, Wiss. Mitarbeiter I8*
Matthias Wachs, *Technische Universität München, Wiss. Mitarbeiter I8*

Kontakt:

{carle,schmitt,guenther,holz,niedermayer,raumer,schlamp,schwaighofer,stauber,wachs}
@net.in.tum.de

Seminarhomepage

http://www.net.in.tum.de/de/lehre/ws1213/seminare/

Inhaltsverzeichnis

Seminar Future Internet

Session 1: Monitoring

Session 2: Mobilkommuikation

Session 3: Netzwerke und Sicherheit

Seminar Innovative Internettechnologien und Mobilkommunikation

Session 1: Netzwerke, Tools und Sicherheitaspekte

Internetzensur: Methoden und deren Beobachtung

Gerhard Hagerer
Betreuer: Lukas Schwaighofer
Seminar Future Internet WS2012/2013
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: gerhard.hagerer@in.tum.de

ZUSAMMENFASSUNG

In der vorliegenden Arbeit geht es um Internetzensur im Sinne von Internetausfällen, wie sie auf Anordnung von den Regierungen in Ägypten und Libyen zu Beginn des Jahres 2011 im Zusammenhang mit den dort auftretenden Revolutionen passierten. Im Referenzpaper [1] wurden diese durch BGP-Routenaktualisierungen und unangeforderten Datenverkehr (*Internet Background Radiation*) in Darknets durch Network Telescopes mit Erfolg beobachtet. Dabei konnten zwei Abschaltmethoden unterschieden werden: der Einsatz von Paketfiltern sowie Techniken, die dem Abtrennen ganzer Teilnetze des Internets (*autonome Systeme*) gleichkommen. Diese Themen werden in ihrer Theorie ausführlich allgemeinverständlich erklärt, um im Anschluss daran die Internetausfälle der Revolutionen in Ägypten und Libyen technisch analysieren zu können.

Schlüsselwörter

Ausfälle, Verbindungsabbruch, Zensur, Darknet, Network Telescope, Internet Background Radiation

1 Einführung

In der heutigen Zeit wird uns zunehmend bewusst, welche gesellschaftlichen und politischen Konsequenzen moderne Technologien wie das Internet haben. Soziale Netzwerke wie Facebook und Twitter ermöglichen vielen Menschen weltweit eine einfache, schnelle und starke Vernetzung miteinander. Dadurch ist es besser als jemals zuvor möglich, sich mit anderen online zu organisieren und auszutauschen. Diese moderne Entwicklung widerspricht dem Interesse von totalitären Regierungen, welche freie Meinungsäußerung und das Versammlungsrecht in ihren Ländern zu unterdrücken versuchen. Somit machen diese das freie Internet zum Feind und versuchen dieses einzuschränken. Dies wird mit den Mitteln der modernen Zensurtechnologie zu erreichen gesucht: entweder werden regierungskritische digitale Inhalte von höchster Instanz blockiert und können nicht heruntergeladen werden oder Internetbenutzern wird der Zugang zum Internet gänzlich verwehrt. Was in der Onlinewelt allerdings passiert, wenn eine ganze Bevölkerung sich gegen ihre eigene Regierung zusammenschließt, konnte am Anfang des Jahres 2011 im Zuge des arabischen Frühlings beobachtet werden: Ägypten und Libyen erlitten vollständige, von den dortigen Regierungen veranlasste Internetausfälle, was weltweit sichtbare Veränderungen im Internet auslöste, sowohl in Routingtabellen von BGP-Routern als auch in der Grundlast des Datenverkehrs [2] [3]. Es kann insofern von einer globalen digitalen Erschütterung gesprochen werden, wie sie bis zu diesem Zeitpunkt noch nie vorgekommen ist. Es dauerte nicht lang, bis Regional Internet Registries und Network Telescopes entprechende Daten veröffentlichten, die alsbald zum Objekt von Forschungen über die Beobachtung von Internetausfällen wurden. In diesem Zusammenhang ist insbesondere das Paper "Analysis of Country-wide Internet Outages Caused by Censorship" [1] zu erwähnen, welches besagte Ereignisse in Ägypten und Libyen zum Forschungsgegenstand hat. Die vorliegende Arbeit nimmt im Wesentlichen auf dieses Bezug, um die in den beiden Ländern angewendeten Internetabschaltungen technisch zu analysieren. Folgende Technologien sind dafür relevant und also solche Gegenstand genauerer Betrachtung:

- das Internet als Verbund von vielen autonomen Systemen (Kapitel 2)
- das Border Gateway Protocol (*BGP*) als wichtigstes Element zur Wegefindung im Internet zwischen autonomen Systemen (Kapitel 3)
  - BGP-Withdrawals als Folge von Internetausfällen
  - Geolocation Datenbanken, Regional Internet Registries und Network Telescopes als Datenquellen, um Internetausfälle zu beobachten
- Paketfilter als weiteres Werkzeug, um Internetzensur umzusetzen (Kapitel 4)

Diese Punkte werden ergänzt durch Kapitel 5, in welchem die in den Revolutionen angefallenen Daten analysiert werden, um anhand davon die vorher erklärte Theorie zur Zensur und zu Internetabschaltungen zu veranschaulichen.

2 Begriffe und Funktionsweise des Internets

Um die verschiedenen Möglichkeiten der beabsichtigten Internetabschaltung bzw. -zensur verstehen zu können, ist es erforderlich, die Funktionsweise, die Organisationsform und entsprechende Fachbegriffe des Internets zu kennen. Diese Dinge werden in diesem Kapitel erläutert.

2.1 Internet und autonome Systeme

Das Internet ist ein großes, dezentral organisiertes Computernetzwerk bestehend aus vielen kleineren Netzwerken, genannt autonome Systeme (engl. autonomous Systems, Abkürzung: *ASs*). Ein Internet-Service-Provider (Abkürzung: *ISP*), eventuell auch eine Universität oder eine größere Firma, stellt seinen Endkunden einen Zugang zum eigenen Netzwerk beziehungsweise AS zur Verfügung. Dieses ist mit

doi: 10.2312/NET-2013-02-1_01

mindestens einem anderen autonomen System anderer ISPs verbunden. Es besteht zwischen zwei beliebigen ASs immer eine Verbindung, entweder direkt oder indirekt über mehrere andere AS. Folglich existiert zwischen allen Endkunden immer mindestens eine Route. Das resultierende Netz wird Internet genannt. [4]

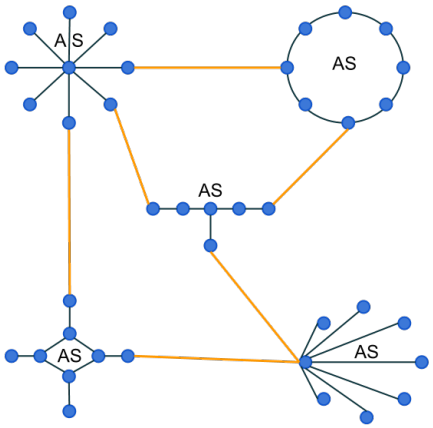


Abbildung 1: Schematische Darstellung von autonomen Systemen (ASs)

2.2 Routing

Wie auf Abbildung 1 ersichtlich besteht ein AS aus zwei Arten von vermittelnden Knotenpunkten (*Routern*): Welche mit Verbindungen nach außen und innen (für die vorliegende Arbeit genannt *Exterior Gateways*) und welche mit ausschließlich inneren Verbindungen. Exterior Gateways stellen die Schnittstellen eines AS zu anderen ASs und damit zum Rest der Welt dar. Somit sind diese Knotenpunkte für die globale externe Routenfindung zuständig. Die anderen dagegen dienen nur der internen Routenfindung innerhalb eines AS. Dementsprechend wird im Allgemeinen zwischen zwei Routingprotokollfamilien unterschieden: *Exterior Gateway Protocols* und *Interior Gateway Protocols*. Ersterer Protokollart ist das Border Gateway Protocol (*BGP*) zuzuordnen, welches als weltweiter Standard Anwendung im Internet findet. BGP wird in Unterkapitel 3.1 erläutert. [4] [5]

2.3 IP-Vergabe und Regional Internet Registries

Ein ISP beantragt für seine ASs jeweils IP-Adressbereiche und eine global eindeutige Nummer (*AS-Nummern*) bei der auf dem jeweiligen Kontinent zuständigen Regional Internet Registry (*RIR*) (diese bekommt diese Daten wiederum von der IANA zugewiesen, siehe dafür [6]) - die Zuständigkeiten der RIRs sind auf Abbildung 2 zu erkennen. Anschließend kann er die IP-Adressen an seine Endkunden weitergeben. Diese gehen damit ins Internet, um Daten weltweit zu senden und zu empfangen. Die AS-Nummern sind zwischen den verschiedenen ASs für die eindeutige Identifikation bei der Routenfindung der Daten (*Routing*) notwendig. Im Speziellen wird darauf im Unterkapitel 3.1 eingegangen. [5] [7]

Abbildung 2: Weltweite Zuständigkeiten von Regional Internet Registries (RIRs) [7]

3 BGP

In diesem Kapitel soll die externe Routenfindung genauer betrachtet werden, um zu sehen, wie das BGP-Routing im Internet zwischen den verschiedenen ASs funktioniert (Unterkapitel 3.1). Dabei wird auch erläutert, durch welche Eingriffe hier Internetzensur im Sinne von gezielten Internetabschaltungen ermöglicht wird (Unterkapitel 3.2). Zudem wird auf zwei Methoden für die externe Messung solcher Ereignisse (mittels Steuerungsdaten und Nutzdaten) eingegangen. Beispiele für die beschriebenen Zensurmaßnahmen finden sich im Kapitel 5. Alle diese Themen gehen, soweit nicht anders vermerkt, zurück auf eine wissenschaftliche Arbeit zum Thema Internetausfälle und ihre Beobachtbarkeit [1].

3.1 Funktionsweise von BGP-Routing

Im Internet sind die am BGP-Routing teilnehmenden Router die Exterior Gateways der ASs. Diese speichern in ihren Routingtabellen Einträge ab, welche alle möglichen Routen zum jeweiligen *IP-Adressraum*, der seinerseits eine Menge von IP-Adressen umfasst, enthalten. Diese Routen sind als Vektoren von AS-Nummern der ASs gegeben, die als vermittelnde Knotenpunkte auf dem Weg zum IP-Zieladressraum liegen.

Ein neues AS macht sich mittels dessen Exterior Gateway (auch *BGP-Router* genannt) im Internet global bekannt, indem es anderen ASs seine Existenz bekanntgibt (*BGP-Announcement*). Es übermittelt an benachbarte ASs (bzw. deren BGP-Router) im Wesentlichen folgende Informationen: seinen IP-Adressraum, seine AS-Nummer und alle durch ihn erreichbaren IP-Adressräume einschließlich der dazugehörigen AS-Routenvektoren (*UPDATE*-Nachricht). Die benachrichtigten BGP-Router ergänzen ihre Routingtabellen um die neuen IP-Adressräume und benachrichtigen selber wiederum ihre externen Nachbarn über die neuen Routeninformationen (*UPDATE*-Nachricht) und so fort, bis alle BGP-Router des Internets mit der neuen Information versorgt sind. Damit ist ein neuer IP-Adressraum von überall auf der Welt her ansprechbar.

Um den Online-Status gegenseitig zu überprüfen, teilen sich benachbarte BGP-Router diesen in regelmäßigen Zeitabständen durch KEEPALIVE-Nachrichten mit. Wenn die Zeitspanne zwischen zwei KEEPALIVE-Nachrichten von einem BGP-Router zu lang wird, so gilt dessen AS bzw. IP-Adressraum vom benachbarten Router aus betrachtet zunächst als nicht mehr erreichbar und der darauf verweisende

doi: 10.2312/NET-2013-02-1_01

Eintrag wird aus der BGP-Routingtabelle gestrichen (*BGP-Withdrawal*). Es kann danach jedoch wieder zu einem BGP-Announcement kommen, sofern es eine andere noch bestehende Verbindung zum AS gibt. Diese Verbindung kann gegebenenfalls indirekt über andere BGP-Router und ASs laufen - Abbildung 1 liefert eine Veranschaulichung für mehrere mögliche Routen zwischen verschiedenen ASs. Derartige Änderungen, wie das Löschen oder das Modifizieren von Einträgen, werden wiederum im Internet mittels UPDATE-Nachrichten verbreitet. [8]

3.2 BGP-Withdrawals und Internetzensur

BGP-Withdrawals können z.B. durch eine Trennung des BGP-Routers vom Netz durch Ziehen des Netzwerkkabels oder durch entsprechende Änderungen der Konfiguration des jeweiligen BGP-Routers ausgelöst werden. In der Folge ist das mit den BGP-Routern assoziierte AS und der entsprechende IP-Adressraum vom Internet abgetrennt. Das bedeutet, dass alle zugehörigen Endnutzer (Clients wie Server) andere Endnutzer außerhalb des ASs nicht kontaktieren bzw. nicht von ihnen kontaktiert werden können. Es gibt keine Möglichkeit, diese Internetsperre zu umgehen, wie das beispielsweise bei Paketfiltermechanismen denkbar wäre. Allerdings kann es sein, dass Teilnehmer innerhalb des ASs noch untereinander kommunizieren können, sofern den Routern innerhalb des AS noch normal funktionieren. Somit ist es möglich, gezielt ganze Gruppen von Endnutzern vom Internet abzukoppeln. Insofern kann diesbezüglich nicht nur von einem Internetzensur-, sondern auch von einem Internetabschaltungs- bzw. Internetausfallmechanismus gesprochen werden.

3.3 Globale Messbarkeit von BGP-Withdrawals

Es stellt sich die Frage, ob es Möglichkeiten gibt von außerhalb zeitnah herauszufinden, wo auf der Welt derartige Internetabschaltungen stattfinden. Dies ist insbesondere dann interessant, wenn großflächige Internetausfälle mit historischen Ereignissen wie Revolutionen oder Bürgerkriegen in Zusammenhang stehen. Das Vorgehen von Regierungen gegen die eigene Bevölkerung würde damit online global beobachtbar.

In den folgenden Unterkapiteln wird eine Methode zur Eingrenzung des zu überwachenden IP-Adressbereichs und zwei Messmethoden für die Überwachung von BGP-Withdrawals vorgestellt. Letztere sind in zwei Kategorien unterteilbar. Einerseits können BGP-UPDATE-Nachrichten an gut vernetzten BGP-Routern daraufhin analysiert werden, ob und welche BGP-Withdrawals darin sichtbar werden. Diese Art von Analyse bezieht sich nur auf Routingdaten, welche als solche ausschließlich zwischen BGP-Routern ausgetauscht werden (Steuerebene, engl. *Control Plane*). Andererseits können normale Pakete, sprich Nutzdaten, beim Empfänger beobachtet werden. Hat dieser die Pakete vorher nicht angefordert spricht man von unangefordertem Datenverkehr. Die kontinuierliche Analyse der Quell-IP-Adressen dieser Daten in einem entsprechend großem Maßstab mittels Network Telescopes machen BGP-Withdrawals sichtbar, wenn derartige Messungen über einen gewissen Zeitraum hinweg durchgeführt werden. BGP-Router dienen dabei weder als Sender noch als Empfänger für diese Art von Daten, sondern leiten diese nur weiter (Nutzdatenebene, engl. *Data Plane*). [9]

3.3.1 Ortung von BGP-Withdrawals

Um genaue Aussagen über Internetausfälle in einer bestimmten Region auf der Welt zu machen, ist es notwendig, den beobachteten IP-Bereich darauf einzuschränken. Dadurch wird eine weniger aufwändige und weitaus solidere Messung von BGP-Withdrawals möglich.
Die fünf RIRs (siehe Abbildung 2) stellen auf ihren Webseiten die Informationen darüber zur Verfügung, welchem Land welches AS bzw. welche AS-Nummer zugewiesen ist. Dabei muss sich ein AS nicht zwangsweise komplett im entsprechenden Land befinden, genauso wie ASs aus anderen Ländern Endnutzern im beobachteten Land Internet zur Verfügung stellen können. Jedoch fallen in vielen Ländern diese Ungenauigkeiten bei den Messungen kaum auf. Somit verfälschen sie nicht die Aussagen über plötzliche große Ausfälle vor Ort, wie in den angeführten Beispielen (Kapitel 5) zu beobachten ist.
Ist bekannt, welche ASs im zu beobachteten Land liegen, liefern öffentlich zugängliche Datenbankabfragen der RIRs über die AS-Nummern die IP-Adressbereiche, die diesen zugeordnet sind. Infolgedessen ist der IP-Bereich für ein Land für jedermann verfügbar, sofern kleinere Unwägbarkeiten ignoriert werden.
Darüberhinaus bieten kommerzielle Anbieter eigene Geolocation-Datenbanken an, welche zur Überprüfung herangezogen werden können. Beispiele hierfür werden in Unterkapitel 5.1 gegeben.

3.3.2 Messbarkeit der Routingdaten

Es soll nun vorgestellt werden, welche Datenquellen für die Routingdaten zur Verfügung stehen, welche Arten von Daten dies sind und welche Aussagen bezüglich BGP Withdrawals daraus ableitbar sind.

Wie zu Beginn dieses Kapitels bereits erwähnt, sind bei den Routingdaten die UPDATE-Nachrichten, die zwischen den BGP-Routern versendet werden, von Interesse. Ein Blick in diese Nachrichten gibt Aufschluss darüber, welche Veränderungen (Announcements und Withdrawals) in der jeweiligen Routingtabelle stattgefunden haben. Diese Nachrichten werden im Internet innerhalb der Control Plane zwischen durch Announcements bekanntgegebenen BGP-Routern ausgetauscht und in deren Routingtabellen diesen abgespeichert. Hierbei sei darauf hingewiesen, dass die Routingtabellen von BGP-Routern in Bezug auf die verschieden zugewiesenen IP-Adressräume vollständig, eindeutig und damit in der Tat sehr groß sind.

Es kann ein Ziel sein, zeitliche Verzögerungen zwischen BGP-Announcements/-Withdrawals und deren Bekanntwerden bei entfernteren BGP-Routern gering zu halten. Ist dies der Fall, so erscheint es sinnvoll, einen gut vernetzten BGP-Router für das Speichern der UPDATE-Nachrichten zu verwenden. Dies hat eine geringere Anzahl Hops zwischen Ziel und Empfänger zur Folge und damit auch geringere Latenzzeiten.

In der Praxis sind vor allem Forschungseinrichtungen sowie die RIRs an den besagten BGP-Daten interessiert. Die Daten werden benutzt, um BGP-Phänomene zu erforschen und das BGP weiterzuentwickeln. Besagte Einrichtungen stellen entsprechende Rohdatensätze zum Download bereit sowie weitere Tools, z.B. zur Analyse und Visualisierung. Es wird im Kapitel 5 auf konkrete Beispieldatenbanken hingewiesen.

doi: 10.2312/NET-2013-02-1_01

3.3.3 *Messbarkeit von Nutzdaten*

Im Gegensatz zu Routingdaten, die nur zwischen BGP-Routern ausgetauscht werden, machen Nutzdaten den eigentlichen Datenverkehr zwischen Endnutzern aus. Es werden Pakete zwischen zwei Rechnern, Sender und Empfänger, übertragen. Beide besitzen eine eindeutige IP-Adresse, um das Routing hin (Anfrage) und zurück (Antwort) zu ermöglichen. Das Fälschen der Absenderadresse nennt sich *IP-Spoofing* (siehe Abbildung 3). Die Folge davon ist eine Antwort des Empfängers hin zu einer anderen IP-Adresse als der des ursprünglichen Senders. Die dadurch im Internet verursachten Daten werden *Internet Background Radiation (IBR)* genannt [10]. Die IBR bildet eine gewisse kontinuierliche überall vorhandene Grundlast im globalen Datenverkehr und kann somit als eine Art Grundrauschen im Internet verstanden werden.

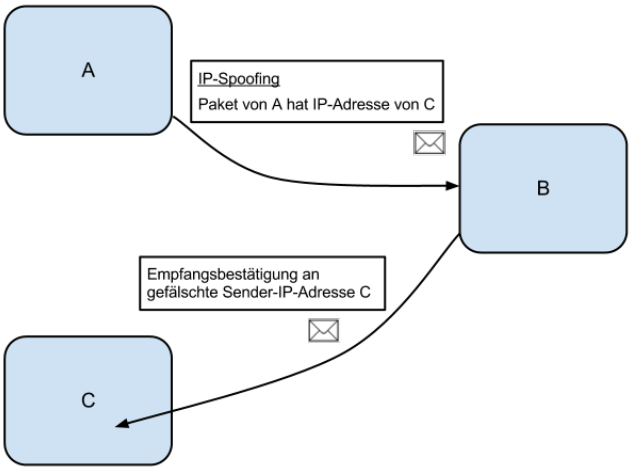


Abbildung 3: Darstellung von IP-Spoofing. Die Quell-IP-Adresse eines Pakets wird gefälscht. Die Bestätigung geht an einen anderen Sender.

Es gibt mehrere Ursachen für IBR:

- ***DoS-Attacken***
  Diese kommen zustande, indem gezielt eine sehr große Menge von Anfragen an Computersysteme gesendet werden [11] mit dem Ziel, diese unter Überlast zusammenbrechen zu lassen. Hat jede dieser Anfragen aufgrund von Spoofing eine gefälschte IP-Adresse, wie dies bei DoS-Attacken tatsächlich Anwendung findet (siehe z.B. [1] Kapitel 5.1.3), wird die Antwort an diese gesendet und damit IBR erzeugt. Dadurch tritt spontan für begrenzte Zeit eine große Menge IBR auf (auch *backscatter-traffic* genannt).
- **Scannen von IP-Adressräumen**
  Um herauszufinden, ob sich hinter einer beliebigen IP-Adresse ein Empfänger befindet, wird an diese eine Anfrage gestellt (z.B. via Ping). Bei Durchführung in großem Stil mit vielen IP-Adressen ist eine entsprechende Menge von IBR die Konsequenz. Insbesondere Würmer wie der *Conficker-Wurm* verfahren auf diese Art und Weise, wodurch neue Opfer aufgefunden und infiziert werden. Die Verbreitung des Conficker-Wurms ist groß genug, dass dieser global ein kontinuierliches Maß an IBR erzeugt (siehe [1] Kapitel 4.3). Diese Art von eigentlich unerwünschtem Datenverkehr erweist sich ironischerweise bei IBR-Messungen als besonders hilfreich.
- **Fehlerhafte oder schlechte Netzwerkkonfigurationen**
  Durch fehlerhafte Einstellungen, z.B. falsche IP-Einstellungen von DNS-Servern oder fehlerhafte Firmware bzw. schlechte Konfigurationen von Routern, die wiederum falsche IP-Zuweisungen zur Folge haben, kommt ebenso IBR auf.

Wird die IBR, die aus bestimmten IP-Adressbereichen stammt, z.B. denen eines Landes, über eine gewisse Zeit hinweg kontinuierlich aufgezeichnet, so weisen starke plötzliche Abnahmen von IBR auf Internetausfälle oder -abtrennungen zu den entsprechenden Zeitpunkten hin.

Die Werkzeuge für die Aufzeichnung von IBR sind sogenannte *Network Telescopes*. Diese benutzen IP-Adressräume, die keinen Rechnern zugewiesen sind (*Darknets*), um die daran adressierten Pakete und deren Quell-IP-Adresse aufzuzeichnen. Auch wenn diese Adressräume nur einen sehr kleinen Bruchteil des gesamten Internets ausmachen können, so konnte in Experimenten herausgefunden werden, dass Internetausfälle sich in IBR-Messungen innerhalb von Darknets via Network Telescopes deutlich niederschlagen (siehe [1] Kapitel 5). In der Konsequenz sind Internetabtrennungen über Nutzdatenanalyse in Darknets global und zu jeder Zeit sichtbar, wobei insbesondere Würmern eine Schlüsselrolle zukommt. Dies wird in den Beispielen dieser Arbeit (Kapitel 5) gezeigt werden.

4 Paketfilter

Auch wenn BGP-Withdrawals kein triviales Themengebiet sind, sind diese doch leicht durch Ziehen des Netzwerksteckers zu erzeugen. Der Effekt mag für Regierungen interessant sein, um schnell und unkompliziert eine ganze Bevölkerung vom Internet abzutrennen. Allerdings lässt sich mittels dieser Methode nicht beliebig nach Quell-IP-Adresse differenzieren, so dass schnell ein ganzer IP-Adressblock, aber nicht spezielle IP-Adressen vom Internet abgetrennt werden können. Insofern ist eine spezifischere Auswahl an Institutionen, die in einem Land besser online bleiben, nicht ohne weiteres möglich: wichtige Unternehmen, Banken, Stadtwerke und viele weitere sind auf das Funktionieren einer modernen Kommunikationsinfrastruktur angewiesen, um ein Land nicht vollends im Chaos versinken zu lassen. Ein weiteres Manko der der BGP-Withdrawals ist zudem die fehlende Möglichkeit, lediglich den Zugriff auf bestimme Bereiche des globalen Internets zu verhindern. In Anbetracht der wachsenden Rolle von sozialen Netzwerken wie Facebook und Twitter zwecks Nachrichtenaustausch und Demonstrationsorganisation erscheint eine entsprechende Restriktion des Zugriffs auf derartige Plattformen für totalitäre Regierungen ebenso als attraktiv [12]. Dafür sind Paketfilter das geeignete Mittel, deren Funktionsweise im Folgenden kurz dargelegt wird.

4.1 Funktionsweise

Das Ziel eines Paketfilters ist es, den in Form von Paketen in und aus einem Netzwerk kommenden Datenverkehr nach bestimmten Kriterien - z.B. Quell- und Ziel-IP-Adresse - zu filtern. Dies wird durch eine Software realisiert, die als Teil einer Firewall auf Routern oder anderen Netzwerkgeräten

doi: 10.2312/NET-2013-02-1_01

zum Einsatz kommt. Dabei wird in Filterregeln definiert, welche Pakete durchgelassen und welche verworfen werden. Dafür wird jedes eintreffende Paket vom Filter geöffnet und dessen Inhalt auf die angegebenen Kriterien überprüft. [13]

4.2 Einsatzmöglichkeiten

Solche Systeme werden standardmäßig in sehr vielen Netzwerken benutzt und dienen in erster Linie der Sicherheit, da dadurch unerwünschter Zugriff von außen ins eigene Netz unterbunden werden kann. Erweiterte Paketfilter können ebenso als Kinderschutz zu Hause eingesetzt werden, indem nicht nur nach Quell- oder Zieladresse von Seiten mit jugendgefährdenden Inhalten, sondern auch nach Inhalt selbst, z.B. bestimmten Stichworten, gefiltert wird. Nach unter anderem genau diesem Prinzip funktionieren auch sehr große Zensurinfrastrukturen wie beispielsweise in China [14]. In dieser Arbeit wird in Unterkapitel 5.3 im Zusammenhang mit der Revolution in Libyen auf eine weitere Anwendung eines Paketfilters eingegangen.

5 Beispiele

Innerhalb dieses Unterkapitels werden nun Beispiele dargelegt, die zeigen sollen, wie sich Internetausfälle praktisch in Routing- und Nutzdaten bemerkbar gemacht haben. Es werden zwei historische Ereignisse angeführt: die Revolutionen in Ägypten und Libyen. Beide fanden im Jahre 2011 im Zuge des arabischen Frühlings statt. Bei beiden Ereignissen wurde auf Anlass der damaligen Regierungen das Internet abgeschaltet (siehe dazu auch Unterkapitel 3.2). In welchen Datenquellen sich die hieraus resultierten BGP-Withdrawals niedergeschlagen haben ist Bestandteil von Unterkapitel 5.1. Im Anschluss daran werden die Daten für jedes Land analysiert - siehe Ägypten (Unterkapitel 5.2)) und Libyen (Unterkapitel 5.3).

5.1 Datenquellen

Um Routing- und Nutzdaten aus bestimmten Ländern beobachten zu können, müssen die den Ländern zugewiesenen IP-Adressräume herausgefunden werden (siehe Ortung von BGP-Withdrawals (Unterkapitel 3.3.1)).

5.1.1 *Ortung*

Da sowohl Libyen als auch Ägypten in Afrika liegen, ist somit AfriNIC [15] die zuständige RIR. Dessen Datenbanken geben Aufschluss über die ASs, welche im jeweiligen Land liegen. Zusätzlich sind diese Länderzuweisungen durch weitere unabhängige *Geolocation Datenbanken* überprüfbar, wie zum Beispiel die von MaxMind [16]. In den vorliegenden Beispielen waren die Unterschiede dieser beiden Datenbanken allerdings klein und haben für die in dieser Arbeit gemachten Beobachtungen keine Relevanz (für genaueres siehe [1], Kapitel 4.1.1).

5.1.2 *Routingdaten*

Die Datenquellen für BGP-UPDATES, welche weltweite Informationen von BGP-Withdrawals enthalten, werden einerseits von der University of Oregon respektive dessen RouteViews Projekt [17], andererseits von RIPE NCC bzw. RIPEstat zur Verfügung gestellt [18]. RIPE NCC stellt zudem für die bei beiden Ereignissen gemachten BGP-Beobachtungen separate Webseiten mit Daten und Analysen parat [2] [3]. Die Kombination dieser beiden Datenbanken - RouteViews und RIPE NCC - ergibt die Grundlage der analysierten BGP-UPDATES (siehe [1], Kapitel 4.1.2).

5.1.3 *Nutzdaten*

Für die Aufzeichnung von IBR (siehe Unterkapitel 3.3.3) in den folgenden Beispielen wird auf die Daten vom UCSD Network Telscope des Internetverbunds CAIDA [19] (Cooperative Association for Internet Data Analysis) zurückgegriffen. Das davon beobachtete Darknet macht 1/256tel des gesamten IPv4-Adressraums aus (siehe [1], Kapitel 4.3). Dieser Adressraum mag sehr klein im Verhältnis zum gesamten Internetadressraum erscheinen. Es zeigt sich jedoch in den betrachteten IBR-Messungen, dass die Internetausfälle darin eindeutig sichtbar werden.

5.2 Revolution Ägypten

Die Revolution Ägyptens vollzog sich in den Anfangsmonaten des Jahres 2011. Sie wurde von großen Unruhen und Protesten im Land begleitet, welche vor allem über soziale Netzwerke wie Facebook organisiert wurden [20]. Infolgedessen ging die ägyptische Regierung massiv gegen die eigene Bevölkerung vor [21]: beinahe das ganze Land wurde am 27.01.2011 kurz vor Mitternacht vom Internet abgetrennt. Nur wenige wichtige Institute, wie z.B. die ägyptische Börse, blieben von außerhalb des Landes erreichbar. Dieses Ereignis ist von historischer Bedeutung, da es die erste beabsichtigte Abtrennung eines ganzen Landes vom Internet war [22].

5.2.1 *Analyse der Routingdaten*

Erste kleinere Anzeichen für den Verlust von IPv4-Internetrouten nach Ägypten, sprich BGP-Withdrawals, wurden ab 20:30 Uhr UTC von den BGP-Datenquellen aufgezeichnet (siehe Abbildung 4). Nach wenigen sehr kleinen Withdrawals wurden die meisten ägyptischen IPv4-Adressen auf einen Schlag zwischen 22:15 Uhr und 22:40 Uhr von außen unerreichbar. Die neueren - und standardmäßig von den meisten Nutzern nicht verwendeten - IPv6-Adressen waren jedoch davon unberührt und blieben damit während des gesamten Ausfalls online.

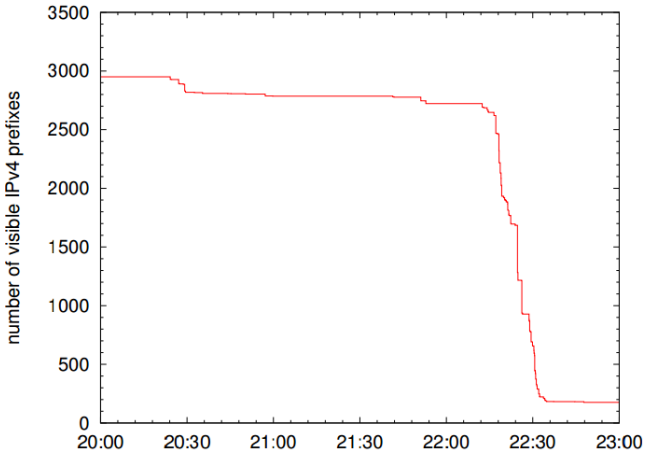


Abbildung 4: Darstellung der BGP-Withdrawals im Zuge der Internettrennung von Ägypten am 27.1.2011 [1]

doi: 10.2312/NET-2013-02-1_01

Der Internetausfall hielt an bis zum 2.2.2011 am Morgen (siehe Abbildung 5). Die ersten vorher getrennten IPv4-Adressen waren ab 9:30 Uhr wieder verfügbar. Um 10:00 Uhr war fast das ganze Land wieder online, bis um 11:45 Uhr wurde die ursprüngliche Konnektivität vollständig wiederhergestellt.
Anhang der Abnahme der erreichbaren IPv4-Adressen lässt sich feststellen, dass das Offline- und Onlineschalten Ägyptens nicht viel Zeit beanspruchte und jeweils innerhalb einer halben Stunde fast vollständig vollzogen wurde.

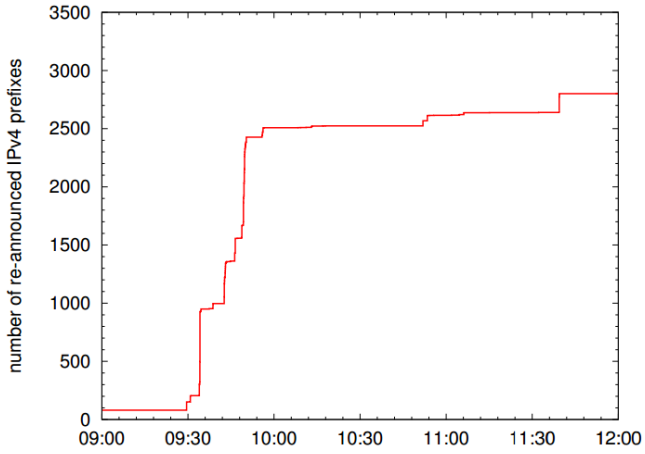


Abbildung 5: Darstellung der BGP-Announcements der Internettrennung in Ägypten am 2.2.2011 [1]

5.2.2 *Analyse der Nutzdaten*

Es wurde bereits in Unterkapitel 3.3.3 erklärt, inwieweit plötzliche Schwankungen in den Nutzdaten, genauer gesagt in der IBR, aus beobachteten IP-Adressbereichen auf BGP-Withdrawals hinweisen. Abbildung 6 ist ein Diagramm, welches die von CAIDAs Network Telescope gemessene IBR aus Ägyptens IP-Adressräumen über die Zeit des Internetausfalls hinweg kontinuierlich darstellt.

Bei dessen Betrachtung fallen die sinus-artigen Datenverkehrsschwankungen auf, die auf wechselhafte Benutzung des Internets bei Tag und bei Nacht hinweisen. Darüberhinaus ist ein deutliches Abnehmen der IBR offensichtlich, beginnend kurz vor dem 28.1.2011 und anhaltend bis zum 2.2.2011. Diese Zeitspanne entspricht der in den vorigen BGP-Veränderungen beschriebenen Offlinezeit Ägyptens. Somit ist dies ein deutlicher Hinweis dafür, dass IBR-Messungen in Darknets via Network Telescopes Aufschluss über Internetausfälle geben können. Dies ist im Gegensatz zur Messung der BGP-Routingdaten eine in diesem Maßstab erstmals gemachte Beobachtung, was als besondere Leistung des Papers "Analysis of Country-wide Internet Outages Caused by Censorship" [1] hervorzuheben ist. Das verdeutlicht, wie viel IBR aus einem relativ kleinem Adressraum ins gesamte Internet verteilt wird. Eine sehr kleine Teilmenge - rund 1/256tel, die Größe des überwachten Darknets - dieser extrem gestreuten Daten reicht vollkommen aus, um stichhaltige Aussagen über Interneterschütterungen weltweit machen zu können. Besonders interessant ist hierbei die Differenzierung der Datensorten, aus welchen die IBR besteht.

Wie aus Abbildung 7 hervorgeht, kommt die IBR während des Internetausfalls ausschließlich von mit dem Conficker-

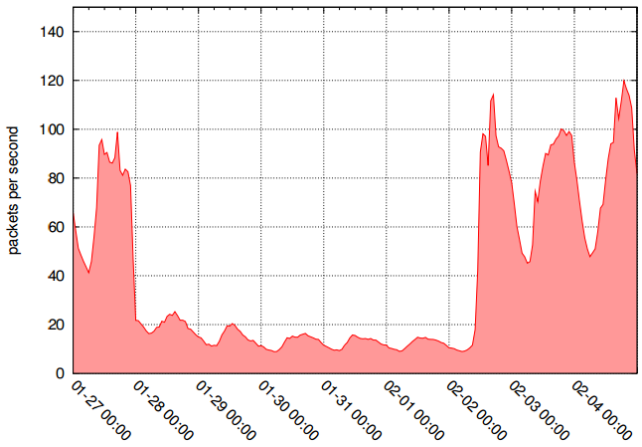


Abbildung 6: Die während der Internettrennung aufgezeichnete IBR aus Ägypten [1]

Wurm infizierten Rechnern, welche nicht durch die Regierung vom Internet getrennt wurden. Hier wird einem besonders stark vor Augen geführt, wie groß die Datenbelastung fürs Internet durch den Conficker-Wurm ist. Sogar ein kleiner Bruchteil des ägyptischen Internets ist so stark mit dem Wurm durchsetzt, dass dieser global aus verhältnismäßig kleinen Darknets heraus messbar ist.
Der Backscatter-Traffic befindet sich fast ununterbrochen auf 0-Niveau mit kurzen, aber sehr ausschlagenden Ausnahmen vom 2. bis 4.2.2011. Dies weist auf DoS-Attacken hin, die während der ägyptischen Revolution gegen das Regime ausgeführt wurden. Diese Attacken stehen in Zusammenhang mit Ankündigungen der Hackergruppe Anonymous, welche in einem kurz vorher veröffentlichten Brief derartige Attacken ankündigte [23].
Alle weiteren Internetdaten (subsumiert unter "other") sinken auf ein verhältnismäßig sehr kleines Niveau ab.

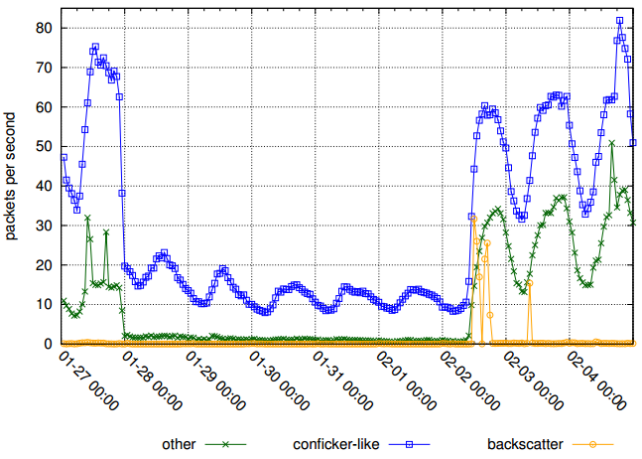


Abbildung 7: Differenzierung der IBR nach Datenverkehr durch Conficker, Backscatter und anderen Ursachen [1]

doi: 10.2312/NET-2013-02-1_01

5.3 Revolution Libyen

In Libyen begannen erste Proteste mit dem Aufkeimen der ägyptischen Revolution. Die Proteste steigerten sich zunehmends, bis sich am 16.2.2011 ein Bürgerkrieg zwischen der Regierung und oppositionellen Gruppen ankündigte [12]. In der Folge koppelte die Regierung am 18.2.2011 fast ganz Libyen über Nacht vom Internet ab. Dies wurde ein weiteres Mal in der darauf folgenden Nacht wiederholt. Darüberhinaus wurde fortan von erschwertem Zugang zu sozialen Medien wie Facebook und Twitter berichtet [12]. Am 2.3.2011 startete eine weitere Offensive der Regierung [24]. Infolgedessen kam es zu Luftangriffen gegen die Opposition und am 3.3.2011 zu einem weiteren Internetshutdown, der diesmal knapp 4 Tage anhielt. [1]

Im Beispiel von Libyen ist der zweite und der dritte Internetausfall von anderer technischer Natur als der aus Ägypten.

5.3.1 *Analyse der Routingdaten*

Da für den dritten Ausfall keine BGP-Routingdaten angefallen sind, werden nur die ersten beiden anhand der BGP-Withdrawals in Abbildung 8 dargestellt. Diese sind in der Abbildung differenziert nach den ASs, in welchen IP-Adressen zurückgezogen wurden. Für die Analysen in dieser Arbeit wird nur das landesweit grösste AS (LyStateAS) betrachtet.

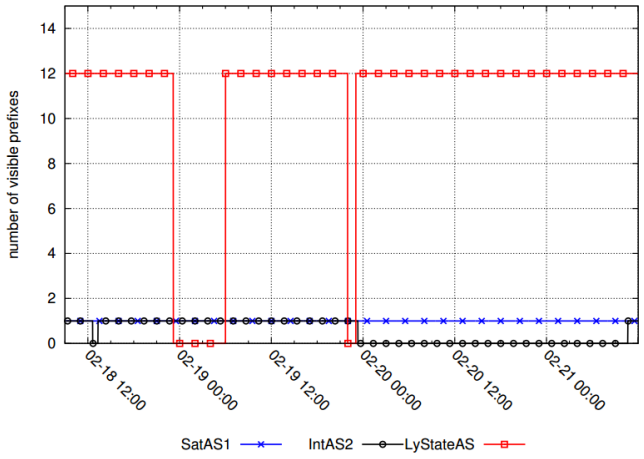


Abbildung 8: BGP-Withdrawals vom 18. und 19.2.2011, differenziert nach drei libyschen ASs. SatAS1 versorgt Libyen via Satellit mit Internet. [1]

Ersichtlich sind die Ausfälle, die jeweils am 18. und 19.2.2011 kurz vor Mitternacht beginnen, anhand der Anzahl der erreichbaren IPv4-Adressräume. Waren beim ersten die IP-Adressen ca. sechs Stunden entzogen, so war dies beim zweiten nur ungefähr eine Stunde der Fall. Die BGP-Withdrawals und -Announcements der IP-Adressräume passierten jeweils auf einen Schlag.

5.3.2 *Analyse der Nutzdaten*

Eine Analyse der vom Network Telescope gemessenen IBR (siehe Abbildung 9) des gleichen Zeitraums zeigt ein vollständigeres Bild.

Beide Ausfälle sind hier durch die beiden größeren Lücken im Datenverkehr zu erkennen. Während die hier betrachtete

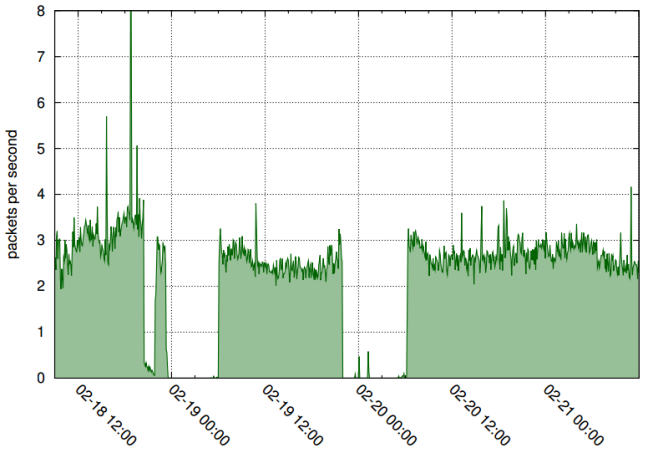


Abbildung 9: IBR aus Libyen, aufgezeichnet vom 18. bis 21.2.2011. Die zwei großen Lücken spiegeln die zwei Internetausfälle wieder. [1]

Dauer des ersten Ausfalls der aus Abbildung 8 entspricht, liegt die Sache beim zweiten anders.

Dessen Dauer beträgt in Abbildung 9 ca. 8 Stunden, was um ein Vielfaches länger ist, als es die vorher betrachteten BGP-Routingdaten in Abbildung 8 vermuten ließen. Zudem entspricht diese längere Zeitspanne der tatsächlichen Offlinezeit. Das bedeutet, dass eine weitere Zensurmaßnahme jenseits von BGP-Withdrawals zum Einsatz gekommen sein muss, wenn der Datenverkehr selbst nach dem Wiederherstellen der BGP-Routen gleichermaßen reduziert blieb.

Dies weist auf den Einsatz eines Paketfilters (siehe dafür Kapitel 4) hin, welcher kurz nach dem Entzug der BGP-Routen installiert und aktiviert worden ist. Er kommt an der Stelle zur Geltung, wo die BGP-Routen wiederhergestellt werden - Abbildung 8, 20.2. kurz vor 00:00 Uhr - und ist fortan für den Internetausfall - wie in Abbildung 9 ab 00:00 Uhr ersichtlich - verantwortlich. Was seine Funktionsweise angeht, so verwirft er alle ein- und ausgehenden Pakete. Die Folge ist, dass die Endnutzer, welche normalerweise über den entsprechenden Router online gehen, über diesen keine Pakete ins Internet verschicken oder von dort empfangen können. Der Effekt ist für den Endnutzer nicht ohne Weiteres von einem BGP-Withdrawal zu unterscheiden. Nur in den hier abgebildeten Messungen ist der Unterschied durch den Vergleich der BGP-Routingdaten mit der IBR sichtbar.

5.3.3 *Anmerkungen*

Es sei der Vollständigkeit halber auf zwei Details verwiesen:

- Die mediale Berichterstattung erwähnte schon vor den Internetabschaltungen einen erschwerten Zugriff auf soziale Netzwerke [12]. Dies ist ein Hinweis auf weitere Paketfilter, die bereits vorher zum Einsatz kamen und deren Filterregeln entsprechende Webseiten blockierten.
- Beim dritten Internetausfall Libyens kommt annähernd derselbe Paketfilter wie beim zweiten zum Einsatz, BGP-Withdrawals finden dort allerdings nicht statt. Für Details wird auf das Referenzpaper verwiesen (siehe [1], Kapitel 5.2).

doi: 10.2312/NET-2013-02-1_01

6 Schluss

Es wurden im Zusammenhang mit Zensur zwei Internetausfallarten sowie Methoden zu deren Beobachtung beschrieben. Dabei wurde gezeigt, dass sich Ausfälle, die BGP-Withdrawals zur Folge haben, von Ausfällen, die durch Paketfiltermechanismen zustande kommen, unterscheiden. Erstere kommen einem Abtrennen des Internetkabels an BGP-Routern gleich und sind global durch Änderungen in den BGP-Routingtabellen zu beobachten. Beide hingegen können von Network Telescopes bemerkt werden, da die IBR in Darknets durch beide Ausfallarten sichtbar abnimmt. Dem Conficker-Wurm, der ein erhebliches Datenvolumen im Internet verursacht, kommt hierbei eine besondere Bedeutung zu. Durch die unterschiedlichen Messergebnisse auf der BGP-Steuerungsebene ist es möglich beide Ausfallarten voneinander zu unterscheiden.
Darüberhinaus hat das Beispiel Libyens gezeigt, dass mit zunehmender Erfahrung der Zensoren Paketfilter bevorzugt eingesetzt werden. Dies ist auf die Filterregeln zurückzuführen, welche eine feinere Anpassung der Zensur und des Ausfalls ermöglichen. In die Zukunft projiziert bedeutet dies, dass bei derartigen Ereignissen keine BGP-Withdrawals mehr zu erwarten sind und die Beobachtung derartiger Online-Erschütterungen über Network Telescopes erfolgen müssen, wie dies im Referenzpapier das erste Mal in der Form durchgeführt wurde [1].

Literatur

[1] A. Dainotti, C. Squarcella, E. Aben, K. C. Claffy, M. Chiesa, M. Russo, A. Pescapé, "Analysis of Country-wide Internet Outages Caused by Censorship", ACM

[2] RIPEstat - Egyptian Internet Outage. https://stat.ripe.net/events/egypt. Abgerufen am 24.9.2012

[3] Unsolicited Internet Traffic from Lybia - RIPE Labs. https://labs.ripe.net/Members/emileaben/unsolicited-internet-traffic-from-libya. Abgerufen am 24.9.2012

[4] RFC 1930 - Guidelines for creation, selection, and registration of an Autonomous System (AS). IETF. Network Working Group. March 1996. http://tools.ietf.org/html/rfc1930. Abgerufen am 29.10.2012

[5] RFC 1771 - A Border Gateway Protocol 4 (BGP–4). IETF. Network Working Group. March 1995. http://tools.ietf.org/html/rfc1771. Abgerufen am 29.10.2012

[6] IANA - Autonomous System (AS) Numbers. http://www.iana.org/assignments/as-numbers/as-numbers.xml. Abgerufen am 29.10.2012

[7] IANA - Number Resources. http://www.iana.org/numbers. Abgerufen am 29.10.2012

[8] RFC 4271 - A Border Gateway Protocol. IETF. Network Working Group. January 2006. http://tools.ietf.org/html/rfc4271. Abgerufen am 29.10.2012

[9] RFC 3746 - Forwarding and Control Element Separation (ForCES) Framework. IETF. Network Working Group. April 2004. ftp://ftp.rfc-editor.org/in-notes/rfc3746.txt. Abgerufen am 29.10.2012

[10] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, L. Peterson. Characteristics of Internet background radiation. In Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, IMC '04, Seiten 27–40, New York, NY, USA, 2004. ACM.

[11] Internet Denial-of-Service Considerations. IETF. Network Working Group. November 2006. http://tools.ietf.org/html/rfc4732. Abgerufen am 29.10.2012

[12] Aufstände in Arabien - Gaddafi kappt Facebook und Twitter. 19.02.2011. rom/dpa/Reuters/dapd. http://www.spiegel.de/politik/ausland/aufstaende-in-arabien-gaddafi-kappt-facebook-und-twitter-a-746597.html. Abgerufen am 24.9.2012

[13] Network Layer Firewall. WanRedundancy.org. http://www.wanredundancy.org/resources/firewall/network-layer-firewall. Abgerufen am 24.9.2012

[14] Bericht: Computer sollen in China nur noch mit Filtersoftware verkauft werden. Andreas Wilkens. 08.06.2009. heise online. http://www.heise.de/newsticker/meldung/Bericht-Computer-sollen-in-China-nur-noch-mit-Filtersoftware-verkauft-werden-179181.html. Abgerufen am 24.9.2012

[15] AfriNIC (Regional Internet Registry für Afrika). http://www.afrinic.net/en/services/whois-query. Abgerufen am 24.9.2012

[16] MaxMind. MaxMind GeoLite Country: Open Source IP Address to Country Database. http://www.maxmind.com/app/geolitecountry. Abgerufen am 24.9.2012

[17] University of Oregon. University of Oregon Route Views project. http://www.routeviews.org. Abgerufen am 24.9.2012

[18] RIPE NCC: Routing Information Service (RIS). http://www.ripe.net/data-tools/stats/ris/routing-information-service. Abgerufen am 24.9.2012

[19] UCSD Network Telescope http://www.caida.org/projects/network_telescope/. Abgerufen am 24.9.2012

[20] Lena Jakat. Sueddeutsche.de - 31.01.2011 - Krise in Ägypten - Die Kinder des 6. April und der Tag der Entscheidung. http://www.sueddeutsche.de/politik/krise-in-aegypten-die-kinder-des-april-rufen-zum-protest-1.1053426. Abgerufen am 24.9.2012

[21] Sueddeutsche.de - 28.01.2011. http://www.sueddeutsche.de/politik/massenproteste-in-aegypten-angekuendigt-tag-des-zorns-beginnt-mit-festnahmen-1.1052282. Abgerufen am 24.9.2012

[22] Ägypten ist offline und ohne Mobilfunk [4. Update]. heise online. 28.01.2011. http://www.heise.de/newsticker/meldung/Aegypten-ist-offline-und-ohne-Mobilfunk-4-Update-1179102.html. Abgerufen am 24.9.2012

[23] OPERATION EGYPT - ANONYMOUS PRESSE-MITTEILUNG. AnonNews.org. 26.01.2011. 27.01.2011. http://anonnews.org/?p=press&a=item&i=299. Abgerufen am 24.9.2012

doi: 10.2312/NET-2013-02-1_01

[24] Libyan Islamists seize arms, take hostages. 2012 AFP. 21.02.2011. http://news.smh.com.au/breaking-news-world/libyan-islamists-seize-arms-take-hostages-20110221-1b19c.html. Abgerufen am 24.9.2012

doi: 10.2312/NET-2013-02-1_01

Controlled Internet Outage Monitoring

Christian Köpp
Supervisor: Lukas Schwaighofer
Seminar Future Internet WS12/13
Chair for Network Architectures and Services
Department of Computer Science, Technical University of Munich
Email: christian.koepp@cs.tum.edu

ABSTRACT

This paper is about the measurement of Internet outages with a focus on the Border Gateway Protocol, as one of the most common routing protocols. First some background information about the protocol and its tasks are described, followed by a concrete example of last year's events in Egypt. Thereafter a self-made analysis regarding the earthquake in New Zealand in 2011 is presented. Finally a conclusion about Internet outages and their monitoring with data collected by the Border Gateway Protocol is given. The conclusion also includes a short passage about recent papers and methods involving Internet outage monitoring in current academic research with a focus on the role of BGP within those.

Keywords

Inter-Domain Routing, Outages, Connectivity Disruption, Border Gateway Protocol, BGP, Earthquake, Censorship

1. INTRODUCTION

Routing protocols are indispensable on the Internet nowadays. Their goal is to guarantee the ability of communication between different networks and they attempt to find an efficient way to get packages from its source to its destination. Furthermore modern routing protocols often counteract events like natural disasters and even political censorship actions, as they are designed to be resilient against communication interruptions. Due to this routing protocols are trying to circumvent these interruptions and to establish new paths to those effected networks. Nevertheless there are events, either triggered by political or natural issues, which lead to serious connectivity disruptions or even result in an outage of a whole geographical area.

Natural disasters like the massive earthquake in Japan with 8.9 magnitude on March 11th of 2011 can lead to outages. For example the mentioned earthquake caused the destruction of technical equipment, especially undersea cables necessary for connecting the Japanese islands with the Asian coastline [1]. But also smaller and more regional events like bad weather can lead to a temporary Internet outage in smaller areas [2].

But natural incidents are not the only events that can cause such critical Internet outages. Governments and dictatorships are able to force (state-owned) providers to cut their lines and to disable their services as a method of censorship. Political issues like these were, for example, observed during the uprising in Egypt and Libya at the beginning of 2011 [3]. Another incident happened in 2008 during Pakistan's try to deny access to YouTube. During this operation a Pakistani Internet service provider unintentionally announced routes globally and hijacked the YouTube traffic [4]. The most recent example is the ongoing civil war in Syria, which led to a temporary Internet outage of Syria in June 2012 [5].

In this paper we will only have a look at events related to the Border Gateway Protocol and its behaviour. BGP is the major protocol for inter-domain-routing between international providers. Due to the popularity of BGP, official Autonomous Systems are expected to be able to communicate with BGP. This is the reason why changes made through the protocol can get propagated throughout the world and therefore lead to global scale changes of routing decisions of core networks. Finally such path changes through BGP can result in different packet flows and connectivity of Internet hosts all around the world.

2. BACKGROUND

This chapter describes the techniques, protocols and standards of the major routing and reachability information sharing issues currently used on the Internet.

2.1 Autonomous Systems

Nowadays the Internet is a giant interconnected network consisting of a huge amount of independent networks with different sizes and geographical locations. An Autonomous System (AS) is a network, which consists of a collection of Internet Protocol routing prefixes under the control of at least one network operator [6]. Typically providers of such networks are Internet Service Providers (ISP) and big companies with multiple connections to different networks. As part of the official process to get registered as an AS, an organisation has to request an Autonomous System Number (ASN). This 32 bit ASN [7] is used to uniquely identify an AS and is assigned to an organisation by the Internet Assigned Numbers Authority (IANA) and its regional representatives, the Regional Internet Registries (RIR).

2.2 Border Gateway Protocol

The Border Gateway Protocol is a protocol for exchanging network layer reachability information (NLRI) between and within autonomous systems. The first version of BGP was introduced in 1989. Until now there have been several changes to the standard and therefore different versions of BGP with version 4 being the current one [8].

doi: 10.2312/NET-2013-02-1_02

Routers communicating through BGP are called peers as they are equally privileged participants during the communication. To establish a certain reliability of the messages, BGP makes use of the Internet Protocol and the Transport Control Protocol (TCP/IP). Traditionally BGP can be found on TCP port 179.

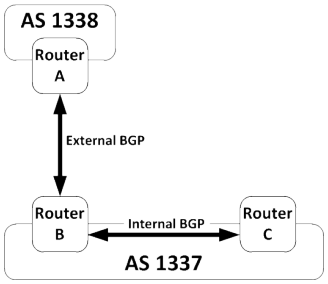


Figure 1: Difference between EBGP and IBGP

If the Border Gateway Protocol is used to share NLRI between autonomous systems it is called External BGP (EBGP). Naturally those exterior exchanges are done between edge routers of different AS while interior information exchanges do not necessarily happen only between edge routers of an AS. An exchange within an AS and between its routers is called Internal BGP (IBGP) like figure 1 demonstrates. This paper is focusing on EBGP as it analyses outages of whole IP ranges within a global scale.

2.2.1 Making routing decisions with BGP4

A BGP4 router is equipped with a special kind of database, the Routing Information Base (RIB) which is made up of two different parts. The central database within the RIB is the routing table. This is the place where all the information needed for making routing decisions is stored. A policy, referred to as local RIB, defines how incoming information of adjacent routers is treated and which information is passed on to other adjacent peers afterwards. The local RIB is also responsible for writing received information to the routing table. Therefore it is possible to call the local RIB a kind of configuration for a router. Figure 2 explains the relations of the different sections involved in routing decisions in a graphical way. Since companies typically try to minimize their costs, this configuration is often not purely based on technical facts but on economical reasons. These costs are usually based on the amount of traffic transmitted to another network and can therefore lead to slower but more economical routing decisions [9]. Due to this reason not all routes learned through BGP are automatically stored within the routing table itself. Furthermore, there is always exactly one route to an unique IP prefix stored in the routing table.

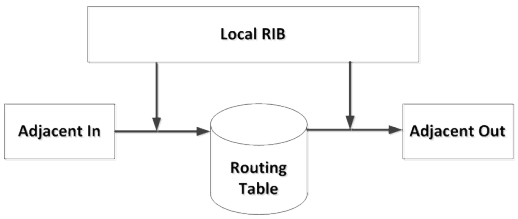


Figure 2: Scheme of the Routing Information Base

Another fundamental rule which all routing decisions are based on is the usage of the most specific IP prefix available. Technical manuals are calling this strategy the longest prefix match [10]. The length of the prefix can easily be determined by looking at the length of the subnet-mask of an entry in the RIB. Figure 3 shows an example usage of this strategy.

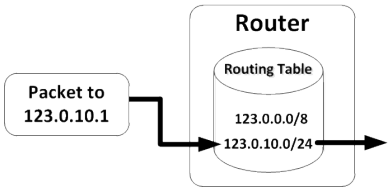


Figure 3: Example usage of the longest prefix match

2.2.2 Information exchange through BGP4

Every BGP message, regardless of its type, consists of a 19-byte header followed by a payload of variable length. Within the header block there is a type field which contains one out of five defined BGP message types. In this paper just the types `OPEN`, `UPDATE`, `NOTIFICATION` and `KEEPALIVE` are of particular interest [8]. Therefore the `ROUTE-REFRESH` message type is not explained nor used in this paper [11].

The first action two peers have to do to exchange reachability information is the establishment of a TCP channel between them. The peers try to keep this connection alive during their whole uptime. If a BGP session between two peers exists they are called neighbours. After the TCP handshake is done the initiator of the BGP session sends an `OPEN` message to the other peer. In the payload of the message there is the unique ASN of which the source router is part of. It also contains a unique ID of the sending router. Usually the router ID is the IP address of the router if BGP is used in combination with TCP/IP. An accepted `OPEN` message is indicated by responding with a `KEEPALIVE` message as this message type consists only of the header block and an empty payload. If both peers sent their `OPEN` messages and received an `KEEPALIVE` message afterwards, the exchange of reachability information is able to start.

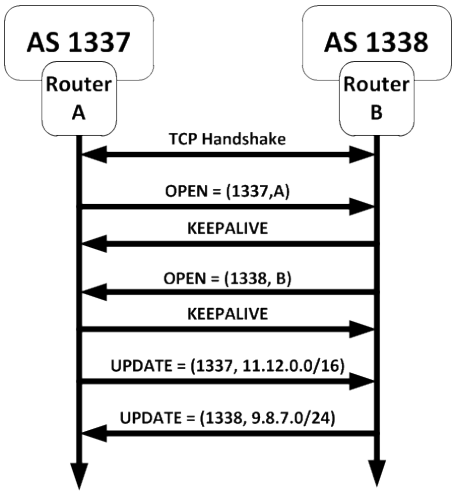


Figure 4: Example BGP session

doi: 10.2312/NET-2013-02-1_02

But the KEEPALIVE messages do have another important function within the protocol. They are used as a kind of heartbeat message between two peers. If a peer doesn't receive a KEEPALIVE message in a certain defined timespan the other peer is assumed to be not available. As a result of this, all routes announced by this peer that were stored in the RIB are treated as invalid and are not used anymore. The router immediately tries to figure out the best route to the affected IP ranges. If there is an alternative route available the router propagates this new route to its neighbours. If the router was not able to find another route a withdrawal is sent to the neighbours as the router is not able to forward packages to this IP range any more.

With the UPDATE message type it is possible for a router to announce new routes to a certain IP range. Included in the message there is always a detailed path-information with all AS on the way to the destination IP range. So other routers can use the whole path to determine if the newly announced route is better than their current one. Therefore an UPDATE message can also be seen as the promise of a peer to forward any datagrams towards the announced prefix.

Furthermore, the UPDATE message can also be used to withdraw previously announced routes. Due to different fields within those messages, it is perfectly valid and possible to withdraw routes and announce new ones in one single message. IP ranges are always given in CIDR [12] notion like 127.0.0.0/8. It is also possible to do an implicit withdrawal which is done when a new route to an IP range is announced even if there still is an existing one stored on the router. In this case the old route is overwritten with the new one.

A NOTIFICATION message is only sent from a peer to indicate that there was some kind of error in either receiving or processing the last BGP message. Therefore this message type contains fields like error code, error subcode and optional information about the reason of sending this message. Some critical errors can also lead to a truncation of the session.

3. ANALYSIS

In this section two events are analysed. One being an event triggered by political issues, the other one was caused by a natural disaster. Both events left traces detectable by having a closer look at publicly available BGP archives. The analysis of the political motivated censorship observed in January 2011 in Egypt is based on existing papers and articles. Contrastingly, the analysis of the earthquake near Christchurch in New Zealand in February 2011 is done by the author in coordination with his supervisor.

3.1 Uprising in Egypt in 2011

During the so called "Arab Spring" there have been several uprisings in African and Arabic countries including Egypt as one of the first countries where protests started [13]. These Egyptian protests were organized using social networks and messaging services on the Internet [14]. Over time the uprising in the capital Cairo became regular and increased in numbers and even similar events started to happen in other major cities of Egypt. Finally those events resulted in the resignation of the Egyptian President Hosni Mubarak on the 11th of February and in massive changes in the political system of Egypt [15].

Nevertheless the former government decided to take actions against the protests in form of blocking the communication infrastructure of the demonstrating people. The government censored access to social media portals and messaging services for Egyptian Internet users on January the 25th 2011 [3]. Although the government officially denied the existence of an order to block services like Twitter and Facebook [16], there were users that verified the blocking of services [17] and even Facebook announced a drop in user activities of its Egyptian users during this time [18].

3.1.1 BGP as a method of censorship

Because the blockade of websites did not stop the protests from happening and even increased them throughout Egypt the government ordered a complete blockade of all Internet traffic for Egyptian people. Due to this decision a whole country including 20 million Internet users [19] temporarily vanished from the Internet for about four days.

To understand how this massive blockade of nearly all Internet-based communication happened, it is necessary to have a look at the Egyptian Internet infrastructure which is dominated by a few big players with the Ramses Exchange being the major hub for their international Internet communication. The Ramses Internet Exchange is one of the biggest Internet exchange points in Northern Africa and the Middle East, connecting Egypt with other countries through submarine cables in the Suez canal [20]. Sources claim that all those big ISPs and of course the Ramses Exchange are controlled by the state [21].

Figure 5: Announced Egyptian IPv4 prefixes as seen from AS20928 on the 27th of January 2011

Keeping this information in mind, it is easier to understand how the simultaneous withdrawal of nearly all Egyptian IPv4 prefixes at around 22:34 UTC on the 27th of January 2011 could happen [22]. Other outage measure methods like the Internet Telescope also observed a significant drop of traffic from Egypt hosts during this time [3]. Figure 5 was created by James Cowie [22] and shows the approximate time of the withdrawals made by Egyptian ISPs through BGP. In the first hours of the 28th of January there were only a few IPv4 prefixes left announced. Nevertheless it was reported that the Egyptian Stock Exchange was still accessible via AS20928 (Noor Group) after nearly all Egyptian IPv4 prefixes were withdrawn [22].

doi: 10.2312/NET-2013-02-1_02

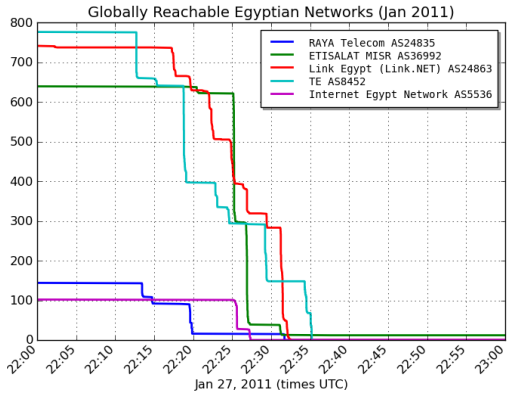


Figure 6: Detailed time-line of the withdrawals of all major Egyptian ISPs

Figure 6 shows a timeline including the exact minutes of the withdrawals made by the major Egyptian ISPs, which was published by James Cowie of Renesys [22]. According to Cowie the national ISP, Telecom Egypt (AS8452) started to withdraw its previously announced IPv4 prefixes at 22:12 UTC with Raya Telecom acted similar at 22:13 UTC. Exactly four minutes later, at 22:17 UTC, Link Egypt (AS24863) began to withdraw their prefixes, too. They were followed by Etisalat Misr (AS32992) at 22:19 UTC and Internet Egypt (AS5536) at 22:25 UTC. Converted to Egyptian local time the withdrawals started at midnight. These observed facts lead to the assumption, that all those ISPs received some kind of order from state officials to take down their services. Considering the few minutes between their actions, the order was maybe transmitted in form of a phone call. Few days later Vodafone Egypt confirmed in a press release that their services were shut down on demand of the Egyptian authorities [24].

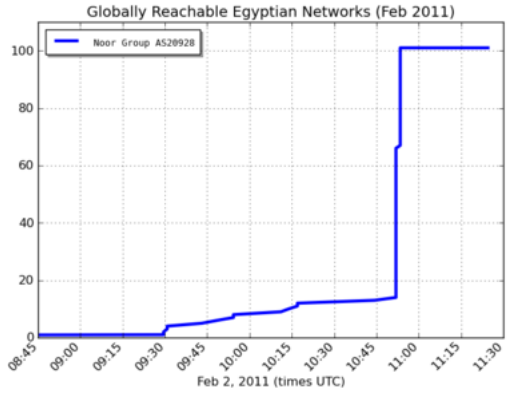


Figure 7: Announced Egyptian IPv4 prefixes as seen from AS20928 on the 2nd of February 2011

The total denial of virtually every Internet communication within Egypt lasted until the 2nd of February. At around 9:30 UTC the first IPv4 prefixes were announced again and a few hours later, at around 11:30 UTC, all Egyptian ISPs returned Internet access to all their customers. Figure 7 was also created by Renesys [23] and visualizes the return of the Egyptian IPv4 prefixes and therefore the end of the massive Internet outage in Egypt.

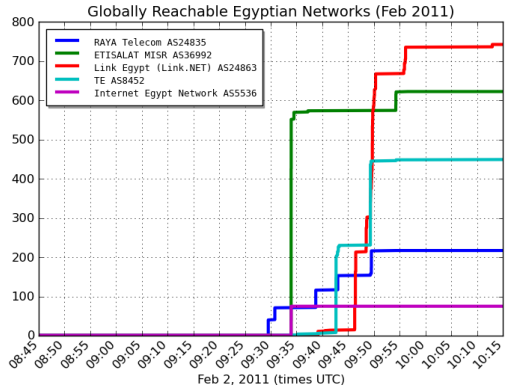


Figure 8: Detailed time-line of announcements made by major Egyptian ISPs

Figure 8 was also created by James Cowie of Renesys [23] and shows the detailed time-line of the readvertisements of routes to their IPv4 prefixes made by the Egyptian ISPs at February 2nd. The fact that all ISPs announced their routes at approximately the same time also indicates a governmental order to end the censorship actions.

3.2 Christchurch earthquake in 2011

In 2011 one of the strongest earthquakes seen in New Zealand hit the city of Christchurch killing 185 people [25]. A magnitude of 6.1 was measured during the quake which was located 10 kilometres south-west of the city of Christchurch [26]. The second largest city of New Zealand was struck by a first quake at 12:51 local time on the 22nd of February 2011 (February 21st 23:51 UTC). A first aftershock was experienced 13 minutes later at 00:04 UTC with a magnitude of 5.8. Thereafter two more aftershocks were observed at 01:50 UTC and 01:51 UTC, both with a magnitude greater than 5. The last shock had a magnitude of 5.0 and was detected at 03:01 UTC, which was 193 minutes after the major earthquake [27].

Those quakes did not only affect buildings and people, but also technical infrastructure like power lines and the water systems were damaged and unusable in some of the suburban parts of the city [29]. Power outages were affirmed in over 80 percent of the city affecting approximately 160,000 people. Within five days the power cuts were repaired and 82 percent of the affected households had power again. Nevertheless some central parts of the city were without power until May 1st [28]. Even with the local electrical companies solving the problem in a rather short timespan, there could still be visible damage in global Internet communication during the earthquake. In the next paragraph of this paper the effect of the earthquake on the announced prefixes of Christchurchs Internet is described in detail.

3.2.1 Geo-location of IPv4 ranges

To determine the effect of the quake on the Internet of Christchurch the first thing that needs to be done is to define the structure and size of Christchurchs Internet. Therefore it is assumed that all IP addresses related to the city of Christchurch by a geo-location database can be seen as the

doi: 10.2312/NET-2013-02-1_02

"Internet of Christchurch". This approach leaves out that these networks are maybe not directly connected with each other within the geographical area of Christchurch, but similar approaches are used in papers by various researchers [1, 3, 2]. To get the data needed to assign IP addresses to geographical locations the GeoLite City Database of MaxMind was used [30]. The comma separated files (CSV) were imported to a SQL-powered database. By querying this database also more complex requests were possible to determine the location of certain IPv4 addresses or even whole IPv4 ranges. Doing this resulted in exactly 320 IPv4 ranges that MaxMind believes to be located in New Zealand's second largest city.

3.2.2 *Allocation of IPv4 ranges to AS*

As those IP ranges are not always announced as a separate range through BGP, but as part of a bigger range, it was necessary to find out which announced IP prefixes those ranges belong to. This was done by calculating the smallest IP address possible laying within an IP range and probing which route a packet destinated to this address would be going. By using routing tables the AS responsible for certain IP ranges can be easily determined. In this paper the publicly available BGP dumps of RIPE NCC were used [31]. By doing this a list with 146 IP ranges has been created. This number means that the 320 ranges found in the geo-location database can be summarised to 146 bigger ranges actually propagated with BGP. Those ranges were announced by 25 different AS with AS4771 (Telecom New Zealand) being the biggest amongst them with 117 announced prefixes. A total of 4 prefixes of the geo-location database were not in use. Figure 9 visualizes the results of this part of the analysis.

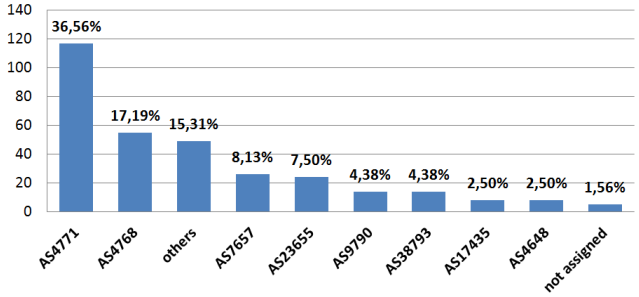


Figure 9: Amount of announced IPv4 prefixes located in Christchurch segmented on base of ASNs

3.2.3 *Analysis of BGP data dumps*

To analyse what actually happened during the earthquake and in the hours later, BGP dumps were necessary. RIPE NCC publishes a complete and uncensored dump of the BGP data received by their routers all around the world [31]. Quite a lot of Christchurch's AS are either directly or indirectly peered with Netgate (AS4648) [32] as is the Telecom New Zealand, as the biggest ISP in Christchurch. Netgate itself is peered with certain members of London Internet exchange point (LINX), like Easynet Global Services (AS4589) and Hurricane Electric (AS6939) [32, 33]. This is why the raw BGP data dumps of a router based within LINX in London was used as source for further analysis [34].

After converting the raw BGP data dumps with `libbgpdump` [35] developed by RIPE, it was possible to perform a detailed search about any announcements or withdrawals made, involving one of the prefixes of Christchurch. A timespan of four hours, from 23:00 UTC on 21st until 04:00 UTC on 22nd of February 2011, was regarded in this analysis.

The results of analysing the BGP dumps showed that the impact of the earthquake on the Internet was quite small. Of all 146 IPv4 prefixes checked there were only three subject to route-changes or withdrawals in the timespan analysed. Figure 10 displays the exact timestamps of the earthquakes according to the Earthquake Commission of New Zealand [36] and the withdrawals as seen at LINX in London.

Due to this it can be assumed that the undersea cables connecting New Zealand with the rest of the world were not badly damaged. The events triggered by such a damage to important core infrastructure could be observed during the massive earthquake in Japan on March 11th of 2011. During the hours after the Japanese quake there were several announcements changing routes to Japanese prefixes observed [1]. Nevertheless, during the first hours after the quake at Christchurch there were no such changes in routing detectable, at least not for IPv4 prefixes based in the city of Christchurch.

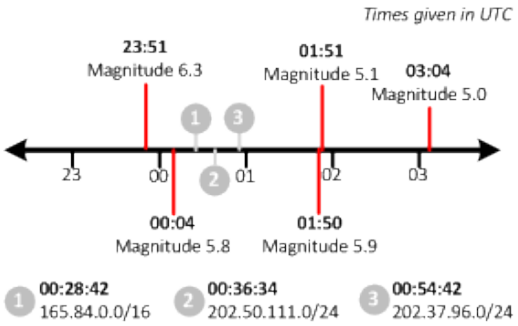


Figure 10: Timeline visualizing the quakes and BGP withdrawals during the earthquake in Christchurch

Christchurch Polytechnic Institute. The first organisation experiencing a total withdrawal of all routes during the earthquake was Christchurch Polytechnic Institute. The Institute is registered with an own ASN, AS45138 [37]. Their IPv4 prefix `165.84.0.0/16` was originally announced by this AS. So no more actions were needed to confirm the property of this IPv4 range. Furthermore, it was the first one affected by withdrawals at 00:28:24 UTC. It seems that there was a serious interruption to New Zealand as seen from Verizon Business (AS701) and Telstra Global (AS4637), which had effects on paths to the mentioned prefix. On 00:30:12 UTC the last path was withdrawn and the Polytechnic Institute of Christchurch vanished from routing tables based at LINX in London. Considering the BGP timeouts which can last up to three minutes, the first withdrawal affecting the prefix was about 20 minutes after the major quake hit Christchurch. Nevertheless major damages on the Polytechnic Institute were reported [38].

doi: 10.2312/NET-2013-02-1_02

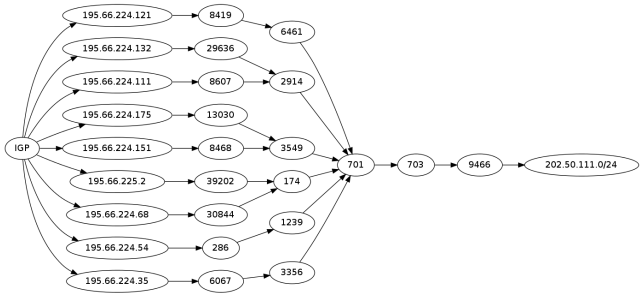


Figure 11: Announced BGP routes to 202.50.111.0/24 as seen from LINX at 16:00 UTC on February the 21st 2011

Figure 12: Last withdrawn route to 202.50.111.0/24 as seen from LINX at 00:38 UTC

MYOB Technology Limited. The second IPv4 prefix observed to go down was `202.50.111.0/24` which belongs to MYOB Technology Ltd, an Australian accounting, payroll and web-hosting provider. The data collected from a whois query at `whois.apnic.net` shows that the affected range belongs to their branch in Christchurch. The range was announced by Snap Internet Limited (AS23655) which also announced a total 7 percent of Christchurchs prefixes like figure 9 shows. The first withdrawal was propagated by a router of Catalyst2 Services (AS29636) at 00:36:34 UTC which was approximately half an hour after the second quake. The last route to the prefix was withdrawn on 00:38:21 UTC. The figures 11 and 12 display the routes and their withdrawal.

Tait Communications. Being the third and last IPv4 prefix withdrawn in the analysed timespan, `202.37.96.0/24` belongs to Tait Communications and their branch in Christchurch according to the whois record. Similar to the passage above the prefix was also announced by Snap Internet Limited (AS23655). The first withdrawal was received by KPN Internet Backbone (AS286) at 00:54:42 UTC and therefore about 50 minutes after a quake hit Christchurch. Until 00:57:21 UTC all routes to the prefix were withdrawn.

3.2.4 Reannouncements and Downtime

To get more information about the amount of time the mentioned IPv4 prefixes stayed unreachable, more data from the RIPE router at LINX was analysed. Until 19:00 UTC on the 22nd February 2011 there were no new announcements involving those prefixes. Due to this it is clear that these downtimes were not just a plain interruption in connectivity but a serious outage.

4. CONCLUSION

Although analysing BGP data collected by routers all over the world can be used as a measuring method for Internet outages, it is also limited to a rough view of events and their impact on global Internet communications. By using geo-location information events can get tracked down to geographical areas. So even if there were only three IPv4 prefixes down in Christchurch because of the earthquake, the overall traffic of the hosts dropped significantly according to current papers [39]. From this it follows that even if the central infrastructure in Christchurch was still available and worked during the first hours of the quake, a whole bunch of common hosts were offline because of the power outages and their lack of own power supplies.

Only looking at common BGP data dumps cannot determine the whole extent of natural events as big ISPs usually make expensive efforts to save their core infrastructure from failing. Small and medium companies, private households and governmental offices usually do not have such fail-safe strategies for their IT infrastructures and computers. Due to this a core router in the ISPs network may be still alive and communicating through BGP with its neighbours, but there is hardly any traffic to be routed. This very issue can not be measured by BGP. Therefore the analysis of BGP data can only detect global scale outages of whole IP prefixes and because of this BGP analysis can be seen as a macroscopic view of the Internet as a collection of different networks exchanging information with each other.

On the contrary to natural disasters where the core infrastructure can still be alive without its previous hosts and its smaller adjacent networks, total outages due to governmental orders can be detected by using BGP data. Not only Egypt experienced these facts, but also Libya and currently Syria [3]. Tunnelling of different communication streams and using all kinds of creative workarounds [40] make it hard for political organisations to control the information flow and content of communication within the Internet. So the only way to successfully deny information leaking into the Internet or to deny that information from the Internet is visible to people, is to force a total shutdown of the locale core infrastructure needed to communicate with the rest of the world. This is what happened in those three countries mentioned above.

But these most massive methods of censorship can usually be measured in BGP, as withdrawals are the easiest and cheapest way to make sure that whole networks are unable to communicate with the rest of the Internet. If there is no routing to a network no bidirectional communication can be established and therefore even physical connected networks are not usable for Internet-based communication any more. Furthermore by using withdrawals, it is also possible to leave certain networks, which are necessary or important, fully working. In analysis of BGP data those efforts can be seen and measured as recent papers proved [3].

4.1 Future Work

As described in this paper BGP alone can mostly be used as a part of a much more widespread analysis method to get a more detailed view on the impact of events. The first papers working with analysis of different technologies and techniques are published [3, 39]. Methods like measurements of the Internet Background Radiation (IBR) and advanced traceroute techniques can be used to have a really close look at networks and their behaviour in the interconnected Inter-

doi: 10.2312/NET-2013-02-1_02

net. Although these methods do also have their downsides as they can be disturbed by censorship issues on the data layer [39] so BGP will always be one part of a much more sophisticated measure facility using different sources to gather data and determine a detailed in-depth look at events involving Internet outages.

5. REFERENCES

[1] Liu, Yujing and Luo, Xiapu and Chang, Rocky K. C. and Su, Jinshu: *Characterizing inter-domain rerouting after japan earthquake*, In Proceedings of the 11th international IFIP TC 6 conference on Networking - Volume Part II, pages 124-135, Prague, Czech Republic, 2012

[2] Schulman, Aaron and Spring, Neil: *Pingin' in the rain*, In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, pages 19-28, Berlin, Germany, 2011

[3] Dainotti, Alberto and Squarcella, Claudio and Aben, Emile and Claffy, Kimberly C. and Chiesa, Marco and Russo, Michele and Pescapé, Antonio: *Analysis of Country-wide Internet Outages Caused by Censorship*, In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, pages 1-18, Berlin, Germany, 2011

[4] *YouTube Hijacking: A RIPE NCC RIS case study*, `http://www.ripe.net/internet-coordination/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study`, March 2008

[5] Cowie, James: *Syrian Internet Shutdown*, `http://www.renesys.com/blog/2011/06/syrian-internet-shutdown.shtml`, June 2012

[6] Hawkinson, John and Bates, Tony: *Guidelines for creation, selection, and registration of an Autonomous System*, Internet Request for Comments RFC 1930, Internet Engineering Task Force, March 1996

[7] Vohra, Quaizar and Chen, Enke: *BGP Support for Four-octet AS Number Space*, Internet Request for Comments RFC 4893, Internet Engineering Task Force, May 2007

[8] Rekhter, Yakov and Hares, Sue and Li, Tony: *A Border Gateway Protocol 4*, Internet Request for Comments RFC 4271, Internet Engineering Task Force, January 2006

[9] Levin, Hagay and Schapira, Michael and Zohar, Aviv: *Interdomain routing and games*, In Proceedings of the 40th annual ACM symposium on Theory of computing, pages 57-66, New York, USA, 2008

[10] Baker, Fred: *Requirements for IP Version 4 Routers*, Internet Request for Comments RFC 1812, Internet Engineering Task Force, June 1995

[11] Chen, Enke: *Route Refresh Capability for BGP-4*, Internet Request for Comments RFC 2918, Internet Engineering Task Force, September 2000

[12] Fuller, Vince and Li, Tony: *Classless Inter-domain Routing: The Internet Address Assignment and Aggregation Plan*, Internet Request for Comments RFC 4632, Internet Engineering Task Force, August 2006

[13] British Broadcasting Corporation: *Arab uprising: Country by country*, `http://www.bbc.co.uk/news/world-12482291`, September 2011

[14] British Broadcasting Corporation: *Egypt protests escalate in Cairo, Suez and other cities*, `http://www.bbc.co.uk/news/world-africa-12272836`, January 2011

[15] British Broadcasting Corporation: *Egypt's revolution: Interactive map*, `http://www.bbc.co.uk/news/world-middle-east-12327995`, September 2012

[16] Reuters: *Egypt government denies disrupting websites -cabinet*, `http://www.reuters.com/article/2011/01/26/egypt-web-idUSLDE70P28720110126`, January 26th, 2011

[17] Garret, S.: *"We can confirm that Twitter was blocked in Egypt around 8am PT today."*, `http://twitter.com/#!/twitterglobalpr/status/30063209247408128`, January 25th, 2011

[18] Reuters: *Facebook says has seen drop in traffic from Egypt*, `http://www.reuters.com/article/2011/01/27/facebook-egypt-idUSN2727880720110127`, Janurary 27th, 2011

[19] Ministry of Communication and Internet Technology, Arab Republic of Egypt: *ICT Indicators Report 2007-2011*, `http://www.mcit.gov.eg/Upcont/Documents/Publications_1382012000_Indicator%20E%202012-final2.pdf`, July 2012

[20] Packet Clearing House Research: *Cairo Internet Exchange* `https://prefix.pch.net/applications/ixpdir/detail.php?exchange_point_id=59`, September 2012

[21] Mahlknecht, Greg: *Greg's Cable Map*, `http://www.cablemap.info`, visited on August 17th, 2012

[22] Cowie, James: *Egypt Leaves the Internet*, `http://www.renesys.com/blog/2011/01/egypt-leaves-the-internet.shtml`, January 27th, 2011

[23] Cowie, James: *Egypt Returns to the Internet*, `http://www.renesys.com/blog/2011/02/egypt-returns-to-the-internet.shtml`, February 2nd, 2011

[24] Toonk, Andree: *Egypt falls off the Internet*, `https://bgpmon.net/blog/?p=450`, January 2011

[25] New Zealand Police: *Christchurch Earthquake - List of deceased*, `https://www.police.govt.nz/list-deceased`, February 2012

[26] Reuters: *New Zealand city of Christchurch hit by strong earthquake*, `http://www.reuters.com/article/2011/02/22/newzealand-quake-idUSWLF00502320110222`, February 2011

[27] National Earthquake Information Center of the United States: *Magnitude 6.1 - South Island of New Zealand*, `http://earthquake.usgs.gov/earthquakes/eqinthenews/2011/usb0001igm/`, February 2011

[28] Orion New Zealand Limited: *Orion earthquake response*, `www.oriongroup.co.nz/downloads/Position_statement_210311_1pm.pdf`, March 2011

[29] Stuff.co.nz: *Power restored to most households*, `https://www.stuff.co.nz/national/`

christchurch-earthquake/4734825/Power-restored-to-most-households, February 2011

[30] MaxMind: *GeoLite Databases*, https://www.maxmind.com/app/geolite, September 2012

[31] RIPE Network Coordination Centre: *RIS Raw Data*, https://www.ripe.net/data-tools/stats/ris/ris-raw-data, September 2012

[32] Hurricane Electric Internet Resources: *AS4648 - Netgate IPv4 Peers*, http://bgp.he.net/AS4648#_peers, September 2012

[33] RIPE Network Coordination Centre: *RRC01 - LINX, London Peer List*, http://www.ris.ripe.net/peerlist/rrc01.shtml, September 2012

[34] RIPE Network Coordination Centre: *rrc01.ripe.net at LINX, London*, http://data.ris.ripe.net/rrc01/, September 2012

[35] RIPE Network Coordination Centre: *libBGPdump repository*, http://bitbucket.org/ripencc/bgpdump/wiki/Home, September 2012

[36] Earthquake Commission of New Zealand: *New Zealand Earthquake Report*, http://www.geonet.org.nz/earthquake/quakes/3468575g.html, February 2011

[37] Hurricane Electric Internet Resources: *AS45138 - Christchurch Polytechnic Institute*, http://bgp.he.net/AS45138, September 2012

[38] Tiaki, Kai: *Earthquake 2011: the February 22 earthquake had a devastating impact on Christchurch Polytechnic Institute of Technology's central city campus, forcing more than 600 nursing students and staff to relocate to Lincoln University*, In Nursing New Zealand - Volume 17, page 10, Christchurch, New Zealand, 2011

[39] Dainotti, Alberto and Amman, Roman and Aben, Emile and Claffy, Kimberly: *Extracting Benefit from Harm: Using Malware Pollution to Analyze the Impact of Political and Geophysical Events on the Internet*, In ACM SIGCOMM Computer Communication Review - Volume 42, pages 31-39, New York, USA, 2012

[40] Andersson, Bjorn: *iodine - IP over DNS tunnel*, http://code.kryo.se/iodine/, February 2010

doi: 10.2312/NET-2013-02-1_02

Ubertooth - Bluetooth Monitoring und Injection

Martin Herrmann
Betreuer: Stephan Günther
Seminar - Future Internet WS2012/13
Lehrstuhl Netzarchitekturen und Netzdienste (I8)
Fakultät für Informatik, Technische Universität München
Email: martin.herrmann@fs.tum.de

KURZFASSUNG

Es gibt nur wenige Geräte, die Bluetooth *Monitoring* oder *Injection* ermöglichen, da sich dies auf Grund spezieller Eigenschaften des Bluetoothprotokolls als schwierig erweist. Kommerzielle Hardware, welche hierzu in der Lage ist, ist oftmals zu teuer für die Allgemeinheit, weshalb Alternativen wie das *Ubertooth Project* entwickelt werden.
Hierdurch wird es möglich, Angriffe auf Bluetooth Verbindungen durchzuführen, was gravierende Sicherheitslücken aufdeckt. Ferner können Probleme besser adressiert werden, da große Freiheiten in der Entwicklung neuer Anwendungen auf der Basis von Bluetooth eröffnet werden.

Schlüsselworte

Bluetooth, FHSS, Monitoring, Injection, Ubertooth, Authentifizierung

1. EINLEITUNG

Obwohl Bluetoothprodukte bereits seit etwa zehn Jahren erhältlich und sehr weit verbreitet sind, gab es bis vor kurzem keine günstigen, der Allgemeinheit zugänglichen Lösungen, um den Datenfluss aller Bluetoothverbindungen der Umgebung zu überwachen. Ebenso war es nicht möglich, eigene Pakete in bestehende Verbindungen zu injizieren. Diese *Monitoring* und *Injection* genannten Techniken sind für verschiedene Zwecke (z. B. zur Analyse von Verbindungen) von Interesse und in anderen Bereichen, insbesondere bei *IEEE 802.11* (Wireless LAN), längst verfügbar.
Auf Grund dieser Tatsache wurde das *Ubertooth Project* ins Leben gerufen, welches zum Ziel hat, eine offene Plattform für Bluetooth Monitoring und Injection zu schaffen. Im Zuge der Entwicklung entstand der *Ubertooth Zero* und sein Nachfolger, der *Ubertooth One*. Dabei handelt es sich um spezielle Hardware, die in der Lage ist, Daten auf dem 2.4 GHz Band, auf dem auch Bluetooth arbeitet, zu empfangen und zu senden.
Diese Arbeit befasst sich im folgenden zweiten Kapitel mit der Bluetooth Technologie sowie Monitoring und Injection an sich, bevor sie sich im dritten Kapitel mit verschiedenen Monitoring und Injection Lösungen für Bluetooth beschäftigt und diese vergleicht. Im vierten Kapitel werden Anwendungen von Bluetooth Monitorung und Injection vorgestellt, welche auch Angriffe auf fremde Datenübertragungen beinhalten.

2. TECHNISCHE GRUNDLAGEN

2.1 Bluetooth

Bluetooth wurde im Rahmen der IEEE 802.15 Arbeitsgruppe entwickelt, welche sich mit *Wireless Personal Area Networks* (WPAN) beschäftigt [1]. Der Zweck eines WPAN ist es, es verschiedenen Geräten (z. B. Handys oder Laptops) zu ermöglichen, drahtlos miteinander zu kommunizieren.
Der Bluetooth Protokollstapel lässt sich in verschiedene Schichten aufteilen (dargestellt in *Abbildung 1*), von denen für diese Arbeit insbesondere die untersten von Interesse sind, da sie für die Problemstellung des Monitorings wesentlich sind.

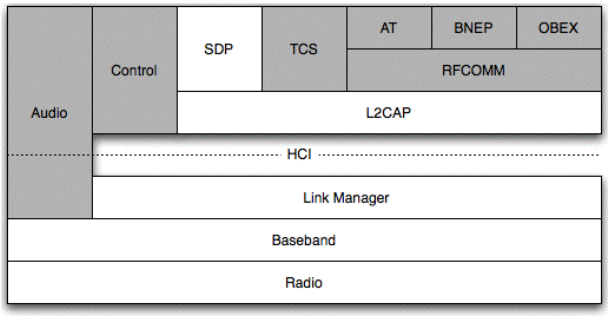


Abbildung 1: Bluetooth Protokollstapel [2]

2.1.1 Radio Layer

Die unterste Schicht ist der so genannte *Radio Layer*. Bluetooth sendet auf dem 2,4 GHz Band, welches zu den ISM Bändern gehört. Da ISM Bänder mit nur wenigen Einschränkungen von jedem genutzt werden können, kann nicht garantiert werden, dass das Signal störungsfrei übertragen wird. Deswegen kommt hier ein *Frequency Hopping Spread Spectrum* (FHSS) genanntes Verfahren zum Einsatz (siehe *Abbildung 2*), welches den Frequenzbereich zwischen 2,4 GHz und 2,4835 GHz in 1 MHz breite Kanäle unterteilt [1]. Außerdem existiert ein 3 MHz breiter Schutzabstand am oberen Rand des Spektrums, sowie ein 1,5 MHz breiter Abstand am unteren Rand, weshalb sich 79 nutzbare Kanäle ergeben, zwischen denen alle 625 µs gewechselt wird. Wenn auf einer bestimmten Frequenz also Störungen vorliegen (z. B. durch parallele Nutzung von anderen Geräten), kommt es nur kurzzeitig zu Übertragungsproblemen, da der Kanal sofort wieder gewechselt wird. Ferner hat dieses Verfahren den Vorteil, dass das gezielte Abhören einer Verbindung erschwert wird, da ein Angreifer die verwendete Sprungfolge kennen muss.

doi: 10.2312/NET-2013-02-1_03

Als Modulationsverfahren wird *Gaussian Frequency Shift Keying* (GFSK) verwendet, welches zusätzlich zum eigentlichen modulieren der Frequenz auch noch einen Gauss-Filter auf das Signal anwendet.

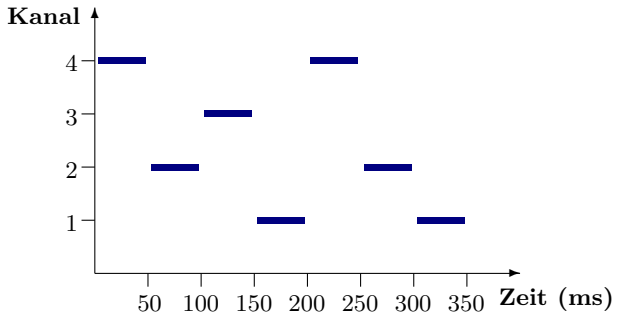


Abbildung 2: Einfaches Beispiel für ein FHSS Verfahren (4 Kanäle, Sprung alle 50 ms)

2.1.2 Baseband Layer

Der *Baseband Layer* verwaltet die physikalischen Verbindungen. Jedes Gerät kann hier über eine 48 Bit lange *Bluetooth Device Address* (BD_ADDR) eindeutig identifiziert werden. Diese kann in *Lower Address Part* (LAP, 24 Bit), *Upper Address Part* (UAP, 8 Bit) und *Nonsignificant Address Part* (NAP, 16 Bit) unterteilt werden (siehe *Abbildung 3*), wobei UAP und NAP zusammen den Hersteller identifizieren, während der LAP von diesem vergeben wird. Für bestimmte Zwecke existieren die reservierten LAPs *0x9E8B00–0x9E8B3F*. Für diese wird 0x00 als UAP verwendet.

16 Bit	8 Bit	24 Bit
NAP	**UAP**	**LAP**

Abbildung 3: Bluetooth Device Address Format [1]

Eine physikalische Verbindungen (als *Piconet* bezeichnet) besteht aus einem Master und mindestens einem Slave. Adressiert wird sie durch die Verwendung eines *Access Code*, welcher aus dem LAP des Masters generiert wird. Damit die Datenübertragung möglich ist, müssen natürlich alle Teilnehmer die gleiche Sprungfolge für die Frequenzen verwenden. Diese wird aus LAP und UAP des Masters errechnet, und ist in etwa gleichverteilt über die 79 Kanäle. Da sich der Master und die Slaves auch an der gleichen Stelle in der Sprungfolge befinden müssen, werden die Slaves zusätzlich auch noch zeitlich mit dem Master synchronisiert [1].

2.1.3 Paketaufbau

Alle Pakete werden nach dem selben Schema aufgebaut (siehe *Abbildung 4*). Sie beginnen immer mit dem *Access Code*, gefolgt von dem *Header* und der *Payload*. Es gibt Pakete, die nur den *Access Code* enthalten, welche die nur den *Access Code* und den *Header* enthalten, und solche, die alle drei Teile beinhalten [1]. Der *Access Code* besteht aus einer 4 Bit Präambel, gefolgt von einem 64 Bit langen *Sync Word*, welches aus LAP und UAP berechnet wird. Falls ein Header folgt, endet der *Access Code* noch mit einem 4 Bit Trailer. Der Header enthält Felder die zur Steuerung des Datenflusses benutzt werden. Dazu gehört zum Beispiel ein Feld, welches den Pakettyp spezifiziert, sowie Felder zur Übertragungskontrolle und zur Adressierung einzelner Slaves.

68/72 Bit	54 Bit	0 – 2745 Bit
Access Code	**Header**	**Payload**

Abbildung 4: Paketaufbau [1]

Eines der einfachsten Pakete ist das ID Paket. Es besteht nur aus dem *Access Code* und hat daher eine Länge von 68 Bit. Für die Berechnung des Sync Word wird die für *Inquiries* reservierte LAP *0x9E8B33* benutzt.
Ein sehr wichtiges Paket für den Verbindungsaufbau ist das *Frequency Hopping Synchronization* (FHS) Paket. Es enthält zusätzlich zum *Access Code*, welcher dieses mal aus dem LAP und UAP eines anderen Gerätes berechnet wurde und somit an dieses adressiert ist, einen Header sowie Payload. Die Payload enthält unter anderem alle drei Teile der BD_ADDR, sowie die interne CLK des Gerätes. Dieses Paket enthält alle Informationen, die zur Berechnung der Sprungfolge benötigt werden.
Ferner existieren noch eine Reihe anderer Pakete, insbesondere solche zur Datenübertragung. Dabei gibt es verschiedene Typen für spezielle Zwecke, von denen einige auch größer als ein Übertragungsslot sind (z. B. 3 oder 5 Slots).

2.1.4 Verbindungsaufbau

Für den Verbindungsaufbau müssen die Geräte zunächst gepaart werden. Dazu schickt der zukünftige Master *Inquiries* aus (ID Pakete). In diesem Fall werden nur 32 der 79 möglichen Kanäle für die Sprungfolge verwendet. Geräte, die auf *Sichtbar* gestellt sind, antworten mit einer *Inquiry Response*, welche bereits die Adresse und die CLK des Gerätes enthält. Um das *Pairing* fortzuführen, berechnet der Master aus der erhaltenen Adresse eine spezifische Sprungfolge und synchronisiert sie mit dem Takt des Slaves, der sich nun im *Page Scan* Modus befindet. Diese Sprungfolge benutzt der Master nun, um dem Slave die nötigen Informationen mitzuteilen, die er für das Errechnen der entgültigen Sprungfolge braucht. Er sendet dazu zwei ID Pakete und wartet im nächsten Sendeslot auf ein ID Paket des Slaves als Antwort. Nun sendet der Master ein FHS Paket, welches der Slave mit einem weiteren ID Paket beantwortet. Da der Slave nun die korrekte Sprungfolge kennt, wechselt der Master nun zu dieser. Grundsätzlich sendet der Master hierbei immer in den geraden Slots der Sprungfolge, während der Slave in den ungeraden sendet.

2.1.5 Sicherheit

Bluetooth verfügt auch über Sicherheitsverfahren, welche in Kapitel 13 der Bluetooth Spezifikationen detailliert beschrieben werden [1]. Wenn ein physikalischer Kanal (synchronisierte Frequenz Sprungfolge) exisitiert, werden über ihn logische Verbindungen aufgebaut. Dabei müssen in der Regel

doi: 10.2312/NET-2013-02-1_03

die PINs auf den Geräten übereinstimmen oder es muss der korrekte PIN für das entfernte Gerät eingegeben werden, falls dieser fest ist (z. B. bei Headsets ohne Interface).
Um die Authentizität einer Verbindung sicherzustellen, wird zuerst der K_{init} Schlüssel generiert. Dazu wird der E_{22} Algorithmus (welcher hier nicht genauer erklärt wird), mit dem verwendeten PIN, einer 128 Bit langen Zufallszahl *IN_RAND* (von Master generiert, an Slave übertragen) und der *BD_ADDR* des Slaves (bei festem PIN des Slaves die des Masters) als Parameter, ausgeführt.
Der K_{init} Schlüssel wird zur Verschlüsselung der Übertragung von zwei weiteren 128 Bit Zufallszahlen (LK_RAND_A und LK_RAND_B) verwendet, welche zusammen mit den BT_ADDR der beiden Geräte zur Berechnung des *Link Key*, welcher nun zur Authentifizierung benutzt werden kann.
Hierfür schicken sich die beiden Geräte wieder gegenseitig 128 Bit lange Zufallszahlen AU_RAND_A und AU_RAND_B, welche dann mit dem zuvor gewonnenem Schlüssel verschlüsselt werden. Die Ergebnisse ($SRES_A$ und $SRES_B$) der Operationen auf beiden Geräten werden verglichen und die Verbindung wird nur zugelassen, wenn sie übereinstimmen [3].
Der Schlüssel wird solange gespeichert, bis das *Pairing* aufgehoben wird. Ferner kann aus diesem Schlüssel ein weiterer generiert werden, der zur Verschlüsselung der Datenübertragung genutzt wird. Dieser hat eine Länge von 8 – 128 Bit.

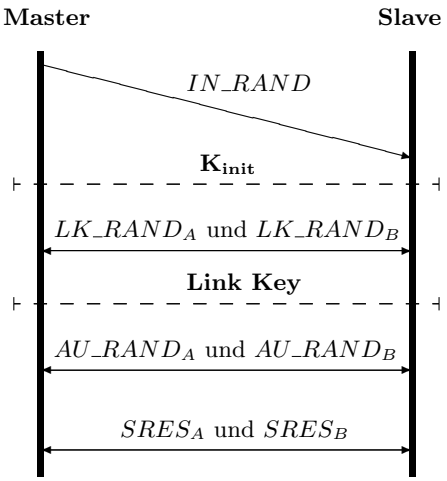


Abbildung 5: Ablauf des Authentifizierungsprozesses

2.2 Monitoring und Injection

Da ein Kanal nur selten exklusiv benutzt wird, kommen immer auch Daten an, welche für andere Empfänger bestimmt waren. Damit diese nicht die eigene Kommunikation stören, wird im normalen Betriebsmodus beim Erhalt eines neuen Pakets zunächst überprüft, ob es für dieses Gerät bestimmt ist. Im Falle von (Wireless) LAN werden hierzu die *MAC-* und die *IP-Adressen* überprüft, während bei Bluetooth der *Access Code* ausschlaggebend ist. Pakete, die nicht für ein bestimmtes Gerät bestimmt sind, werden von diesem ignoriert und nicht an höhere Schichten weitergeleitet.
Manchmal ist es allerdings von Interesse, auch solche Pakete zu erhalten, die nicht für einen selbst bestimmt sind (z. B. zur Überwachung von Netzwerken). In diesem Fall werden alle Pakete, die von der Netzwerkhardware empfangen werden, an höhere Schichten weitergeleitet, was man auch als *Monitoring* oder *Monitor Mode* der Hardware bezeichnet. Die empfangenen Daten werden dann in der Regel von einem Programm wie *Wireshark* [4] oder *Kismet* [5] verarbeitet und dargestellt. *Abbildung 6* zeigt Wireshark, das dank *Monitor Mode* Funktionalität des WLAN-Adapters alle in der Umgebung verschickten Pakete anzeigt.

Destination	Protocol	Length	Info
Wistron_29:43:03	802.11	1583	QoS Data, SN=565, FN=0, Flags=.p....F.C
Wistron_29:43:03	802.11	1583	QoS Data, SN=566, FN=0, Flags=.p....F.C
Wistron_29:43:03	802.11	1583	QoS Data, SN=567, FN=0, Flags=.p....F.C
Wistron_29:43:03	802.11	1583	QoS Data, SN=568, FN=0, Flags=.p....F.C
Wistron_29:43:03	802.11	1583	QoS Data, SN=569, FN=0, Flags=.p....F.C
Wistron_29:43:03	802.11	1583	QoS Data, SN=570, FN=0, Flags=.p....F.C
Wistron_29:43:03	802.11	1583	QoS Data, SN=571, FN=0, Flags=.p....F.C
Tp-LinkT_f3:72:1a (RA	802.11	58	802.11 Block Ack, Flags=........C
Tp-LinkT_f3:72:1a (RA	802.11	46	Request-to-send, Flags=........C
Wistron_29:43:03 (RA)	802.11	40	Clear-to-send, Flags=........C
Tp-LinkT_f3:72:1a	802.11	135	QoS Data, SN=32, FN=0, Flags=.p.....TC

Abbildung 6: Monitor Mode und Wireshark

Wenn Daten gesendet werden sollen, wird in der Regel eine von der Netzwerkhardware (bzw. von ihrem Treiber) bereitgestellte Schnittstelle verwendet. Üblicherweise muss sich die Anwendung nicht um den genauen Ablauf der Kommunikation kümmern, die Übertragung erfolgt also transparent. Für gewisse Anwendungen ist es allerdings vonnöten, tiefer in den Kommunikationsablauf einzugreifen. So gibt es die *Injection* Technik, bei der eigene Datenpakete in fremde Verbindungen eingeschleust werden, um so z. B. Verschlüsselungen zu knacken oder *Spoofing* Angriffe auszuführen. Dies muss natürlich auch von der Hardware bzw. vom Treiber unterstützt werden, da hier die normalen Senderoutinen umgangen werden. Eine sehr bekannte Anwendung ist hier das Knacken der WEP-Verschlüsselung mittels der *Aircrack-ng Suite* [6], welche auch die Bedeutung der Verfügbarkeit dieser Techniken aufzeigt.
Selbst wenn gewisse Verfahren (wobei hier WEP als Paradebeispiel dient) in der Theorie längst nicht mehr sicher sind, werden sie trotzdem weiterhin eingesetzt, sofern sich die praktische Umsetzung dieser Angriffe schwierig gestaltet. Der Gedanke dahinter ist oftmals, dass es die Allgemeinheit nicht betrifft, was dazu führt, dass sich einem kleineren Personenkreis weite Angriffsmöglichkeiten eröffnen. Wenn theoretische Angriffe allerdings für jedermann leicht ausführbar werden, wie dies bei WEP durch die Verbreitung von *Monitoring* und *Injection* fähiger Hardware sowie entsprechender Tools geschah, ist man gezwungen sichere Verfahren zu entwickeln und auch einzusetzen, was einen erheblicher Gewinn darstellt.

3. MONITORING UND INJECTION LÖSUNGEN FÜR BLUETOOTH

Wie bereits in der Einleitung erwähnt, gab es für Bluetooth lange Zeit kaum Geräte, welche über einen *Monitor Mode* oder die Möglichkeit der *Packet Injection* verfügten.
Hindernisse hierfür hierfür werden in Abschnitt 3.1 diskutiert. In Abschnitt 3.2 werden einige professionelle Lösungen vorgestellt, bevor sich Abschnitt 3.3 mit *Ubertooth* im Detail beschäftigt. Zum Abschluss werden in Abschnitt 3.4 die verschiedenen Lösungen miteinander verglichen.

doi: 10.2312/NET-2013-02-1_03

3.1 Gründe für den Mangel

Das erste Hindernis beim Abhören von Bluetoothverbindungen ist die Verwendung des FHSS Verfahrens (siehe *Abschnitt 2.1.1*). Um eine komplette Datenübertragung zu empfangen, muss man entweder alle 79 Kanäle gleichzeitig abhören und die richtigen Pakete herausfiltern, oder die verwendete Sprungfolge kennen. Ersteres setzt hohe Ansprüche an die verwendete Hard- und Software, für die zweite Möglichkeit muss man warten, bis man Pakete empfängt und daraus versuchen, die Sprungfolge zu rekonstruieren.
Ein weiteres Problem ist die Filterung der eingehenden Pakete auf Grund ihres *Access Code*. Da dies in gängigen Bluetoothsticks laut dem Ubertooth Entwickler Michael Ossmann [7] meist direkt auf Hardware-Ebene geschieht, gestaltet es sich mit diesen schwierig, alle empfangenen Pakete an den PC weiterzuleiten. Ferner liegt der Gedanke nahe, dass die Hersteller kein Interesse daran haben, entsprechende Funktionalitäten zu implementieren, da sie so eine gewisse *Security by obscurity* aufrecht erhalten können.

3.2 Professionelle Lösungen

Trotz der im vorherigen Abschnitt beschriebenen Probleme existiert kommerzielle Analyse-Hardware für Bluetooth, wie der *Ellisys Bluetooth Explorer 400*. Laut Herstellerinformationen [8] ist dieser ist in der Lage alle 79 Bluetooth-Kanäle gleichzeitig zu überwachen, was es auch möglich macht, alle Piconetze der Umgebung gleichzeitig zu beobachten. Ferner untersützt er alle neueren Bluetooth Standards wie *Enhanced Data Rate* (EDR) und *Low Energy* (LE) und ist in der Lage, den verwendeten PIN einer Bluetoothübertragung zu berechnen und damit die Verschlüsselung zu knacken (mehr hierzu in *Kapitel 4*), sofern ein Pairing-Prozess beobachtet wird. Die Hardware wird mittels einem USB 2.0 Anschluss an einen PC angeschlossen, entsprechende Software ist im Lieferumfang. Eine Anfrage hat ergeben, dass sich der Preis für ein komplettes System um die 20 000 US Dollar bewegt.
Eine weitere Bluetooth-Monitoring Lösung ist der *ComProbe BPA500*. Dieser unterstützt laut Hersteller [9] ebenfalls EDR und LE und kann bei Beobachtung eines Pairing Prozesses den verwendeten PIN errechnen. Ferner lässt sich die Hardware noch um weitere Funktionalitäten (wie zusätzliches WLAN Monitoring) erweitern. Eine Preisanfrage wurde noch nicht beantwortet, die Kosten sind aber wahrscheinlich mit denen des Bluetooth Explorers vergleichbar.

3.3 Ubertooth

Im Gegensatz zu der in *Abschnitt 3.2* beschriebenen Lösungen liegt das Ziel des *Ubertooth Project* nicht in der Schaffung eines fertigen, kommerziellen Endprodukts, sondern der Entwicklung einer open-source Plattform, welche von jedem verwendet werden kann, der er über grundlegende Kenntnisse im Bereich der Netzwerktechnik verfügt [10]. Die Kosten für einen *Ubertooth One* liegen bei ca. 100 US Dollar.

Für die Hardware (siehe *Abbildung 7*) wird eine 4-lagige Leiterplatte verwendet. Es existiert ein *RP-SMA* Anschluss [12] für eine Antenne. Ein *CC2591 RF Front End* von Texas Instruments verarbeitet das analoge Signal für einen *CC2400 Wireless Transceiver*, welcher im 2.4 – 2.4835 GHz Bereich empfangen und senden kann [13][14]. Das Signal wird von einem Mikroprozessor (*Arm Cortex-M3* Architektur) der *LPC175x* Serie verarbeitet. Die vorliegende Hardware arbeitet mit einem *LPC1756*, welcher über 256 kB Flash Memory für die Firmware verfügt. Er arbeitet bei einer Taktfrequenz von maximal 100 MHz und wird mit einem USB 2.0 Anschluss verbunden [11].

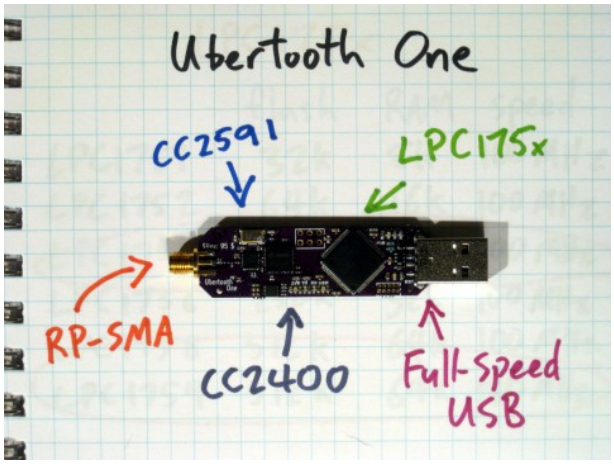


Abbildung 7: Aufbau des *Ubertooth One* [11]

Aus dieser Hardwarekonfiguration ergibt sich auch direkt eine Beschränkung: der CC2400 kann maximal mit 1 Mbit/s senden und empfangen [13]. Dies führt dazu, dass neuere Bluetooth Standards ab der Version 2.0 nicht komplett unterstützt werden können, da diese sich unter anderem durch *Enhanced Data Rate* (EDR) auszeichnen, welche bis zu 2,1 Mbit/s erreichen kann.
Um den *Ubertooth One* zu verwenden, muss man zunächst eine aktuelle Version der Firmware installieren, was am einfachsten mit der vorhandenen USB DFU Utility ist. Der auf der Projektseite verfügbare Quellcode enthält bereits einige Tools, die der Reihe nach vorgestellt werden. Diese funktionieren problemlos mit aktuellen Linux-Systemen, sofern einige zusätzliche Pakete installiert werden. Eine Portierung auf andere Systeme (z. B. MAC oder Windows) gestaltet sich problemlos.
Mit Hilfe des *Specan UI* Tools kann man sich den Signalstärkeverlauf des Frequenzbereichs in Echtzeit graphisch anzeigen. Da auch WirelessLAN in diesem Bereich sendet, kann man auch dieses so beobachten. So erkennt man auf *Abbildung 8* beispielsweise ein aktives WLAN um 2,415 MHz. Alternativ zu *Specan UI* kann man auch *ubertooth-specan* benutzen, welches die Rohdaten in einem für entsprechende Analysesoftware geeignetem Format ausgibt.

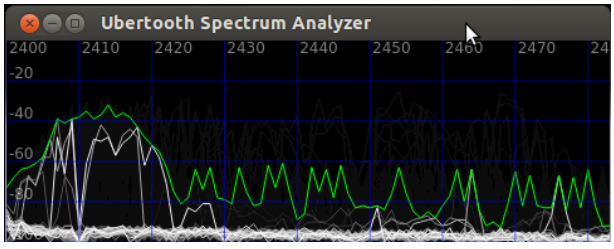


Abbildung 8: Signalstärkeverlauf mit *Specan UI*

Mit *ubertooth-util* lassen sich die Betriebsmodi des Gerätes

doi: 10.2312/NET-2013-02-1_03

konfigurieren. Hier kann man zum Beispiel den zu überwachenden Kanal auswählen, das Gerät in den *Device Firmware Upgrade* Modus versetzen oder sich diverse Informationen (wie die aktuelle Firmware Version) anzeigen lassen.
Das Tool *ubertooth-lap* ermöglicht Monitoring auf einem spezifizierten Kanal und gibt den LAP aus. Mittels *ubertooth-uap* lässt sich zu einem gegebenen LAP auch der dazugehörige UAP bestimmen. Obwohl hiermit schon eine einfache Beobachtung von Bluetoothverbindungen möglich sind, lässt sich dies noch besser mit dem Kismet-Plugin für Ubertooth realisieren. Dies bestimmt sowohl den LAP als auch den UAP der empfangen Pakete und kann diese als *pcapbtbb* Dumpfile ausgeben, die man dann in anderer Software weiter analysieren kann (siehe *Abbildung 9*).

Time	Source	Destination	Protocol	Length	Info
11.749314	00:00:00_c0:36:a6	00:00:00_00:00:00	Bluetooth	14	ID
12.258369	00:00:00_c0:36:a6	00:00:00_00:00:00	Bluetooth	14	ID
12.626241	00:00:00_14:69:bb	00:00:00_00:00:00	Bluetooth	14	ID
12.658602	00:00:00_14:69:bb	00:00:00_00:00:00	Bluetooth	14	ID
12.869856	Camex_c0:36:a6	00:00:00_00:00:00	LMP	27	LMP_setup_complete
12.908252	Camex_c0:36:a6	00:00:00_00:00:00	Bluetooth	51	DH1/2-DH1
12.911112	Camex_c0:36:a6	00:00:00_00:00:00	LMP	38	LMP_accepted
13.234873	Camex_c0:36:a6	00:00:00_00:00:00	Bluetooth	23	NULL
13.263188	Camex_c0:36:a6	00:00:00_00:00:00	Bluetooth	23	POLL
13.263595	Camex_c0:36:a6	00:00:00_00:00:00	Bluetooth	23	POLL
13.461768	Camex_c0:36:a6	00:00:00_00:00:00	Bluetooth	23	NULL
13.490820	Camex_c0:36:a6	00:00:00_00:00:00	Bluetooth	23	POLL

Abbildung 9: Bluetoothpakete in Wireshark

Leider scheint das Verschicken von Paketen über die USB-Schnittstelle bisher noch nicht möglich zu sein. Es gibt allerdings die Möglichkeit mit zwei Ubertooth Sticks einen Testmodus zu starten, bei dem sie sich gegenseitig Bluetooth ähnliche Pakete zusenden, was darauf hindeutet, dass die Firmware in absehbarer Zeit auch *Packet Injection* unterstützen wird.
Zusammenfassend lässt sich sagen, dass der *Ubertooth One* eine sehr mächtige Plattform für Bluetooth Monitoring und Injection darstellt, auch wenn viele Funktionalitäten (noch) nicht vorhanden sind. Ferner kann man die Ubertooth Plattform (entsprechende Firmware vorausgesetzt) auch für andere Systeme, welche im gleichen Frequenzbereich arbeiten, verwenden.

3.4 Vergleich

Da es sich bei den in *Abschnitt 3.4* vorgestellten Lösungen um kommerzielle Bluetooth-Monitoring Hardware handelt, kann der *Ubertooth One* natürlich nicht komplett mithalten, da diesem zum Beispiel (wie bereits in *Abschnitt 3.3* erwähnt) die EDR-Funktionalität fehlt. Außerdem befindet sich das Ubertooth-Projekt noch in der Entwicklungsphase, was bedeutet, dass viele Funktionalitäten noch nicht komplett umgesetzt sind. All dies bedeutet, dass der *Ubertooth One* im Moment noch nicht für professionelles Bluetooth-Monitoring, wie dies zum Beispiel bei der Entwicklung neuer Hard- und Software nötig ist, geeignet ist.
Dennoch ist er gerade für akademische Zwecke sehr gut geeignet, da es hier unter anderem auf geringe Kosten ankommt. Dass das Projekt quelloffen ist, ist ein weiterer Vorteil, da es somit sehr gut an spezielle Bedürfnisse angepasst werden kann, ohne dass man in langwierige Verhandlungen mit den Herstellern treten muss.

4. ANWENDUNGEN

Nachdem in *Kapitel 3* nun einige Möglichkeiten des Bluetooth Monitorings vorgestellt wurden, werden hier nun einige Anwendungen beschrieben, die über die bloße Beobachtung, beispielsweise zu Debugging-Zwecken, hinausgehen.

4.1 Knacken der Authentifizierung

In ihrer Arbeit [3] haben Yaniv Shaked und Avishai Wool ein Bruteforce Verfahren beschrieben, welches in der Lage ist den PIN zu berechnen, falls das *Pairing* beobachtet wird. Neben allgemein bekannter Informationen wie die Adressen der Geräte sind hierzu sind die Zufallszahlen IN_RAND, LK_RAND_A und LK_RAND_B (jeweils mit K_{init} verschlüsselt), AU_RAND_A und AU_RAND_B, sowie die Ergebnisse $SRES_A$ und $SRES_B$ nötig.
Man beginnt mit einer Annahme für den PIN (in der Regel 0) und berechnet hierzu den K_{init}, mit dem man dann das erste Paar an Zufallszahlen entschlüsselt. Aus diesen kann man wiederum den *Link Key* berechnen, um mit ihm das zweite Zufallsvariablenpaar zu verschlüsseln. Stimmt dieses mit den Ergebnissen $SRES_A$ und $SRES_B$ überein, so verfügt man über den verwendeten PIN (siehe *Abbildung 10*), andernfalls wiederholt man den Prozess mit einem weiteren PIN.

Da sowohl die Authentifizierung zwischen den Geräten, als auch die Verschlüsselung der Verbindung auf dem PIN aufbaut, ist es nun ein leichtes verschlüsselte Übertragungen zu belauschen oder sich gar als einer der Teilnehmer auszugeben. Ferner können diese Berechnungen sehr effizient implementiert werden, so dass man kurze PINs auf moderner Hardware innerhalb weniger Sekunden knacken kann. Demonstriert wird dies unter anderem durch das Tool *BT-Crack* [15], welches genau nach diesem Prinzip arbeitet und frei zugänglich ist.

4.2 Erzwingen eines erneuten Pairings

Da Geräte einen einmal berechneten *Link Key* in der Regel auf Dauer abspeichern, ist es nicht immer möglich, ein *Pairing* zu beobachten. Dies ist allerdings für den zuvor beschriebenen Angriff nötig, da man nur so in den Besitz der benötigten Informationen gelangen kann. Hierfür kann man allerdings die Tatsache ausnutzen, dass es möglich ist, den *Link Key* zu löschen. Dies ist zum Beispiel der Fall, wenn der Benutzer das *Pairing* an seinem Gerät löst [1].
Wenn nun das andere Gerät versucht, eine Verbindung aufzubauen, wird es wie üblich eine AU_RAND Zufallszahl schicken, nur das dieses mal ein *LMP_not_accepted* Paket als Antwort hierauf geschickt wird. Wenn ein Angreifer nun also zumindest einen Verbindungsaufbau beobachten kann und zusätzlich in der Lage ist, eine *Packet Injection* durchzuführen, kann er dem eigentlichen Slave zuvorkommen, und *LMP_not_accepted* schicken, was das *Pairing* löst. Des weiteren kann ein Angreifer auch ein falsches $SRES$ injizieren oder ein IN_RAND Paket schicken, bevor die AU_RAND Zufallszahl gesendet wurde [3].
Alle drei beschriebenen Techniken haben zur Folge, dass die Geräte erneut ein *Pairing* durchlaufen, und man den eigentlichen Angriff auf den PIN starten kann.

doi: 10.2312/NET-2013-02-1_03

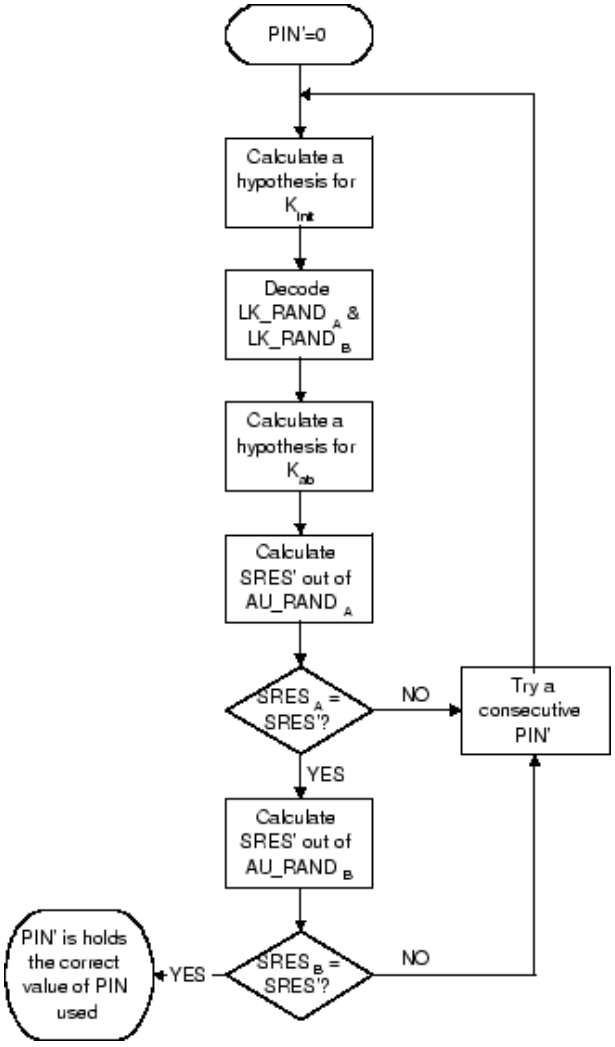


Abbildung 10: Ablauf des Angriffes [3]

4.3 Modifizierungen des Bluetoothprotokolls

Für bestimmte Anwendungen kann es von Interesse sein, das Bluetoothprotokoll in leicht modifizierter Version anzuwenden. Da Bluetooth für eine große Anzahl an Anwendungsmöglichkeiten entwickelt wurde, ist es nicht immer optimal für einen bestimmten Zweck.
So können Sicherheitsfunktionen auf höheren Schichten implementiert worden sein, welche eine erneute Authentifizierung und Verschlüsselung auf niedrigeren Schichten obsolet machen. Außerdem kann es ausreichend (oder sogar von Vorteil) sein, stets eine feste Sprungfolge zu verwenden, so dass es eine Reihe von Systemen komplett störungsfrei nebeneinander betrieben werden können.
Da solche Modifizierungen tiefer in die Abläufe eingreifen, braucht man spezielle Hardware, die unter anderem *Monitoring* und *Injection* unterstützen muss.

5. ZUSAMMENFASSUNG

Es besteht ein Bedarf an Bluetooth *Monitoring* und *Injection* Lösungen, welche auch für die Allgemeinheit zugänglich sind. Professionelle Hardware erfüllt zwar alle technischen Anforderungen problemlos, ist aber auf Grund ihrer sehr hohen Kosten hierfür nicht geeignet. Das *Ubertooth Project* ist hier eine sehr viel versprechende Plattform, insbesondere da sie (trotz gewisser Einschränkungen) sehr gut auch an speziellere Bedürfnisse angepasst werden kann und gleichzeitig sehr günstig ist. Leider kann mit bisher die Möglichkeiten der Hardware noch nicht voll ausnutzen, weshalb zu hoffen bleibt, dass die Entwicklung zügig voranschreitet.
Insbesondere auf die Sicherheit von Bluetooth dürfte sich die Weiterentwicklung des *Ubertooth Projects* positiv auswirken, da hier große Schwächen vorliegen, welche durch die Verbreitung von Bluetooth Monitoring und Injection stark in den Vordergrund rücken werden.

6. LITERATUR

[1] IEEE Computer Society. IEEE Standard 802.15.1-2005, 2005.

[2] Greg Hackmann. 802.15 Personal Area Networks. `http://www.cse.wustl.edu/~jain/cse574-06/ftp/wpans/index.html`, 2006. [Online; Stand 23. September 2012].

[3] Yaniv Shaked and Avishai Wool. Cracking the Bluetooth PIN. In *in Proc. 3rd USENIX/ACM Conf. Mobile Systems, Applications, and Services (MobiSys*, pages 39–50, 2005.

[4] Wireshark. `http://www.wireshark.org/`, 2012. [Online; Stand 23. September 2012].

[5] Kismet. `http://www.kismetwireless.net/`, 2012. [Online; Stand 23. September 2012].

[6] Aircrack-ng. `http://www.aircrack-ng.org/`, 2012. [Online; Stand 23. September 2012].

[7] Michael Ossmann. ShmooCon 2011: Project Ubertooth: Building a Better Bluetooth Adapter. `http://www.youtube.com/watch?v=KSd_1FE6z4Y`, 2011. [Online; Stand 23. September 2012].

[8] Ellisys. Ellisys Bluetooth Explorer 400 Datasheet. `http://www.ellisys.com/products/download/bex400_brochure.pdf`, 2010.

[9] Frontline Test Equipment. ComProbe BPA500 Datasheet. `http://www.fte.com/docs/datasheet_bpa500.pdf`, 2012.

[10] Michael Ossmann. Ubertooth Project. `http://ubertooth.sourceforge.net/`, 2012. [Online; Stand 23. September 2012].

[11] Michael Ossmann. Ubertooth One. `http://ubertooth.sourceforge.net/hardware/one/`, 2012. [Online; Stand 23. September 2012].

[12] International Electrotechnical Commission. Radio-frequency connectors, 1979.

[13] Texas Instruments. CC2400 Datasheet. `http://www.ti.com/lit/ds/symlink/cc2400.pdf`, 2008. [Online; Stand 23. September 2012].

[14] Texas Instruments. CC2591 Datasheet. `http://www.ti.com/lit/ds/swrs070a/swrs070a.pdf`, 2008. [Online; Stand 23. September 2012].

[15] Thierry Zoller. BTcrack OSS - Source code. `http://blog.zoller.lu/2009/02/btcrack-oss-source-code.html`, 2009. [Online; Stand 23. September 2012].

doi: 10.2312/NET-2013-02-1_03

Radio-Frequency Identification - Overview

Lukas Grillmayer
Supervisor: Dipl.-Inf. Matthias Wachs
Seminar Future Internet, WS 2012/13
Chair for Network Architectures and Services
Department of Informatics, Technical University of Munich
Email: grillmay@in.tum.de

ABSTRACT
Radio-frequency identification (RFID) is a wireless technology for automatic identification using electromagnetic fields in the radio frequency spectrum. In addition to the easy deployment and decreasing prices for tags, this technology has many advantages to bar codes and other common identification methods, such as no required line of sight and the ability to read several tags simultaneously. Therefore it enjoys large popularity among large businesses and continues to spread in the consumer market. Common applications include the fields of electronic article surveillance, access control, tracking, and identification of objects and animals. This paper introduces RFID technology, analyzes modern applications, and tries to point out strengths and weaknesses of RFID systems.

Keywords
RFID, transponder, active / passive tag, Auto-ID, Smart Card, EPC, NFC

1. INTRODUCTION
During the last decades bar codes have probably been the most successful identification technology, as other procedures didn't yet exist or were too expensive to be deployed efficiently, although technologies like RFID have many advantages. RFID technology is cheap, fast, does not require a line of sight and supports simultaneous reading of several tags at once over distances ranging from few centimeters to hundreds of meters. Due to technological progress the technology behind RFID has improved, lowering the costs to reasonable levels allowing it to enter the mass market. Every RFID system consists of a reader, a transponder, and an application software. The transponder is a data-carrier which is commonly called tag. They come in various shapes and sizes, mostly in form of a so called smart cards or smart labels, which are sticky and can easily be attached to objects. Widespread applications can be found in the fields of anti-theft, access control, logistics, supply chain management, animal identification and tracking, and electronic payment systems, to name few examples for the countless uses of RFID.
In the next section the history of RFID technology is presented, followed by an introduction to the principles and components of RFID systems along with important standards, which specify the basics of RFID. Afterwards modern applications are analyzed before introducing possible risks and attack vectors which can result out of the use of RFID technology.

2. HISTORY
The invention of radar in the mid 1930s made it possible to detect aircraft from vast distances, but during World War II the issue of distinguishing friendly from enemy planes became a problem. For this purpose the Germans would fly upside-down on command before entering friendly airspace, altering the reflected radio signal to identify the aircraft as friendly [1]. One can call this the first passive RFID system, which was simplistic to spoof by Allied forces. This called for a more sophisticated methods of identification and as a result IFF (Identify Friend or Foe) systems were developed, for example the German FuG 25a "Erstling" manufactured by GEMA in 1941 which sent back a pre-defined signal to ground radar stations when asked for identification [2, 3]. This method of automatic identification via radio-frequency signals is one of the first active RFID systems. In 1948 Harry Stockman described in his publication "Communication by Means of Reflected Power" basic theoretical principles for passive RFID systems and suggested more active foundational research in this area [4]. Radio-frequency identification was invented [5].

More experimental work followed in the upcoming decades, such as Donald B. Harris' "Radio Transmission Systems with Modulatable Passive Responder" in 1960. Harris patented a wireless communication system similar to the military's "Walkie Talkie" systems with the main difference that only one station needs an external power source whereas the portable station draws the required energy to reply out of the received radio signals [6]. Another RFID-related development was Robert M. Richardson's invention "Remotely Actuated Radio Frequency Powered Devices" primarily focusing on efficient usage of radio frequency energy "approaching the theoretical maximum" [7].

As RFID technology was mostly used by the military, in the 1960s first companies recognized the advantages it brings for civilian commercial usage. One of the most widespread and well known inventions of this time were RFID based electronic article surveillance (EAS), to prevent theft and loss of merchandise, which was developed by Arthur Minasy later the founder of the company "Knogo" (1966). Also "Sensormatic Electronics Corporation" (1960) and "Checkpoint Systems, Inc." (1969) competed in this area of EAS systems, which mostly uses 1-bit passive tags and several readers called gates which form a detection zone. If a functional tag is detected in this area an alarm is triggered to signalize an unauthorized passing, whereas no radio signal

doi: 10.2312/NET-2013-02-1_04

can be received when the tag is either destroyed, deactivated or removed [8, 9, 10, 11].

In 1975 three scientists from the Los Alamos Scientific Laboratory published their work on "Short-Range Radio-Telemetry for Electronic Identification, Using Modulated RF Backscatter" and presented an innovative method of communication for passive RFID tags. They were developing a passive "electronic identification system for livestock", which reported the animals identification number as well as its body temperature over a distance of several meters between tag and reader [12]. During the 1970s further novel developments of RFID systems included also the fields of vehicle identification and access control systems [13, 14].

So far RFID technology was mostly a subject of research and development, whereas widespread commercial applications were yet to come in the following decades. Since the late 1980s several countries began to deploy electronic toll collection systems, first of all Norway in October 1987, soon followed by the United States, Italy, and France [15]. As there was only little competition in this market section prices for RFID systems were very high, hindering it from becoming a mainstream technology, although improvements were made considering the size, weight and range of tags. This enabled applications such as animal tracking with implanted tags under the skin, and container tracking which was a development of the Association of American Railroads and the Container Handling Cooperative Program [16].

The 1990s and 2000s are characterized by RFID technology becoming part of everyday life of consumers and the first establishment of industrial standards along with governmental regulations concerning the power and used frequencies of RFID systems. New applications developed in the 1990s included systems for electronic toll payment, ski passes, vehicle access and article tracking. The Auto-ID Center at the Massachusetts Institute of Technology was founded in 1999 supported by the EAN International (today known as GS1) and Uniform Code Council Inc. (UCC, today known as GS1 US) along with Procter & Gamble and Gillette in order to develop the Electronic Product Code (EPC) and standards for low-cost UHF RFID tags used in supply chain applications [17]. As the Auto-ID Center gained supporters and became a worldwide operating research facility, the need for global RFID standards was recognized and resulted among other developments in two air interface protocols (Class 0, Class 1) and the EPCglobal Network, a multicorporate network for real-time tag tracking via the Internet. Until the late 1990s most aspects of modern RFID technology were standardized. These standards applied to animal identification (ISO/IEC 11784, 11785 and 14223), contactless smart cards (based on ISO/IEC 7810: ISO/IEC 10536, 14443 and 156693), container identification (ISO 10374), anti-theft systems for goods (VDI 4470) and item management (ISO/IEC 18000, EPCglobal Network) [18]. All standards are subject to continuous revisions as technology progresses.

In 2003 the UCC and EAN International formed the nonprofit organization EPCglobal Inc. to commercialize the EPCglobal Network internationally. It kept developing new protocols parallel to the combined efforts of the International Standards Organization (ISO) and International Electrotechnical Commission (IEC). Although EPCglobal tried to create the ISO compatible Gen2 standard it took until 2006 before it was merged wth the ISO/IEC 18000 standards [19, 20]. In the meantime large organizations like Wal-Mart and the US Department of Defense set up mandates, which required many of their suppliers to apply RFID tags on their shipments. [16] In 2004 the US Food and Drug Administration (FDA) eased the way for human RFID implants containing a unique ID number, which lead to ethical discussions lasting to this present day [21]. Further current RFID applications will be discussed in the following section.

3. TECHNOLOGY AND STANDARDS

Every RFID System is formed out of a reader/writer, transponders, which are commonly called tags, and an application for further processing of the read data. For instance the application can be an access control system or a supply chain management system. In the following various classes and aspects of RFID systems are described, in order to develop a basic understanding of this technology.

3.1 Transponder / Tag

A tag or transponder contains always an antenna for receiving and sending signals and a silicon chip which holds (and processes) the data to be transmitted to the reader. Tags are usually of a flat appearance, often deployed on plastic foils or paper for smart labels, in plastic casings for access cards and car keys, or in a glass housing for the use as implant [18].

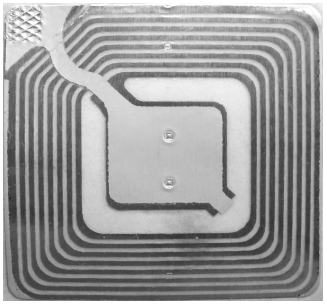


Figure 1: RFID sticker on a DVD for EAS [22]

As described earlier EPCglobal Network introduced classes to characterize the functionality of RFID tags. With the introduction of new classes more features were added to the previous classes.

Table 1: RFID transponder classes[23]

#	Layer Name	Added Functionality
1	Identity	passive identification tags
2	Higher Functionality	read/write memory
3	Semi-Passive	battery power for chip
4	Active AdHoc	communication with tags
5	Reader	can act as reader

Another class of transponders implentents the "Generation 2" (Gen2) standard which was introduced in 2004 and can be described as an improved class 1 tag with additional features as described below [18, 24].

doi: 10.2312/NET-2013-02-1_04

3.1.1 Class 1: Identity Tags

Class 1 tags are passive-backscatter tags, which have no power source, so they draw power from the reader's emitted radio signal in order to reply. Because only a fraction of the emitted power can be used, the reflected energy is very weak, which limits the maximum distance between a passive tag and reader to a few meters. Advantages are low production costs, their small size and very high durability, which allows commercial mainstream mass market deployment for example for electronic article surveillance [25].
Identity Tags are writable once and carry one 64-bit or 96-bit EPC identifier, one tag identifier for manufacturer identification and an implementation of a kill-command, which disables the tag permanently. Optional features include password protected access-control, user memory and the functionality of turning the tag temporarily on and off. This class of transponders operates at 860-930 MHz and 13.56 MHz [18, 26].
Please note that former class 0 and class 1 were merged to class 1 since both had the same functionality of being passive and writable only once [25].

3.1.2 Gen2 Tags

Gen2 tags have a dense reader mode to prevent multiple readers from interfering with each other, improved memory segmentation, barcode applications allowing identical electronic product codes (EPC) on several tags, and changed bit coding to improve detection rates. Another feature of Gen2-tags allows global usage as it meets European (860 MHz - 868 MHz) and North American (902 MHz - 928 MHz) radio frequency regulations and operates at any frequency between 860 MHz and 960 MHz. According to an RFID hardware manufacturer [27] ICs on Gen2 tags have 40-50 year data retention and support up to 100,000 write cycles. The tag's memory is segmented in reserved memory, for 96 bit EPC, 32-64 bit tag identifier, 32 bit kill password and 32 bit access password, and optional user memory ranging up to several kilo bit.

3.1.3 Class 2: Higher Functionality Tags

In addition to class 1, Higher Functionality Tags have an extended tag identifier, extended rewritable user memory up to 65kbit and authenticated access control. Further features of class 2 transponders are not yet specified in this standard [27].

3.1.4 Class 3: Semi-Passive Tags

These tags have a power source, usually in form of a battery, for supplying energy to the circuitry of the tag. It has to draw energy from the reader only for communication purposes, allowing longer ranges and more reliability concerning the successful transmission of data. The life time of class 3 tags is generally shorter than for entirely passive tags [25].

3.1.5 Class 4: Active AdHoc Tags

Active RFID tags are equipped with an autonomous transmitter so they need their power source for communication purposes as well. This enables long reading ranges up to several kilometers using the microwave band. The simultaneous reading of several hundred tags at once, high data transfer rates and the possibility to build complex systems by adding sensors made this technology especially attractive for container tracking, refrigerated transport and other fields of logistics [25]. Drawbacks are lifetime cycles of 5 - 10 years, mostly limited by the battery life, and high costs of about $10-$50 for one active RFID tag [28, 29, 30].
In addition to class 3, Active AdHoc Tags can communicate with other class 4 tags through the on-board transmitter. Therefore class 4 tags are able to initiate communication, whereas tags of class 3 and below can only answer to the reader which provides the power for communication [18, 27].

3.1.6 Class 5: Reader Tags

Class 5 supports all features of class 4 tags and has the ability to act as reader for passive and active tags of classes 1, 2, 3, and 5 [18].

3.1.7 Power Supply and Antenna Design

As mentioned before passive tags have no battery to power communication functionalities whereas active tags have an on-board power source. In the following, various methods to supply power to passive transponders and the resulting antenna design are described. Antennas of active tags may differ in size and shape, but the presented principles apply to active tags as well as to passive tags.

1-bit tags, which are mostly used in EAS, use fairly simple physical exploits as they only have to communicate their presence and no further information. In systems using radio frequency for tag detection the transponder has coils, which generate a current on the tag when it is located in the magnetic alternating field emitted by the reader (see Figure 1). This current causes a weakening of the magnetic field which can be meassured.
Tags in EAS systems operating at microwave range use a dipole antenna connected to a diode. When the tag is in the range of the reader, the emitted carrier frequency causes a flow of current in the diode which re-emits harmonics of the carrier wave, which can be measured. Similar systems work with subharmonics or even inharmonics of the carrier frequency.
A different approach is the use of acoustomagnetic systems, where tags are made of two special metal strips contained in a plastic housing. When placed in a magnetic alternating field one metal strip starts to vibrate and it keeps oscillating for a while when the magnetic field is turned off. This causes the tag to emit a weak, but measurable magnetic field for a short period of time.

Transponders supporting full- or half-duplex comunication use different principles than 1-bit tags since they need more energy to power their chip. Tags using inductive coupling draw power of the reader's emitted magnetic alternating field through the use of coils. The resulting current powers the chip and two parallel capacitors, which form the resonant frequency of the tag.
Electromagnetic backscatter coupling is used for long-range systems (> 1 m) operating at UHF frequencies. Long distances between reader and tag result in a high loss of energy of the emitted electromagnetic field, which has to be considered for the design of the tags dipole antenna.
Close coupling is used for ranges between 1mm and 1 cm, therefore the energy loss over distance is minimal. These tags are usually inserted in or placed onto the reader. The

doi: 10.2312/NET-2013-02-1_04

reader has coils on a ring-shaped or U-shaped core, whereas the transponder's coils can be placed coplanar. The energy transfer is similar to the principle of transformers.
At last tags for systems using electrical coupling have an antenna, which is made of two large conductive surfaces, each acting as an electrode. When the tag is placed in the electric field of the reader, there is a voltage between the electrodes, which can be transformed to a lower voltage to power the transponder's chip [18].

3.2 Reader

A Reader is responsible for the communication with tags and the RFID system's application software. It provides an electromagnetic field to activate the tag and initiates the communication process with one or more tags in range. This field is used by passive and semi-passive tags to power their communication ability as they do not have an onboard power source for this purpose. Readers handle the connection establishment, and anticollision and authentication procedures exclusively. In order to write data to a tag's read/write memory, special readers can also act as writers. Modern readers are able to connect directly to a network or the internet to communicate with the application software using IP, TCP, and UDP, or they are connected to a computer, which redirects or processes the gathered data via an USB or RS232 connection. There are handheld models and stationary readers of which latter ones might support multiple antennas or can be deployed in an array of readers forming a larger detection zone.
A current development is the integration of RFID technology in cell phones and tablets, enabling mobile devices to operate as reader and tag. These applications use the Near Field Communication (NFC) standard which relies on RFID technology [18].

3.3 Frequency

The name "radio-frequency identification" already indicates that RFID operates in parts of the radio frequency spectrum which ranges from about 3 kHz to 300 GHz, although mostly frequencies between 30 kHz and 6GHz are used in today's RFID applications. In the following, four frequency bands will be discussed as can be seen in table 2 in respect to ISO/IEC Standards, which will be presented later in this paper. Please note that the use of various frequencies and maximum signal strengths are heavily regulated by governments all over the world. Details to read ranges are exemplary as read ranges improve with technological progress and can be extended using more sophisticated technologies.

Table 2: Frequency Ranges used for RFID

Frequency	Abbr.	Designation
30 kHz - 300 kHz	LF	Low Frequency
3 MHz - 30 MHz	HF	High Frequency
300 MHz - 3 GHz	UHF	Ultra High Frequency
2 GHz - 30 GHz	-	Microwave

3.3.1 Low Frequency

Applications using LF operate usually below 135 kHz and are characterized by short reading ranges of < 0.5m and a low data transfer rate of <1 kbit/s. Performance issues arise when metal objects are disturbing the transmission. Typical applications of LF RFID systems are animal tracking and access control [25].

3.3.2 High Frequency

The very common RFID frequency of 13.56 MHz can be found in the HF band. With a read range of < 1.5m and decent data transfer rates of about 25 kbit/s it very suitable for the deployment of smart labels and product authentication. Once again it cannot penetrate metal [25].

3.3.3 Ultra High Frequency

UHF RFID systems use the following frequencies: 433.92 MHz and 860 MHz to 930 MHz. Data transfer rates are about 30 kbit/s. This band is supported by the Electronic Product Code which is currently used by Wal-Mart and the US Department of Defense in their supply chains. Read range is up to 100m for 433.92 MHz and between 0.5-5 m for 860-930MHz.Water and metal cannot be penetrated [25].

3.3.4 Microwave

Microwave systems operating with frequencies of 2.45 GHz or 5.8 GHz have a long read range of 10m and above, and a high data transfer rate of about 100 kbit/s. This technology is used in toll collection systems all over the world with the drawbacks of being more expensive than other systems and the lack of water and metal penetration.

LF and HF RFID systems can be counted to near field communication systems, using inductive coupling from a magnetic field, whereas UHF and microwave systems use far field communication with backscattering of radio waves [25].

3.4 Standards

As RFID technology began to spread rapidly all over the world, global standards needed to be accomplished in order to clearly define RFID technology as multiple standards serve as entry barrier. Standards were developed by several organizations such as the International Standardization Organization (ISO) in corporation with the International Electrotechnical Comission (IEC) called ISO/IEC, and as earlier introduced EPCglobal Network. Both developed their standards hoping for an adoption to a global standard, which was accomplished in 2006 [16]. In the following several important RFID standards are introduced.

3.4.1 ISO/IEC 14443 - Smart Cards

ISO/IEC 14443 "Identification cards - Contactless integrated circuit cards - Proximity cards" is a standard for contactless smart cards with a range of 7-15cm. This standard is commonly utilized in ticketing and payment systems later described in this paper [31].

ISO/IEC 14443 - Proximity-Coupling Smart Cards:
Part 1: Physical characteristics
Part 2: Radio frequency power and signal interface
Part 3: Initialization and anti-collision
Part 4: Transmission protocol

The first part describes the physical characteristics of the tag, such as standardized size and shape as well as material properties [18].
Part two specifies this type of tag to be passively powered

doi: 10.2312/NET-2013-02-1_04

at a frequency at 13.56 MHz and introduces two communication interfaces, which are not interchangeable during an ongoing transmission. Type A and Type B operate at the same subcarrier frequency of 847 kHz for tag-to-reader communication and have the same data transmission rate of 106 kbit/s for any directions of communication whereas they differ by used modulation and synchronization methods. Type A implements frame synchronization by detecting start-of-frame and end-of-frame marks, Type B relies on one start- and one stop-bit per byte [18].

Next "Initialization and Anticollision" introduces frame structures and anti-collision procedures for each type. Collisions occur when more than one tag tries to communicate with the reader at the same time.
When a type A card enters an active read range of a reader, it puts itself into the IDLE mode, not interrupting an ongoing communication. Now the reader can issue a REQA command (Request-A) which is replied by an ATQA block (answer to request) causing the card to switch to the state READY. The reader triggers the anti-collision algorithm based on a dynamic binary search tree by sending a SELECT command. After having chosen the tag for communication the reader issues a SELECT command with the target tag's serial number. The selected card switches into ACTIVE mode and is now ready for transmission.
Type B also switches into IDLE mode when entering an interrogation field waiting for a REQB (Request-B) command which triggers the anti-collision algorithm utilizing a dynamic slotted ALOHA procedure with parameters dictated by the reader. The REQB frame includes two parameters, one for choosing the card's application group for preselecting and one for communicating available slots. After a random waiting interval the card starts to transmit when a slot marker indicated the start of a new slot. The tag sends an ATQB (answer to request B) command, receives additional protocol parameters (ATTRIB), then switches into ACTIVE state and is now ready for transmitting. Readers can cut idle slots by issuing a new slot marker to save time [18].

Lastly transmission protocols are introduced for transmission error handling. ISO/IEC 14443-4 is based upon a standard for non-wireless contact smart cards (ISO/IEC 7816-3) which is an advantage for the implementation of dual interface smart cards as both protocols are compatible. Type A cards need additional information about protocol parameters, which Type B cards received with the ATTRIB command, FSDI (frame size device integer) which defines the maximum frame size and an optional CID (card identifier) for addressing an individual card in the interrogation zone. After ensuring the availability of the 14443-4 protocol data is sent using the following frame structure [18, 32]:

PCB	**[CID]**	**[NAD]**	**[INF/APDU]**	**CRC**
1 byte	1 byte	1 byte	N byte	2 bytes

Figure 2: Frame structure in ISO/IEC 14443

The PCB field (Protocol Control Byte) provides additional protocol parameters, NAD (Node ADress) is optional, and INF (Information) / APDU (Application Protocol Data Unit) is optional and forms the payload of the frame. Lastly CRC is used for error detection in the payload [18, 32].

3.4.2 ISO/IEC 18000 - Air Interface Standards

In 2004 a committee created by ISO and IEC released the so called RFID Air Interface family of standards: ISO/IEC 18000 Series, which standardized transponders and defined an "Air Interface" to be used for communication between tags and readers at various frequencies. This standard has been modified several times, adapting to the fast development progress of RFID technology.

ISO/IEC 18000 - RFID for Item Management:
Part 1: Generic parameters for Air Interfaces for globally accepted frequencies
Part 2: Air Interfaces below 135 kHz
Part 3: Air Interface at 13.56 MHz
Part 4: Air Interface at 2.45 GHz
Part 5: Air Interface at 5.8 GHz
Part 6: Air Interface at 860 MHz - 930 MHz
Part 7: Air Interface at 433.92 MHz

3.4.3 ISO/IEC 15691 and 15692

These standards define functionalities of readers and the communication interface between readers and applications. ISO/IEC 15691 defines commands, responses and error messages, which can be used for communication between the application host and the reader. For writing data from the reader to a tag, ISO/IEC 15692 defines memory mapping rules. This is necessary since the tag's memory is organized in blocks and segments so data can only be transferred in blocks. Therefore the reader preprocesses and segments the data according to ISO/IEC 15692 before sending it to the tag [18].

3.4.4 Electronic Product Code (EPC)

When applying RFID technology to countless products produced by thousands of companies it seemed important to introduce a numbering scheme to RFID tags to be able to track back where a certain product came from. Therefore the MIT Auto-ID Center developed the Electronic Product Code (EPC), a standard for numerous numbering schemes. An example would be the General Identifier (GID-96) which can be seen as an improvement to the existing barcode numbering scheme (UPC) which uniquely identified manufacturer and product. In addition to this functionality GID-96 can also identify every single article of each product group, so every item has its own unique number.

Header	**EPC Manager**	**Object Class**	**Serial No.**
8 bit	28 bit	24 bit	36 bit

Figure 3: General Identifier Format (GID-96)

Figure 3 illustrates the used numbering scheme: The header field describes the used scheme, for example for GID-96 this would be 0x35. The EPC Manager field is a registered number unique for every manufacturer, Object Class contains an identifier for the object class i.e. "Some-Brand's Cookies", and the Serial Number field gives every single unit of this object class a unique number, so in this example every package of cookies could be uniquely identified. Other common EPC numbering schemes are the serialized global

doi: 10.2312/NET-2013-02-1_04

trade item number (SGTIN), serial shipping container code (SSCC), serialized global location numer (SGLN) and the global individual asset identifier (GIAI) [33].

4. RFID APPLICATIONS

Modern RFID technology is already deployed in numerous fields: asset utilization, asset monitoring and maintenance, item flow control in processes, inventory audit, theft control, authentication, payment systems, etc. Currently production costs for low-cost passive RFID tags are about 5 cents, with continuing research further improvements in range, size, costs can be expected, as RFID experts dream of a 1 cent tag finally making the costs neglectable and allowing RFID to be deployed ubiquitously [34].

4.1 Electronic Passport

Since August 2006 all 24 EU members are required to integrate ISO/IEC 14443 conform RFID technology into the so called ePassport. The main purpose of this development is to improve passport security against forgery. A chip integrated into the data page or the cover stores personal information about the holder, such as name, date of birth, gender, biometric features of the face, and since 2008 fingerprints. This information uses about 32kByte of memory, which is a specified minimal requirement. The contents are secured against forgery by a digital signature which can be checked with a public code from the country's signing certification authority to ensure data integrity and authenticity. Due to a data transfer rate of 848 kBit/s at a read range of 10 cm this technology allows border controls at airports to process more travelers in less time. RFID technology is ideal for this purpose, as it is cheap, fast, easy to integrate into a passport, and makes forgery of passports more difficult. Several countires outside the EU use this technology as well, for example Japan, Singapore, and the US [18].

4.2 Credit Cards

In 2005 several major credit card organizations started to offer contactless credit card payment to their customers. Embedded passive RFID tags operating at a frequency of 13.56 MHz, allow the card holder to pay on the fly, by placing the credit-card-containing wallet close to the reader (max. 4 cm) which then requests the stored information required for a successful transaction. Worldwide established systems are called PayPass (MasterCard), ExpressPay (American Express), and payWave (Visa), and follow the ISO/IEC 14443 standard for identification cards, contactless integrated circuit cards and proximity cards [18]. RFID technology is well suited for the purpose of making transactions easier and faster. As this application is especially prone to attacks, payments neither requiring a signature nor a PIN are limited so small amounts such as $25 whereas larger values need to be authorized by the card-holder [31]. Possible attacks and securities issues as well as solutions to the problem of vulnerability of contactless credit cards are discussed later in this paper.

4.3 Access Control

One of the most popular applications of RFID technology is access control. Compared to other systems it has many advantages in addition to operating wirelessly, such as very high adaptability. Keys can easily be excluded from the system, permissions can be altered, temporary access can be granted, the system can be extended by additional cards and readers, etc. All of the mentioned changes can be made very fast and easily. Online systems are formed out of cheap tags which only carry a number, which can be received by a reader connected to a database via a network. The database contains permissions associated with tag numbers, telling a door to open or to remain closed. In offline systems the key contains the data needed for access. The stored data can be altered and in case of a lost key, every reader has to be informed about which key will be excluded.
RFID enabled electronic access control systems are very useful since a large number of people with different access authorizations can carry a single key without having to plan a complicated lock system before installing it. For example the cleaning personnel can receive keys which only open doors at nighthours or temporary workers can receive a time restricted key [18].

4.4 Prospects

Today and in the near future much more functionality will be transferred into mobile phones with internet access using NFC technology, enabling mobile devices to act as credit cards, tickets for public transportation or public events, access control cards, tracking device, etc. This is one of the developments contributing to the Internet of Things (IoT), described in the IoT action plan for Europe by the Commission of the European Communities [35]. As legal restrictions are starting to allow the deployment of RFID technology in humans, future applications will most certainly include the wide spread use of human implants for previously named applications and beyond [21].

5. RISKS AND ATTACK VECTORS

With RFID technology increasingly being deployed in security relevant applications such as anti-theft, access control, and electronic payment systems, it is prone to be attacked. In the following various attack vectors are presented along with possible countermeasures.

5.1 Basic / Physical Attacks

Attacks on RFID systems on the physical layer can be very simple. Since only a fully operable tag can be detected by a reader in range, removing the tag from the tagged object is sufficient to fool for example an EAS systems. Common countermeasures implemented in tags are acoustic alarms or the release of colour from inside the tag when someone tries to remove it improperly, which draws attention to a probable thief or permanently damages the object it was attached to. Tag destruction can be achieved through the application of too much mechanical force or for most RFID tags by applying a very strong electromagnetic field, which can destroy an essential part of the tag. Since most transponders are very fragile, they must be protected by a more robust design or by placing it inside the tagged object. The proper way to permanently disable a transponder is to issue the kill-command which is specified by EPCglobal. The unautherized execution of the kill command can be prevented by implementing a sufficiently strong password protection, which is required for Gen2-tags.
Temporarily disabling tags can be achieved as easily as by

doi: 10.2312/NET-2013-02-1_04

shielding the tag with common aluminum foil. As mentioned before electromagnetic waves cannot efficiently penetrate metal, therefore the tag cannot communicate with the reader. Attackers can also jam the frequency range, which is used by tag and reader by applying a signal in the same frequency range. This interferes with communication signals and as a result the reader receives only noise. The described effects can also be caused by unintentionally placing the tag in a shielded area or by a noisy environment. Detection of disabled tags by an RFID system is impossible, therefore other means of security, such as video cameras or locked doors should be used additionally in order to prevent physical attacks [36].

5.2 Eavesdropping

The interception of communication (eavesdropping) is one of the most popular attacks on wireless systems. Data is intercepted at some point in the range of tag and reader, an area which can range from few square centimeters up to several hundred square meters, which can even be significantly extended by using more sensitive antennas such as beam antennas. Detection of useful data from vast distances depends on many environmental factors like the presence of metal objects or water can weaken and block the sent signals [18].
After successful interception, the received information can also be used in more sophisticated attacks such as cloning and replay attacks. Car theft can be an application for replay attacks, which can be stolen by replaying the signals, intercepted when the car owner opened or closed the car using a remote wireless key using RFID technology [37].
An exemplary countermeasure is the limitation of signal strength and use of shielding or directional transmission to restrict the reading range to the actually needed area. The deployment of multi-channel communication and communication protocols with strong security policies can prevent replay attacks but realizing secure protection against eavesdropping itself is very difficult if not impossible [18].

5.3 Cloning

Cloning is the task of creating a replica of the original object. This method has been used for decades to duplicate credit and debit cards in the area of credit card fraud for decades as criminals copied the stored information of a credit card and applied the data to a different card which contains all needed data to withdraw money from the victim's account. The original card usually leaves the hand of the card owner and disappears into a machine involved in a financial transaction or a waiter takes the card away to the cashier. In the meantime the card's information can be copied and the regular transaction takes place as expected, arousing no suspicion by the card owner. This acquisition of data always needed mechanical contact with a reader which can possibly be out of sight. As major credit card organizations deployed wireless technology in credit and debit cards, one of the benefits seemed to be that the card never left the owner's hand, was always in sight and could not be mishandled by a corrupt cashier, supposedly increasing the safety concerning credit card fraud. Au contraire this new technology offers a new interface to retrieve information in a split second without somebody possibly noticing it as these cards broadcast their information in every direction through the air to a reader up to 4 cm away when a request was received. Richard Vanderhoof from Smart Card Alliance stated he does not see an increased risk in this use of RFID technology [38].
In June 2012 "Report München" a German television show aired a report about how easily this information can be extracted. They developed an application for a NFC enabled smart-phone and were able to retrieve several sets of credit card information in public by holding the mobile device close to pedestrians' wallets, unnoticed. It is to note that no encryption or password protection was used [39].
Similar methods can be used to clone other RFID tags, like access cards, price tags, etc and to apply the gathered data to similar tags with read/write memory. The usage of cloned tags may be undetectable as for the reader there is no difference in the cloned tag compared to the original.
Ways to avoid tag cloning on the physical layer in advance are the disabling or destruction of the transponder, or to shield the tag by wrapping it into aluminum foil. More applicable is the inclusion of security means in the backend, such as the storage of already read serial numbers as some tags might only be read once, or by disallowing the simultaneous use of supposedly unique tags in two different locations. Common ways to protect data from being accessed without authorization are the use of passwords and encryption on the tag, thus causing a tag's price to increase as usually more hardware is needed [38].

5.4 Virus Attacks

Most RFID are connected to rather complex backend systems, processing the data received through an RFID transmission. In general the communicated tag content contains a unique serial number for identification and some further information about the object. As modern reusable tags have data storage capacities exceeding several kilobytes of read/write memory this information can be altered or completely new tags are introduced to an existing RFID system. Additional or wrong data can be written on the tag causing unintended data to be sent to the backend system which now can perform differently than expected from the initial legit data processing. If this attack scenario has not been accounted for in the development of the affected backend system, additionally received and executed commands have the potential to harm the entire system as the attacker now has control over some system components. Databases could be copied, altered, or deleted (i.e. SQL-injection), web-based components could download hazardous files from the internet containing malware (for example by using enabled JavaScript in Browsers) and even files could possibly be removed or altered (code insertion) causing the entire system to fail.
Virus attacks can be prevented by a well designed system, which does not allow unintended behavior. For example to prevent SQL-injection in databases, developers can use predefined statements, telling the system exactly which kind of data to expect and clearly define how to use it in the next query. As this form of defense does not require additional hardware in tags or readers it does not contribute noticeable to additional costs for an RFID system [38].

5.5 Privacy

With RFID technology spreading rapidly into many aspects of every day life, RFID systems are able to collect enormous amounts of data. As it is a wireless technology the data can be read unnoticed, no contact is necessary and common tags

doi: 10.2312/NET-2013-02-1_04

cannot log who tried to access which information. Countless items are already enabled to act as RFID tag, for example passports, credit cards, ID-cards, customer loyalty cards, library books, event tickets, bank notes, and even regular grocery purchases may contain RFID tags. If the data on those transponders is unsecured or the used security measures are easy to breach, it can be read by anyone in range. Possible privacy violations are apparent in the following example: Someone shops in a large grocery store, which uses RFID everywhere. Customer loyalty cards enable the store to create movement profiles of every single customer and associate items to a person, which tells the store preferences and habits. Even if that person carries other companies' RFID enabled products, this store is able to tell where else this person goes for shopping and what was bought. Bank notes acting as tags could reveal a customers financial situation to anyone in range. Out of collected data one can most certainly conclude information which one never agreed on sharing, which is a privacy violation.
One can avoid privacy issues by designing RFID systems properly with tags supporting password protection and encryption as well as by limiting the read range of reader and tags to the necessary distance not allowing someone to access stored information unauthorized [40].

6. CONCLUSION

RFID technology connects objects with associated data, in other words the real world with the virtual world. Therefore it can contribute vastly to the development of the Internet of Things. Although RFID makes many aspects of life easier and more comfortable, there are always risks involved which should not be underestimated. With a proper system design one can try to use this technology for the better and take advantage of the great opportunities RFID technology provides. With decreasing prices for tags and rapidly developing applications RFID is becoming more and more ubiquitous, it will soon be impossible to imagine a life without RFID.

7. REFERENCES

[1] Andy Jones: *Proceedings of the 3rd European Conference on Information Warfare and Security*, page 81, Academic Conferences Limited, 2004

[2] TM E 11-219 Directory of German Radar Equipment: *Airborne Radar: FuG 25 Airborne IFF Transmitter-Receiver*, April 20, 1945

[3] Jerry Scutts: *German Night Fighter Aces of Word War 2*, page 84, Osprey Publishing, Oxford, GB, 1998

[4] Harry Stockman: *Procedings of the I.R.E. - Communication by Means of Reflected Power*, pages 1196-1204, The Institute of Radio Engineers Inc., NY, USA, October 1948

[5] Jeremy Landt: *The History of RFID*, AUTO-ID Labs at MIT, November 2005

[6] Donald Harris: *Radio Transmission Systems with Modulatable Passive Responder*, United States Patent Office, No. 2927321, March 1, 1960

[7] Robert M. Richardson: *Remotely Actuated Radio Frequency Powered Devices*, United States Patent Office, No. 3098971, 1963

[8] Justin Patton: *RFID as electronic article surveillance: Feasibility assessment*, University of Arkansans, AR, USA, December 2008

[9] Daniel F. Cuff: *BUSINESS PEOPLE; An Anti-Theft Specialist Bolsters Its Top Ranks*, New York Times, NY, USA, November 8, 1990, `http://www.nytimes.com/1990/11/08/business/business-people-an-anti-theft-specialist-bolsters-its-top-ranks.html/`

[10] Bloomberg Businessweek: *Company Overview of Sensormatic Electronics, LLC*, `http://investing.businessweek.com/research/stocks/private/snapshot.asp?privcapId=302947/`

[11] Bloomberg Businessweek: *Checkpoint Systems Inc (CKP:New York)*, `http://investing.businessweek.com/research/stocks/snapshot/snapshot.asp?ticker=CKP/`

[12] Alfred R. Koelle, Seven W. Depp, Robert W. Freyman: *Short-Range Radio-Telemtry for Electronic Identification, Using Modulated RF Backscatter*, Los Alamos Scientific Laboratory, University of California, Los Alamos, NM, USA, Febuary 10, 1975

[13] Cardullo et al: *Transponder Apparatus and System*, United States Patent Office, No. 3713148, 1973

[14] Charles A. Walton: *Electronic Identification and Recognition System*, United States Patent Office, No. 3816709, 1973

[15] Timothy D. Hau: *Congestion Charging Mechanisms for Roads: An Evaluation of Current Practice*, The World Bank, Washington DC, USA, December 1992

[16] V. Daniel Hunt, Albert Puglia, Mike Puglia: *RFID: A Guide to Radio Frequency Identification*, John Wiley & Sons, 2007

[17] EPCglobal Inc.: *GS1 EPCglobal - Frequently Asked Questions*, January 2007, `http://www.gs1.org/docs/epcglobal/ Frequently_Asked_Questions.pdf`

[18] Dr. Klaus Finkenzeller: *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near Field Communication*, Wiley, 2010

[19] Mark Roberti: *The History of RFID*, RFID Journal, `http://www.rfidjournal.com/article/view/1338/`

[20] *A Summary of RFID Standards*, RFID Journal, `http://www.rfidjournal.com/article/view/1335/`

[21] Thomas C. Greene: *Feds approve human RFID implants*, The Register, GB, October 14, 2004, `http://www.theregister.co.uk/2004/10/14/ human_rfid_implants/`

[22] Image of an RFID sticker on a DVD for EAS, photo by Lukas Grillmayer

[23] Jongwoo Sung, Daeyoung Kim: *Sensor Profile Requirements for Sensor Network Capability Information in the EPCglobal Network*, Auto ID Labs, Korea Advanced Institute of Science and Technology, March 2009

[24] Bill Glover, Bhatt Himanshu: *RFID Essentials*, O'Reilly Media Inc., 2006

[25] Gaynor Backhouse: *RFID: Frequencies, standards, adoption and innovation*, JISC Tech Watch, May 2006

[26] *EPCglobal Tag Class Definitions*, November 2007, `http://www.gs1.org/docs/epcglobal/ TagClass-Definitions\_1\_0-whitepaper-20071101.pdf`

doi: 10.2312/NET-2013-02-1_04

[27] Sky RFID Inc.: *RFID Gen 2 - What is it? - Smart RFID!*, http://www.skyrfid.com/RFID_Gen_2_What_is_it.php
[28] *RFID System Components and Costs*, RFID Journal, http://www.rfidjournal.com/article/ article-view/1336/1/129/
[29] *What is the Lifespan of an Active Tag*, RFID Journal, http://www.rfidjournal.com/article/view/4673/
[30] *NephSystem Technologies*, http://www.nephsystem.com/
[31] Visa Inc.: *Visa payWave - FAQ*, http://usa.visa.com/merchants/payment_ technologies/paywave_faq.html/
[32] *ISO/IEC 14443-4: Identification cards - Contactless integrated circuit(s) cards - Proximity cards - Part 4: Transmission protocol*, 2008
[33] GS1 EPCglobal: *GS1 EPC Tag Data Standard 1.6*, September, 2011
[34] Mark Roberti: *A 5-Cent Brakethrough*, RFID-Journal, May 1, 2006, http://www.rfidjournal.com/article/ article-view/2295/1/128/
[35] Commision of the European Communities: *Internet of Things - An action plan for Europe*, Brussels, June 18, 2009
[36] Aikaterini Mitrokotsa, Melanie R. Rieback and Andrew S. Tanenbaum: *Classification of RFID Attacks*, http://www.cs.vu.nl/~ast/publications/iwrt-2008.pdf
[37] Aurélien Francillon, Boris Danev, Srdjan Capkun: *Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars*, ETH Zurich, Switherland
[38] Frank Thornton, John Kleinschmidt: *RFID security*, Syngress Pub., Rocklannd, MA, USA, 2006
[39] Ronald Eikenberg: *Kreditkartenklau per Smartphone*, June 6, 2012, Heise Online, http://www.heise.de/newsticker/meldung/ Kreditkartenklau-per-Smartphone-1611874.html
[40] Dirk Henrici: *RFID Security and Privacy*, Springer, May 2008

doi: 10.2312/NET-2013-02-1_04

NFC - Possibilities and Risks

Uwe Trottmann
Betreuer: Matthias Wachs
Seminar Future Internet WS2012
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: uwe.trottmann@tum.de

ABSTRACT
Near Field Communication (NFC) is an emerging close range, low bandwidth, induction based communication standard. It is already and will be more broadly integrated tightly with modern smartphones, devices and operating systems. Payment services, setup of high-bandwidth connections, information sharing and identity verification become possible by just touching two NFC devices together. This paper tries to give an overview over how NFC technology works, what some of its current and potential applications are and which risks and exploits come along with its simplicity.

Keywords
NFC, NDEF, SNEP, payments, ISIS, wallet, ticketing, RFID

1. INTRODUCTION
Near Field Communication (NFC) is a consumer-oriented wireless technology using magnetic fields and induction to communicate data over a distance of centimeters at low bandwidth. It is backed by the NFC Forum, consisting of more than 160 members including industry heavy-weights like Samsung, Sony and NXP. They push and certify integration of NFC technology in modern consumer electronics like smartphones and operating systems like Android and Windows 8. Current applications include payment and ticketing by just waving your phone, the setup of Bluetooth or Wi-Fi connections between devices by touching them together and embedding of information or device configurations in passive entities, so called NFC tags.

The following will give an overview over the history and technology behind NFC (2), showcase the breadth of current and future applications (3) as well as discuss risks, attack vectors and existing exploits (4).

2. HISTORY AND TECHNOLOGY
In 2004 ISO/IEC standard 18092 "Near Field Communication - Interface and Protocol (NFCIP-1)" [2] specified a technology for exchange of information and telecommunications. It is based on and expands on existing RFID standards for contactless cards by ISO/IEC (ISO/IEC 14443 [3]) and Sony (Felicity Card/FeliCa [4], based on JIS X 6319-4 [5]). Implementations and applications in various forms and devices based on this standard are backed and certified by the NFC Forum, an industry consortium which came to life the same year. The NFC Forum now includes more than 160 members like electronics manufacturers Samsung and Sony, chip providers NXP and Broadcom and even financial services companies Visa and Mastercard. Mid-2006 initial specifications for a data exchange format (NDEF, see 2.3) and record types - the first few instances being text, URIs and smart posters - were announced [6].

Both prior standards ISO/IEC 14443 and FeliCa differentiate between dedicated reading and writing devices and integrated circuit cards. The cards are mainly passive objects and do not have a power source of their own. The used fields operate in the globally available, unlicensed ISM[1] band of 13.56 MHz and require small distances in the range of centimeters between the reader and the contactless card.

According to the new ISO/IEC 18092 standard [2], NFC devices operate in the same ISM band of 13.56 MHz and are required to be compatible with ISO/IEC 14443 and FeliCa. Communication involves an active initiator device generating a magnetic field in close proximity to a passive target device, typically at about four centimeters or less. All devices by default are in target mode and wait for an incoming command by an initiator. If a field is noticed by a target device, it will manipulate it to transmit information back to the initiator. This mode is similar to the reader and card scheme in ISO/IEC 14443 and FeliCa and is called passive mode. One should note that using special resonant circuitry a passive target device may be built which does not require its own power source similar to existing contactless cards in various form factors. These are called tags and are discussed in more detail later on (see section 3.1).

In contrast to the previous standards, however, any NFC device may opt to become the active component, the initiator. This active mode enables both devices to take turns in generating their own fields while the other listens for data, effectively establishing a half-duplex connection.

Both the passive and active mode provide data rates of 106, 212 or 424 kbit/s using the Manchester coding scheme with amplitude shift keying for modulation for all mode and data rate combinations. Active mode at 106 kbit/s is an exception, it uses a form of Miller coding (both codings are described in [2]).

NFC connections as specified by ISO/IEC 18092 themselves

[1]Various frequency bands widely reserved for Industrial, Science and Medical use

doi: 10.2312/NET-2013-02-1_05

	NFC	ZigBee	Bluetooth (2.1)	WLAN (802.11ac)
Range	< 10cm	< 100m	< 100m	< 250m
Data rate	<424 kbit/s	<250 kbit/s	<2.1 Mbit/s	<866.7 Mbit/s
Network size	2	2**64	8	2007
Frequency band	13.56 MHz	868/915 MHz, 2.4 GHz	2.4 GHz	2.4/5 GHz

Table 1: NFC compared to other wireless technologies. [7]

do not require any form of encryption. It is of the discretion of the implementing applications to provide any layer of security, which might not be as reliable on a application-to-application basis compared to built-in security. In addition the application layer may be more prone to vulnerabilities and outside influences especially considering an active NFC device always listening and reacting to incoming commands as mentioned before. Serious privacy concerns may arise if personal information can simply be read from a NFC-equipped mobile phone due to sloppy applications handing out data without approval. Or even programs being executed resulting in the infection of the users device. Therefore the last chapter (4) will shed more light on possible and existing attacks.

2.1 Comparison

As Near Field Communication is based on RFID technology it shares similar properties. Taking a closer look at RFID, it is designed to read data from tags or cards via radio-frequency (RF) electromagnetic fields for the purposes of identification or tracking of objects or people. Tags and cards are always passive, only reading devices active. Communication, however, may occur on a wider variety of frequencies, including the 13.56 MHz ISM band NFC uses. More importantly RFID is capable of operating without line-of-sight making it feasible for different automation tasks, like the scanning of stacks of crates passing a RFID reader equipped warehouse gate. In comparison NFC is restricted to very short distances but allows devices to function either as, in RFID terms, a reader or tag/card. [8]

In the following multiple other existing wireless technologies, namely ZigBee, Bluetooth and WLAN are described and compared to NFC as listed in table 1.

ZigBee is a wireless technology designed to support significantly higher ranges with improved maximum data rates requiring only minimal hardware. Like RFID it is mainly used in commercial applications, but seldom in consumer products. ZigBee allows to interconnect devices into a mesh like network while consuming as little energy as possible. It extends the IEEE 802.15.4 standard, operating mainly on 2.4 GHz, and may support thousands of devices in one network, for example to report sensor measurements or automate devices through remote commands. [9]

Bluetooth, specified by the Bluetooth SIG[2], is similar to ZigBee. It, however, restricts the device architecture to one master and up to seven slave devices. More devices may be linked into this Personal Area Network, albeit in a passive, parked mode. The resulting piconet was supposed to get rid of cables required to connect computer accessories like printers or digital cameras. Nowadays, Bluetooth is included in pretty much every mobile handset and is commonly used for file transfers or spontaneous device networks with data rates significantly higher than NFC at a larger range in the ball bark of meters (see table1). However, setup is far more complicated and time consuming requiring pairing and the exchange of secrets. [10]

WLAN was designed to extend the range of LANs to mobile devices like laptops. It is specified as the IEEE 802.11 standard which has been extended numerous times to include more features, MIMO support and higher bandwidth. As the n extension is becoming the default in current mobile phones and access points, 802.11ac carries forward with even more supported antennas and even higher bandwidth (see table1) exceeding Bluetooth and NFC by orders of magnitude. With 802.11ac WLAN will transition completely to the 5 GHz band away from 2.4 GHz which has become somewhat crowded: Bluetooth, ZigBee and many other wireless technologies use it. Of all presented technologies WLAN has the highest possible range for consumer equipment in the realm of hundreds of meters. It also supports connection encryption out of the box. [10]

2.2 Operating modes

According to the NFC Forum there exist three basic modes of operation for NFC Forum devices: reader/writer mode, peer-to-peer mode and card emulation mode. In reader/writer mode a device is able to read or write NFC tags as specified by the NFC Forum. These include smart posters or tags with embedded text, URLs or signatures. This mode is conform with ISO/IEC 14443 and the FeliCa standard.

Peer-to-peer mode allows devices to exchange small chunks of data with each other, examples would be setup parameters for Bluetooth or Wi-Fi connections or virtual business cards. This behavior is newly specified by ISO/IEC standard 18092.

At last there is a card emulation mode which gives NFC devices the capability to emulate traditional, contactless, read-only RFID smart cards. This allows NFC devices to integrate with existing legacy RFID infrastructure, like ticketing systems in public transport, without any modifications of the legacy system.

2.3 NFC Data Exchange Format

To store and exchange information the NFC Forum has specified the NFC Data Exchange Format (NDEF, [11]). It is a binary message format allowing data exchange between NFC Forum devices or between a NFC Forum device and one of four NFC Forum tag types (type 1 through 4, [12]).

[2] `http://www.bluetooth.com`

doi: 10.2312/NET-2013-02-1_05

There exist NFC Forum Well-known Types, specified according to the NFC Forum Record Type Definition (RTD) specification [15]. These are simple URNs using the namespace identifier "nfc", prefixing "wkt" for well-known types or "ext" for self-defined types. A sample URN would be "urn:nfc:wkt:Sp" for the NFC Forum Smart Poster record type or "urn:nfc:ext:example.com:foo" for a self-defined type. The Smart Poster RTD [18] is based on the Text RTD [16] and URI RTD [17] in combination with actions that may trigger the launch of a web browser or sending of a SMS message. The Signature RTD [19] defines additional signature fields and suitable algorithms to allow verification of the authenticity and integrity of records inside a NDEF message, but employs no restrictions on the use of a certification architecture nor requires one at all. There also exists a Generic Control RTD [20], it is, however, deprecated and will be removed on or after August 9, 2012.

A NDEF message itself can include an unlimited amount of application-defined payloads in so called records. Each message is started with a record flagged as *MB* (Message Begin) and ends with a record flagged as *ME* (Message End). NDEF messages can be nested by including them inside a record of an existing NDEF message.

NDEF Message				
R_1, MB=1	...	R_i	...	R_n, ME=1

Figure 1: Example of an NDEF message with multiple records [11].

In addition to several flags providing support for chunking payloads over records (CF) or signaling very short records (SR), each record specifies the length of its payload as well as its type and an optional payload identifier which may be used to establish links between payloads in other records (see figure 2). Payload types may be formatted as NFC Forum specified Record Type Definitions as mentioned above, any MIME type as specified by RFC 2046 [13], URIs [14] and various others. When encoded in a NDEF message the namespace identifier and prefix of a Well-known type are dropped and represented by appropriate Type Name Format (TNF) field values in the record header. However, all types are understood as mere guidelines for overlying applications on how to parse payloads.

3. APPLICATIONS

According to the NFC Forum [1] NFC technology is or may be used in the areas of

- Access control,
- Consumer electronics,
- Healthcare,
- Information collection and exchange,
- Loyalty and coupons,
- Payments and
- Transport.

7	6	5	4	3	2	1	0
MB	ME	CF	SR	IL	TNF		
Type length							
Payload length 3							
Payload length 2							
Payload length 1							
Payload length 0							
ID length							
Type							
ID							
Payload							

Figure 2: NDEF Record layout. IL = ID length is present flag [11].

In general NFC is predestined to become a widely deployed technology as it requires only simple active hardware and can use cheap, easily mountable integrated circuit tags. Consumers will have NFC functionality embedded in their regular phones and setup and usage is as simple as bringing devices into close proximity of another. Applications on top may add value ranging from automation to payments.

As applications are endless we will focus on a few interesting areas: passive NFC tags (3.1), sharing between devices (3.2), financial services (3.3) - which will probably have the biggest (monetary) impact - as well as identification documents like tickets (3.4).

3.1 Tags

Storing pieces of information inside small, flexible circuitry embedded into stickers, little plastic shells or paper is one of the main visible use cases for NFC. These tags may be read by a NFC device, supporting the NFC Forum specified reader/writer mode, close to them and may contain simple text or URLs up to complex application specific data like configuration instructions. The NFC Forum specifies four different tag types, simply named NFC Forum Type 1 Tag through 4. Type 1 and 2 are based on the ISO/IEC 14443A contactless card standard [3] and are read- and re-writable. Type 1 tags store at minimum 96 bytes, Type 2 tags at minimum 48 bytes. Both at most 2 kbyte. Type 3 is basically the JIS X 6319-4 [5] contactless smart card standard used by FeliCa from Sony [4] which includes an additional read-only mode with available memory varying up to a maximum of 1 MByte while allowing multiple services on one card. Similarly Type 4 tags support multiple services, read-only mode and vary in memory size up to 32 KByte per service. However, they are based on the ISO/IEC 14443A/B standard.

Sony was the first company to offer its NFC tags, called Xperia SmartTags [21], as a consumer oriented accessory for their Android smartphones. Internet and some brick and mortar stores start to catch on to the trend and are offering tags with more memory in form factors ranging from stickers, classic card sizes and wrist bands to Sony

doi: 10.2312/NET-2013-02-1_05

SmartTags-like plastic badges[3]. Most of them make use of NXPs MIFARE chip series which are compatible with ISO/IEC 14443 [3], read- and writable by any NFC Forum device in reader/writer mode. Common use cases include mode or profile switching of a phone on touching a tag. For example putting your phone near a tag in your car may start the navigation app and enable Bluetooth pairing to the infotainment system. To write and react to the tags Sony provides a companion app, however there exist multiple free alternatives which support a wider range of tag types and commands[4][5].

3.2 Sharing

As the rise of Web 2.0 technology has shown social integration like sharing content between people is highly popular. Using NFC Forum devices in peer-to-peer mode enables a dead simple way of sharing content. Contact information, website URLs or small files are quickly passed on by a simple, intuitive action: touching devices together. For this particular purpose the NFC Forum defined a Simple NDEF Exchange Protocol (SNEP) [22] which uses the connection-oriented mode of the transport protocol Logical Link Control Protocol (LLCP) providing sequenced and guaranteed data delivery. It is also defined by the NFC Forum [23]. SNEP is a versioned request-response protocol between a client and a server. A client may send SNEP request messages to store (Put) or retrieve (Get) NDEF messages from a server over a LLCP data link connection.

Notably Android Beam [24], introduced by Google with Android 4.0 (Ice Cream Sandwich) in late 2011, is a peer-to-peer data exchange protocol for Android devices based on SNEP. It also supports Androids own, older NDEF Push protocol as a fallback. The sending user has to run the Android application he wants to share data from in the foreground, the receiving device needs to be unlocked and on its home screen. On touching the two devices together the sending device will show a "Touch to Beam" confirmation button allowing to transmit the desired information. Android Beam can be used by any application implementing its API. For example the built-in *People* app shares contacts, the *Browser* app shares URLs. Upon completion, the appropriate application will automatically be launched on the receiving device to handle the shared data.

To share bigger files like high-resolution photos or videos NFCs highest data rate of 424 kbit/s is barely sufficient. To work around this limitation, with Android 4.1, Google introduced establishing a separate, faster transport connection like Bluetooth via Android Beam taking over data transfers from NFC after successful connection[6]. The NFC Forum happens to have a specification of exactly this process titled Connection Handover [25], initially released in 2008, revised in 2010. In addition to a "Negotiated Handover" like Android Beam uses, it defines a "Static Handover" where parameters are stored and read off a passive tag.

Looking further back in time, Nokia was the first company to integrate NFC into a consumer purchasable phone in January 2007: the Nokia 6131. They showcased interaction with smart posters and sending images over to a digital picture frame[7]. In June 2011 they introduced their N9 smartphone [26] with the MeeGo operating system featuring tight integration with NFC in concert with a set of NFC-enabled wireless speakers (Nokia Play 360 [27]). By touching the phone to the speakers a Bluetooth connection would automatically be established and music would play on the speakers.

Meanwhile, in addition to Android and MeeGo, NFC peer-to-peer capability has been integrated into Microsoft's Windows Phone operating system as well as directly into Microsoft's new touch-focused tablet and desktop operating system Windows 8 [28]. In any case the focus has been on friction-less sharing between devices, on occasion handing over to a high-throughput connection when necessary to support larger data transfers.

3.3 Financial services

One of the biggest areas of interest, at least from an industries perspective, is financial services: paying with your phone acting as a digital wallet. As of 2012 there are three approaches on paying with NFC: you carry a NFC enabled banking or credit card, your phone includes a Secure SIM which handles encryption and authentication over the NFC interface, or you install an application which handles all payment processing.

There are currently two competing major application-based payment solutions: Google Wallet [29] and ISIS [30]. So far these are only available in the United States of America. Both require the installation of an application on your phone, then credit or debit cards have to be linked to their services to make them available for checking out within the app. On checkout the app is simply openend, unlocked with a PIN, and the phone placed on the merchants terminal to complete the transaction. Google Wallet tries to monetize their free service with Google Offers, displaying deals at the current shopping location. It works with any Mastercard PayPass [31] or Google Wallet enabled terminal with Mastercard claiming hundreds of thousands of supported locations world-wide. Non-partner credit cards may still be linked, but all transactions are handled through a virtual Mastercard. ISIS on the other hand is announced to support any NFC-enabled device of the three partners at&t, T-Mobile USA and Verizon launching in summer of 2012 and later. In contrast to Google Wallet, ISIS will only support some major cards directly, forcing users to rely on a prepaid card option, which can be charged in advance. However, ISIS is being backed by mobile phone providers which may come in as an advantage as Google Wallet is currently simply blocked on all Verizon phones while ISIS is readied for release [32]. ISIS is scheduled for a limited

[3]XDA developers, Where to Buy NFC Tags `http://forum.xda-developers.com/showthread.php?t=1662367`
[4]NFC Task Launcher `https://play.google.com/store/apps/details?id=com.jwsoft.nfcactionlauncher`
[5]NFC TagWriter by NXP `https://play.google.com/store/apps/details?id=com.nxp.nfc.tagwriter`
[6]Android 4.1 APIs `https://developer.android.com/about/versions/android-4.1.html#Connectivity`
[7]NFC in action `http://www.nearfield.org/2007/01/video-of-6131-nfc-phone-in-use`

doi: 10.2312/NET-2013-02-1_05

release on October 22nd 2012 [33].

Notably, as of September 2012 Mastercard has released an Android and BlackBerry SDK for their contactless payments product PayPass to enable any NFC-equipped device to pay through their service stepping into direct competition with Wallet and ISIS[8].

On the bright side, the competing standards for payment systems may soon be on the verge of getting unified or at least made compatible as the Electronic Transactions Association - a global trade association representing more than 500 companies including Google, ISIS, Visa and Mastercard - announced the launch of a Mobile Payments Committee on August 9, 2012 [34] including all four major US mobile carriers. At first, members promised to hold monthly meetings to update each other on their activities to slowly expand cooperation with the final goal of achieving industry wide solutions at some point.

Meanwhile as of April 2012, the Deutsche Kreditwirtschaft, a union of all major banks in Germany, has rolled out "girogo" [35]. A huge one-year pilot project covering the region of Hannover plus the cities of Braunschweig and Wolfsburg featuring NFC-enabled banking cards, the "girocard". It allows payments of up to 20 Euros with a swipe over a terminal without any PIN or signature. On the downside, the card has to be charged in advance with as much as 200 Euros via an ATM or a merchant checkout terminal secured by PIN. Single transactions will also not be visible on bank statements. The project has support from a big German grocery store, a gas company and various small retail chains totaling in a few hundred locations so far [36].

At last Microsoft announced [37] to integrate their payment solution in the upcoming Windows Phone 8 operating system. Their "Wallet" will require the phone to carry a Secure SIM element for paying, which is a conventional SIM packing additional functionality for encryption and authentication. It is handed out by the wireless carriers themselves. When transactions are triggered via the phones NFC interface further communication with a terminal will happen directly with the Secure SIM element avoiding the potentially less secure application environment of the phones operating system. "Wallet" will also have support for coupons, similar to Google Wallet, and store virtual boarding passes.

On a broader scale, the GSM Association announced its "Pay-Buy Mobile" initiative as early as 2007 [38] for embedding payment solutions inside SIM cards. Following years of trials, May 2010 saw the roll out of NFC services in Nice, France, for information access, public transport ticketing, coupons, loyalty programs and contactless payments in cooperation with major banks. More cities are planned. Similar undertakings made NFC services available in South Korea, Turkey, The United Kingdom and Tanzania. Mobile provider Orange is in the process of deploying NFC technology to various European countries. 2012 will see further projects by telecom providers KPN, T-Mobile and Vodafone in cooperation with banks in the Netherlands and as previously mentioned the release of ISIS in the US [39]. Several network operators like Deutsche Telekom, Orange and Telenor have promised through GSMA to launch NFC services throughout the world in 2012 [40].

[8] `http://www.mastercard.com/mobile/mobile-paypass.html`

3.4 Ticketing and Identification

Ticketing in public transport or sporting events, access control to restricted areas and embedding in identity documents are further applications for NFC technology. Public transport companies are already using or at least experimenting, access control and identity documents are mainly still promoted by the NFC Forum [41]. In both areas various RFID solutions, like Transport for London's Oyster[9], have been in use for several years using chips like NXPs MIFARE Classic smart card [42]. Biometric contactless readable passports are used in multiple countries around the world including the United States and Germany [43]. Moving public transport systems like Oyster over to NFC provides the freedom to freely exchange cards with customers phones. Due to security requirements, long standardization processes and small renewal cycles this will likely not happen for passports or other federal identity documents anytime soon. Similar concerns and missing standards hinder health care applications, like a virtual patient file linked in a personal NFC-enabled device which could provide relevant health data in a critical situation to speed up and reduce errors during care.

As for public transport, Deutsche Bahn offers their Touch&Travel program in Germany since November 2011. By scanning so called Touchpoints passengers can determine start and end of a journey and be automatically charged for the traveled distance. Showing the ticket to train personnel is realized via the passengers NFC phone interface in combination with an installable app [44]. Transport of London already trialed NFC phone replacements for its Oyster cards with high customer satisfaction, but to this date is waiting for Secure SIM based solutions to match the speed of the traditional Oyster, MIFARE Classic powered, smart card [45].

4. RISKS, ATTACK VECTORS AND EXPLOITS

As with many wireless communication technologies NFC is not invulnerable, despite its short range. Basic connections as specified by ISO/IEC 18092 using the NFCIP-1 protocol [2] are unencrypted and there are no checks for authenticity. Applications on top of NFC are expected to handle encryption and authentication by themselves. Notably, for authentication the NFC Forum already provides a Signature RTD specification [19] to embed signatures in NDEF messages.

4.1 Eavesdropping

As a wireless technology NFC is especially prone to eavesdropping. Despite connections occurring at a range of about four centimeters attackers might still exploit special circumstances and use specialized hardware to listen in on a connection. Similar attacks already exist for RFID contactless cards, in particular those using ISO/IEC 14443

[9] `https://oyster.tfl.gov.uk`

doi: 10.2312/NET-2013-02-1_05

implementations which NFC is also supporting [46]. When using active mode, so both devices are taking turns in generating their own fields, eavesdropping distances of up to 10m may be possible. In passive mode, where the target responds by modulating the initiators field, the range of attack drops significantly to around 1m [47].

4.2 Denial of Service

The simplest form of attack prohibits the use of the device or disturbs communication. As each device reacts to an incoming signal in some form, may it be a user interface requiring interaction, an attacker might spam the device with empty tag signals to make it unusable. The only solution is the inclusion of an off switch to disable NFC altogether [48]. Otherwise, communication might be prohibited by disturbing the data flow through transmitting at NFC frequencies with the correct timing. This would result in scrambled signals the receiving device is unable to decode. The attack just requires fitting hardware and sufficient knowledge of the used modulation and coding. However, NFC devices may easily detect such active corruptions as they require significantly more power on the attackers field [47].

4.3 Data modification

To actually modify data more thought has to be put in how the signals are modulated and coded so data still appears valid to the receiving device. Attacks occur on the bit level, switching single 1s to 0s and vice versa. When using the modified Miller coding only certain bits may be flipped. However, for almost all modes and data rates Manchester coding is used (see 2), allowing to modify any bit of the communication [47]. Prevention involves checking for third-party field influences and stopping communication on detecting any (which NFC devices do by default as specified by NFCIP-1), or using an encrypted channel on a higher layer. There exists an ISO/IEC standard describing NFC-SEC to already provide security on the data link layer complementing application layer security [49].

4.4 Relay attack

A relay attack is executed by sitting in the middle of two communicating parties and simply "relaying" requests and responses effectively making oneself invisible to either party (see figure 3). Relay attacks exist already for RFID systems and have been perfected to work with regular NFC phones by just installing specific pieces of software. An attacker would require two phones to act as proxies connected to each other with a high-speed link such as Bluetooth. One proxy device interfaces with the NFC token or device of a victim functioning as a proxy-reader. It forwards all messages over the high-speed link to the second proxy device imitating a NFC token to interface with the actual NFC reader, acting as proxy-token. Relaying even allows circumventing dynamic authentication on newer NFC card models as using the relay link introduces only small delays still accepted by current card readers. This attack concerns any ISO/IEC 14443 implementing contactless system, many, not NFC, of which are widely in use like the previously mentioned NXP MIFARE products. Possible countermeasures include aborting the communication if round trip times or the location of the pairing device are not as expected. For this purpose it has been suggested to tap into positioning via GPS or cellular networks as present in modern smartphones [50].

Figure 3: Schematic of a relay attack using a contactless smartcard, two NFC equipped phones and a reader terminal.

4.5 Others

Unlike technical attacks there still remains the risk of simply losing the NFC-enabled banking card or phone, with the phone being unencrypted or only secured via a weak PIN opening up abuse by third parties. Authentication should be handled by a separate factor to prevent any of those issues. Phishing by replacing original tags or readers with malicious units can be avoided by signing the content of exchanged messages [48], for example using the Signature RTD as specified by the NFC Forum [19]. Man-in-the-Middle attacks are practically impossible as either initiator or target are able to detect additional fields by a third party as mentioned before [47].

4.6 Implementation vulnerabilities

Attack surfaces do not only exist within the technology but also in implementations and attached services. Vulnerabilities in software handling or parsing NFC messages may open up access to sensitive data or a whole device. An example exploit was demonstrated by Charlie Miller at the Black Hat 2012 conference [51] which can take control of a Nokia N9 running MeeGo or a Galaxy Nexus running the Android 4.0 mobile operating system. Both operating systems accept incoming data beamed from the attackers device or tag and then automatically open a malicious web page or a modified file in a vulnerable app. Protective measures include modifying apps to let the user always confirm the triggered action before executing it.

5. CONCLUSION

Near Field Communication is on its way to become an essential part of our daily lives. It provides simple means of making information available by using NFC tags embeddable everywhere one can think, readable with the NFC-enabled mobile device in your pocket. Sharing of text, websites or setting up Bluetooth or Wi-Fi connections for large file transfers or advanced interaction is as far away as touching two devices. Waving your credit card or phone at checkout makes paying a new experience. However, certain risks are associated with mostly unencrypted data transfers, security holes in still young software libraries or by actual theft. Communication in the range of centimeters may appear to make exploitation difficult, but it is very much possible. As of 2015 every second smartphone may be equipped with the technology [52], standards for better security and interoperability will likely emerge. Near Field Communication is going mainstream, for the better or worse.

doi: 10.2312/NET-2013-02-1_05

6. REFERENCES

[1] *NFC Forum*, http://www.nfc-forum.org

[2] ISO/IEC 18092 *Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1)*, 2004

[3] ISO/IEC 14443 *Identification cards – Contactless integrated circuit cards – Proximity cards*, 2000

[4] *Sony Global - FeliCa*, http://www.sony.net/Products/felica

[5] JIS X 6319-4 *Specification of implementation for integrated circuit(s) cards – Part 4: High speed proximity cards*, 2005

[6] *NFC Forum Unveils Technology Architecture And Announces Initial Specifications And Mandatory Tag Format Support* http://www.nfc-forum.org/news/pr/view?item_key=0b210bbd23e9c1a07cb3d975e6317d1d650ed51f, June 5, 2006

[7] Jin-Shyan Lee, Yu-Wei Su, Chung-Chou Shen: *A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi*, Industrial Electronics Society, 2007

[8] Ari Juels: *RFID Security and Privacy: A Research Survey*, IEEE Journal on Selected Areas in Communications, Volume 24 Issue 2, 381-394, February, 2006

[9] http://www.zigbee.org

[10] Erina Ferro, Francesco Potorti: *Bluetooth and Wi-Fi Wireless Protocols: A Survey and a Comparison*, IEEE Wireless Communications, February, 2005

[11] *NFC Data Exchange Format (NDEF) Technical Specification*, NDEF 1.0, NFC Forum, July 24, 2006

[12] *NFC Forum Tag Type Technical Specifications*, http://www.nfc-forum.org/specs/spec_list/#tagtypes

[13] N. Freed, N. Borenstein: *RFC 2046 - Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, 1996

[14] T. Berners-Lee, R. Fielding, L. Masinter: *Uniform Resource Identifier (URI): Generic Syntax*, RFC 3986, January, 2005

[15] *NFC Record Type Definition (RTD) Technical Specification*, RTD 1.0, NFC Forum, July 24, 2006

[16] *Text Record Type Definition Technical Specification*, RTD-Text 1.0, NFC Forum, July 24, 2006

[17] *URI Record Type Definition Technical Specification*, RTD-URI 1.0, NFC Forum, July 24, 2006

[18] *Smart Poster Record Type Definition Technical Specification*, SPR 1.1, NFC Forum, July 24, 2006

[19] *Signature Record Type Definition Technical Specification*, SIGNATURE 1.0, NFC Forum, November 18, 2010

[20] *Generic Control Record Type Definition Technical Specification*, GC-RTD 1.0, NFC Forum, March 7, 2008

[21] *Sony Xperia SmartTags* http://www.sonymobile.com/gb/products/accessories/xperia-smarttags/

[22] *Simple NDEF Exchange Protocol Technical Specification*, SNEP 1.0, NFC Forum, August 31, 2011

[23] *Logical Link Control Protocol Technical Specification*, LLCP 1.1, NFC Forum, June 6, 2011

[24] *Beaming NDEF Messages to Other Devices*, Android Developers, http://developer.android.com/guide/topics/connectivity/nfc/nfc.html#p2p

[25] *Connection Handover Technical Specification*, Connection Handover 1.2, NFC Forum, July 7, 2010

[26] Heidi Lemmetyinen: *Introducing the Nokia N9: all it takes is a swipe!*, Conversations by Nokia, June 21, 2011, http://conversations.nokia.com/2011/06/21/introducing-the-nokia-n9-all-it-takes-is-a-swipe/

[27] Adam Fraser: *Nokia steps it up a gear, with new accessories*, Conversations by Nokia, June 21, 2011, http://conversations.nokia.com/2011/06/21/nokia-steps-it-up-a-gear-with-new-accessories/

[28] *Microsoft MSDN Windows.Networking.Proximity namespace*, http://msdn.microsoft.com/en-us/library/windows/apps/windows.networking.proximity

[29] *Google Wallet*, http://www.google.com/wallet/

[30] *ISIS*, http://www.paywithisis.com/

[31] *Mastercard PayPass*, http://www.paypass.com/

[32] Amir Efrati, Anton Troianovski: *War Over the Digital Wallet*, The Wall Street Journal, December 7, 2011, http://online.wsj.com/article/SB10001424052970204770404577081610232043208.html

[33] Nathan Ingraham: *ISIS confirms October 22nd launch in Salt Lake City and Austin*, The Verge, October 17, 2012, http://www.theverge.com/2012/10/17/3516778/isis-october-22nd-launch-salt-lake-city-austin

[34] *ETA Launches Committee To Guide Emerging Mobile Payments Industry*, August 9, 2012, http://www.electran.org/docs/releases/2012/ETALaunchesMobilePaymentsCommittee.pdf

[35] *Deutsche Kreditwirtschaft führt neues Markenzeichen girogo für das kontaktlose Bezahlen ein*, January 11, 2012, http://www.die-deutsche-kreditwirtschaft.de/dk/pressemitteilungen/volltext/backpid/26/article/deutsche-kreditwirtschaft-fuehrt-neues-markenzeichen-girogo-fuer-das-kontaktlose-bezahlen-ein.html

[36] *girogo*, http://girogo.de

[37] *Announcing Windows Phone 8*, June 20, 2012, http://windowsteamblog.com/windows_phone/b/windowsphone/archive/2012/06/20/announcing-windows-phone-8.aspx

[38] *GSM Association Aims For Global Point Of Sale Purchases by Mobile Phone*, GSMA, February 13, 2007, http://www.gsma.com/newsroom/gsm-association-aims-for-global-point-of-sale-purchases-by-mobile-phone/

[39] *NFC - Market by Market*, GSMA, http://www.gsma.com/mobilenfc/the-gsma-and-mobile-nfc/nfc-market-by-market/

[40] *World's Leading Mobile Operators Announce Commitment to NFC Technology*, GSMA, http://www.gsma.com/newsroom/worlds-leading-mobile-operators-announce-commitment-to-nfc-technology/

[41] *NFC as Technology Enabler*, http://www.nfc-

forum.org/aboutnfc/tech_enabler/

[42] Flavio D. Garcia, Gerhard de Koning Gans, Ruben Muijrers, Peter van Rossum, Roel Verdult, Ronny Wichers Schreur, Bart Jacobs: *Dismantling MIFARE Classic*, ESORICS 2008, LNCS 5283, pp. 97-114, 2008

[43] Jaap-Henk Hoepman, Engelbert Hubbers, Bart Jacobs, Martijn Oostdijk, Ronny Wichers Schreur: *Crossing Borders: Security and Privacy Issues of the European e-Passport*, arXiv, January 25, 2008, http://arxiv.org/abs/0801.3930v1

[44] *Touch&Travel*, http://www.touchandtravel.de

[45] Dan Balaban: *London Oyster Card Chief: NFC Not Ready for Fast-Paced Fare Payment*, NFC Times, May 30, 2012, http://nfctimes.com/news/london-oyster-card-chief-nfc-not-ready-fast-paced-fare-payment

[46] G.P. Hancke: *Practical eavesdropping and skimming attacks on high-frequency RFID tokens*, Journal of Computer Security, Volume 19 Issue 2, 259-288, 2011

[47] Ernst Haselsteiner and Philips Semiconductors: *Security in Near Field Communication (NFC)*, Workshop on RFID Security RFIDSec, 2006

[48] Gerald Madlmayr, Josef Langer, Christian Kantner, Josef Scharinger: *NFC Devices: Security and Privacy*, Availability, Reliability and Security - IEEEARES, 642-647, 2008

[49] ISO/IEC 13157-1 *Information technology – Telecommunications and information exchange between systems – NFC Security – Part 1: NFC-SEC NFCIP-1 security services and protocol*, 2010, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=53430

[50] Lishoy Francis, Gerhard Hancke, Keith Mayes, Konstantinos Markantonakis: *Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones*, 2011

[51] Charlie Miller: *Don't stand so close to me: an analysis of the NFC attack surface*, July 25, 2012, http://www.blackhat.com/usa/bh-us-12-briefings.html#Miller

[52] Peter Schüler: *Nichtöffentlicher Nahverkehr - Die Nahfunktechnik NFC in Smartphones und Chipkarten*, c't 2012, Heft 14, 140-143, 2012, http://www.heise.de/ct/inhalt/2012/14/140/

doi: 10.2312/NET-2013-02-1_05

Content-Centric Networking

Fabian Oehlmann
Supervisor: Heiko Niedermayer
Seminar Future Internet WS2012/2013
Chair for Network Architectures and Services
Faculty for Informatics, Technische Universität München
Email: oehlmann@in.tum.de

ABSTRACT

This paper introduces to the alternative internet architecture of information-centric networking as opposed to the common host-to-host oriented design approach. Since the rise of computer networking until the internet becoming an incremental part of people's everyday life, it has undergone severe changes in usage scenarios. Users are interested in content and services and not in locations where to find them. In order to reflect those better in the underlying architecture of the internet, *Content-Centric Networking* (CCN) tries to loosen content and services from addresses in the network of machines. Although being challenged by a new set of difficulties, the approach yields some desirable system-inherent consequences that alone make the research effort worthwhile. Despite of its different mindset, a development of the common internet architecture towards this direction does not seem unrealistic.

Keywords

content-centric, information-centric, internet architecture, future internet

1. INTRODUCTION

The internet architecture today is still strongly driven by the ideas and necessities of the early days of computing and communication. A central design principle in computer networks is the end-to-end argument. As described in [12], it formulates the necessity of a complete function implementation only at the endpoints of a communication system. It is a pledge for a well-defined boundary between the communication subsystem and the rest by a strict interface. From this derives the simplicity, flexibility, and eventually also the universality, which made the internet as successful as it is.

Evidently, the internet has changed since its origins, making it reasonable to reconsider even its founding principles [2]. In the beginning it was an internetwork between few institutions, connected for purposes of resource sharing, where there would be many users on one machine. Today it is a continuously growing, ubiquitous phenomenon. Not only a selected group of people, but everyone interacts with it over many devices. An astonishing plethora of new usage scenarios for the internet developed, introducing new stakeholders and interests, such as those of governments, ISPs or enterprises whose business model is built on the internet. Grave is the loss of trust towards the rest of the internet, as there is today a wide range of security issues that have to be handled, when using it. For this and other purposes there have already been introduced mechanisms, which are not consistent with the end-to-end argument, such as firewalls, traffic filters or network address translators.

Having identified the distribution of content, such as videos, music, and news as the main use of the internet today, several propositions suggest to further adapt the architecture accordingly [5, 15]. The idea is to replace locations, where information can be found, with the information itself as the central element of networking, thus breaking out of the mindset of the end-to-end argument. This is termed as content-centric or information-oriented networking. Much like peer-to-peer mechanisms and content distribution networks facilitate this in the current architecture, new propositions try to manifest this on lower levels. This comes with certain inherent advantages, but also poses new challenges to be solved.

The next Section gives an overview of the ideas of those alternative architectures with up- and downsides. Afterwards Section 3 explains the mechanisms and practicability of content-centric networking (CCN) as one incarnation of the proposals in closer detail. Section 4 then gives an overview of the related proposals. The work is concluded in Section 5.

2. INFORMATION-ORIENTATION

The concept of information-orientation internetworking is to replace the location of content (or services) in the architecture, i.e. IP addresses, with the content itself. Thus everything used for packet forwarding is some sort of identifier for the contained data. The motivation is to provide a network participant means to declare interest in a piece of information to the network, which then replies with the content, preferably in an efficient manner.

With this concept in mind a couple of suggestions [13, 10, 9, 14] developed more or less closely related to the publish/subscribe paradigm described in [6]. The common principle is to have a producing side (publishers) and a consuming side (subscribers). The latter subscribe to content, thus posing their interest, while the publishers somehow put content into the system, that is then automatically forwarded to the subscribers. Furthermore, the pub/sub paradigm demands a decoupling of those two actions in time, space and synchronization, which means that they can be performed independently from one another. Effectively, this realizes a pull model for the receiving side, which decides on what

doi: 10.2312/NET-2013-02-1_06

content to receive and nothing else, opposed to the packet forwarding mechanism in the current internet. As a consequence the network is given the task of finding and delivering the content over just providing a host-to-host connection. The counterpart models of the current internet architecture and those of an information-oriented one are depicted in Table 1.

Original internet	Information-oriented
Sender	Content producer (publisher)
Receiver	Content consumer (subscriber)
Sender-based control	Receiver-based control
Client/Server communications	Publish/Subscribe sender and receiver uncoupled
Host-to-host	Service access/Information retrieval
Topology/Domain	Information scope
Unicast	Unified uni-, multi- and anycast
Explicit destination	Implicit destination
End-to-end	End-to-data
Host name (look-up oriented)	Data/Content name ("search" activity)
Secure channels, host authentication	Integrity and trust derived from the data

Table 1: The opposing concepts of the original and an information-oriented internet architecture [5]

2.1 Aimed benefits

Mostly following these principles the current designs promise a list of advantages over location-based communication. Depending on the design the advantages may be more or less inherent.

The first advantage of the concept is an easy way to implement communication primitives, such as multicast or anycast, while unicast is just a special case of the regular mechanism. Content that is subscribed by many end-points is just replicated according to the subscriptions, making this an inherent feature.

Secondly, mobility support is achieved by the lack of a need for receivers in a new network to obtain an address or identity. A sender can simply republish the offered content.

The freedom of location eases the use of many interfaces (multihoming), since just any connection can be used to publish or subscribe content, without further ado. Related to the feature of anycast support, this additionally improves persistence. Content can be published using one name, at many locations, over many channels.

Leaving out the destination in a packet header, makes it useful for many possible destinations. This renders the possibility of content caching at every intermediate node in a delivery tree. Similar to the inherent multicast, content that is demanded more than once, can be cached and replicated. This leads to a higher availability by reducing latency and increasing reliability through effective mirroring at nearby nodes.

The on-demand characteristic of the communication allows stateless connections. Interruptions in data transfer can therefore easily be tolerated, within the boundaries of application specific demands.

Especially from the pull-based messaging characteristic derives an increased security. DDoS, spoofing or spam could be practically eliminated, assuming a working authentication of the content, since the architecture only delivers packets a consumer has signaled interest in.

2.2 Challenges

The challenges for an implementation are manyfold and depend on a clear definition of the interface and therefore the tasks, that are to be implemented by the communication system. Over all the communication should in the end be more efficient in some way, making the development worthwhile.

The question of routing is the most crucial one, as it is the core of the architecture. The approaches taken are either inspired by existing routing protocols or build up on distributed hash tables. The difference to IP routing lies in the fact, that the names unlike IP addresses can not give a location hint as it is the case with IP prefixes and subnetting. This relates to the concept of introducing interface independent identities on top of the IP layer as in [3]. Furthermore, the mechanism has to be able to cope with the vast size of today's and the future's internet. Therefore a scalable solution is essential. Especially considering the fact, that there is more content than hosts in the network, an increased size of routing tables as to taken care of.

Additionally, a challenge is to meet requirements, which are driven by the internet's stakeholders like ISPs. Inter-domain routing policies have to be possible to be reflected in the new architecture as well. Preferable is also the possibility of an incremental deployment of the architecture, as well as the compatibility to existing machinery. A sudden replacement of the infrastructure would be out of scope, as costs would easily surpass the achievable advantages.

Lastly, ensuring security of the architecture has to be taken care of with highest priority. There could be risks unthought of in new routing designs, as opposed to the old ones, such as sibyl attacks in DHTs. Most importantly in a content-based architecture is guaranteeing authentication and integrity of the delivered content. If the proposed architecture can not give means to a consumer to be provided with the content she is asking for, its goal is not met. Thus, a strong protection against the compromisation of content is of vital importance.

3. THE CONTENT-CENTRIC NETWORKING DESIGN

This chapter outlines the general functioning mechanisms proposed by Jacobsen et al. in [9]. A first implementation of the concept is under development in the CCNx project[1].

The basic idea of content-centric networking is, as stated before, making content the central element in networking operations. Accordingly the architecture differs from the

[1]http://www.ccnx.org

doi: 10.2312/NET-2013-02-1_06

common ISO/OSI layer model as depicted in Figure 1. Instead of IP, depicting source and destination of a packet, it is chunks of explicitly named content, that form the waist of the hourglass of the stack. By this, the content becomes the universal agreement between every network participant. It is decoupled from the end-hosts of the connection. Thus, on the new layer the source of the desired content is of no relevance anymore. Additionally, there is also a strategy layer and a security layer introduced below and on top of the content layer.

The introduced layers can be on top of the IP layer, but might as well be directly on top of the MAC layer or anything else delivering packets. The so called faces are managed by the strategy layer, which contains policies to when it is appropriate to use the correct one. Notably content is largely independent of the connection type it is served upon. Therefore it natively supports multihoming as well as it is tolerant towards disruptions of the connectivity. The authors chose the word faces instead of interfaces to point-out the possibility to have application processes work as those as well besides network interfaces. The security layer ensures the authenticity and integrity of the content, further described in Section 3.5.

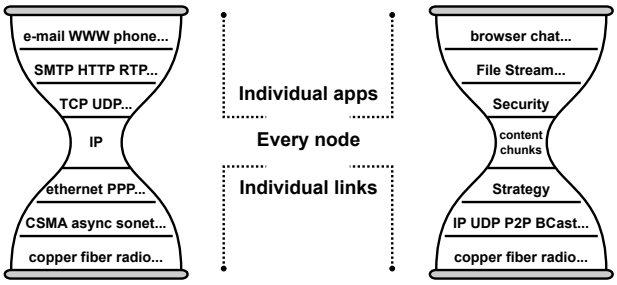


Figure 1: Changing the bottleneck of the communication stack to chunks of content [9]

3.1 Packet types

The CCN model knows two packet types. First is the *Interest* packet as depicted in Figure 2. By sending these over the outgoing interfaces, a node announces its demand for the content named by the packet. It is simply broadcasted on the available interfaces in hope to get the according data returned by the mechanisms of CCN. Naturally, the packet contains the name of the desired content. Additionally, it is accompanied with selection information, such as the scope within the network where the data should come from or certain filter information. Finally, it contains a nonce, used to detect duplicate interests.

In response to an incoming interest, *Data* packets are used as can be seen in Figure 3. The data packets are said to "satisfy" interest in that they maintain a one-to-one relation,

Content Name
Selector
Nonce

Figure 2: Interest packet

Content Name
Signature
Signed Info
Data

Figure 3: Data packet

where data consumes interest. This rule maintains a flow balance at each hop and prevents congestion in the middle of a connection path. The names in CCN are hierarchical with the consequence, that data only serves an interest, if its name-prefix matches the name of the interest. Apart from the name and arbitrary binary data, the packet also contains a digital signature of some cryptographic digest of the packet, as well as signed info. The last mentioned field gives additional information on the packet such as the publisher's ID, where to locate the key to check the signature or a timestamp. By these means of verfication it is to be ensured that the packet is authenticating and identifying itself and does not need legitimacy by the channel it was transferred over. This is further explained in Section 3.5. The design specifically allows interests for data that does not exist yet. To serve these "active names" publishers may generate content dynamically to meet the demands of the modern internet.

3.2 Forwarding engine model

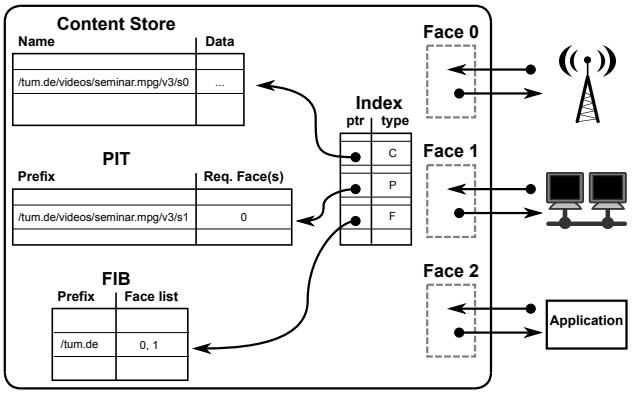


Figure 4: CCN node model [9]

A CCN node (see Figure 4) performs similar operations compared to those of a regular IP router. Packets that were delivered on a face undergo a longest-prefix matching on their name field and are processed according to the information stored in the three main data structures maintained by every node:

The content store (buffer memory) serves as a cache of content. Since content is self-authenticating and self-identifying each packet might be useful to any potential participant in the nearby network. The ability to serve content directly instead of generating further lookups minimizes overall bandwidth usage and latency. Compared to an IP router the most-recently-used replacement strategy of the buffer is replaced by a least-recently or least-frequently used strategy to increase the probability of a cache hit.

doi: 10.2312/NET-2013-02-1_06

The pending interest table (PIT) keeps track of Interests issued on the nodes faces. Hereby it does not matter whether the interest originates from the node itself or is one forwarded from another node. Interests are the only packets routed in CCN towards the source(s) of the content. As soon as an interest reaches a content source the PIT serves as a mark in the data's trail towards its requester(s). The authors compare this mechanism with "bread crumbs" consumed by the data on their way. By these means the data satisfies the interest and the entry in the PIT is removed. Unsatisfied interests time-out eventually and have to be reissued by the consumer.

The forwarding information base (FIB) acts like the routing table in a common IP router. It stores the information on which faces interests are to be forwarded upstream towards the source(s) of the content in question. The design hereby allows multiple entries that may be queried in parallel, because forwarding is not restricted to a spanning tree.

The processing of interest packets is managed according to the above order of the data structures. If the desired content is to be found in the cache, the node serves the request directly, thus, satisfying the interest. In case of a cache miss, the PIT is checked for an exact-match entry, and the arrival face is added to its list of requesting faces. This means an interest in that data has already been forwarded upstream. When it gets satisfied, the node copies the data on its way downstream. Lastly, it is matched for an entry in the FIB. The requesting face is removed from the FIB entry, an interest is sent out on the remaining faces of the entry and a PIT entry is stored. If the content name does not match any of the above the interest is discarded, as the node neither can satisfy it nor does it have the knowledge where to forward it. Duplicate interests might arrive at different faces, which is prevented by detecting and discarding ones that carry an already known nonce.

Data packets are processed in a similar way. First the name is matched with the content store. If it is already present on the node, it is a duplicate and can be discarded. Second the name is matched with the PIT. If there is no match, the data can be discarded as well, since there was no demand for it. In case of a match the data may be validated (see Section 3.5) and afterwards cached in the content store. Consequently it is forwarded on all faces in the list of the PIT entry.

3.3 Naming and transport

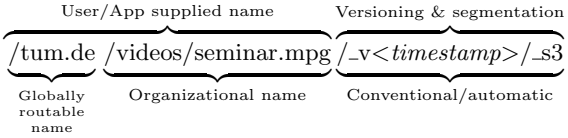


Figure 5: Human readable example data name

The naming scheme of CCN is designed to have a mostly human readable form for purposes of usability. At the same time it tries to comply with demands from routing, i.e. efficient routing capabilities on names, and is inspired by functioning mechanisms of TCP. Therefore, as can be seen in Figure 5, names are hierarchically structured and put together from several components. For notational convenience the human readable representation is chosen by the designers with / signs between the components as in common URIs. Notably, the binary representation differs from just the string representation of this. It is divided into a name provided by the user or an application, which consists of a globally routable name and the name of the content within the organizational structure of its origin. Secondly, the tail of the name is supposed to be a standardized naming convention reflecting the version of a certain content and its segmentation. The scheme allows to have a total order on the content that can be reflected in a content tree as in Figure 6. Accessing content means the traversal of the tree. An incremental feature obtained by imposing a total order is the ability to address content relative to known information. E.g. for accessing a video a consumer can issue an interest for '/tum.de/videos/seminar.mpg' and the selector primitive 'RightmostChild'. From having a certain chunk of video the standardized naming convention allows expressing interests in the following chunks by adding an offset according to the segmentation rules. The similarity to TCP's window size is hereby given with the number of the amount of next interests sent in parallel. The selective acknowledgement is given by the one-to-one serving mechanism of the chunks.

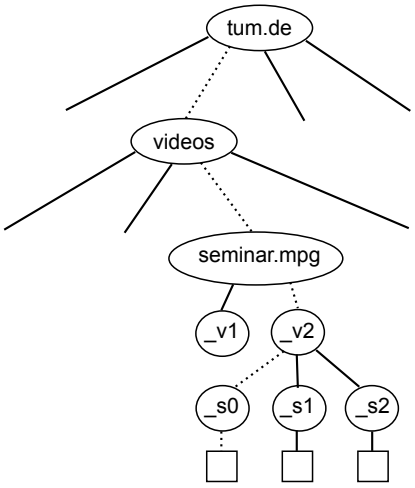


Figure 6: Name tree traversal [9]

3.4 Intra-domain Routing

Intra-domain routing in CCN is claimed to be very much compatible to routing schemes known in IP networks. Both rely on longest-prefix matching lookups, to hierarchically aggregate a more detailed connection information in the process of getting closer to the content. The similiarity is reflected in the close relation of the FIB and an IP routing table. The key difference is that the FIB has more than just one outgoing interface. The reason for this is the semantic difference of the entries. While an IP router can reach all hosts starting with a prefix via the stored interface, the CCN router only can reach some of the content with this prefix there. As a result the CCN router broadcasts interests to all of the faces on this entry to gather complete informa-

doi: 10.2312/NET-2013-02-1_06

tion. In an IP network this operation would lead to loops in the topology. In contrast, this is no problem in CCN, since neither interest nor data packets can loop. Thus, CCN is free from maintaining a spanning tree within the topology. For instance, if two nodes in a topology announce the same prefix, in IP this would mean that via both every content with such a prefix can be reached. Therefore, the forwarding mechanism has to pick the better one of them. In CCN, such two nodes would not both announce reachability to the whole content under this prefix, but maybe just a subset. Thus, both nodes have to be queried. This broadcast based mechanism raises questions of scalability [15].

Despite these differences, existing link-state routing protocols, namely IS-IS and OSPF, should be easily adaptable to the needs of CCN. The mentioned routing protocols basically handle two distinct tasks. The first is discovering local connectivities to adjacent nodes. The second is to announce available prefixes/resources throughout the network. By these means a CCN enabled node can distribute link-state-announcements containing the attached prefixes it can serve. The authors of CCN now propose that this is already possible with existing deployments of these protocols. Both of them announce connected resources using a general 'type label value' (TLV) scheme, which is capable of distributing CCN prefixes. Following the specification, nodes that do not support CCN-TLVs would simply ignore them. CCN enabled routers would directly add the distant CCN enabled router's IP address as a face. This way CCN routing could be implemented attached to that of IP. Additionally this can be done incrementally as not every router needs to be CCN enabled. CCN's interest broadcast mechanism would of course find better routes with an increased ratio of CCN enabled routers in the network.

3.5 Security

In the common internet architecture security of content is largely provided by authentication of the host from which content is delivered as well as securing the communications channel to prohibit man-in-the-middle attacks. The CCN architecture decouples content from its origin in a way that it does not matter by which means a consumer got hold of it. As a consequence authenticity and integrity of the content have to be possible to be checked just by the information provided alongside the data. In other words, the essential goals are to ensure that a certain piece of content was in fact published by the right source and that the content was not changed in any way. Thus, a digital signature mechanism has to be introduced. In [7, 9] the authors therefore demand publishers of content to maintain public and private key pairs to sign data packets. As given in Figure 3 data is transferred not only with its name, but also with the signature and some signed information. The approach is to have publishers not just sign the content, but both the content and its name together, in order to authenticate the linkage between them. The signed info does not need to be verifiable and may include the location of the public key or the public key itself. Thus, the consumer can verify whether the content in question was published with this name by a given key or not.

Eventually this mechanism does not provide trust in the key yet. As a solution a public key infrastructure is suggested. This can be easily realized by the CCN mechanisms themselves. Given a name tied to an identity, trust can be formed in the following way: An authority simply signs the name and public key of the identity as its content. This equals to an identity certificate issued by the authority. Additionally, this functioning allows fine grained certification of just a subtree of a given namespace.

The CCN design does not provide architectural instruments to protect content or enforce access control. The suggested solution is to rely on ordinary encryption of data, since CCN handles any binary sequence of data the same way.

3.6 Usability

Having described the basic architectural concepts, remains the question of the actual usability of the proposal in the real world.

3.6.1 Implementation

On the CCN project website one can find source code for an early stage implementation of the new architecture. At its core it consists of the ccnd linux userspace deamon, enabling basic CCN functionality for research purposes. Alongside to that an Android implementation, application libraries and several applications can be found. The latter prove the usability of the concept for the contexts of chatting, HTTP GET requests, media streaming and file transfer. Thus, it covers most of today's internet use cases.

As an example shall be given a closer insight on the most sophisticated application, given by the VoIP alternative VoCCN, as depicted in [8]. The application realizes phone calls between two parties using the Session Initiation Protocol SIP and RTP media streams. Used was a modified version of the Linphone VoIP client. Standard SIP connection establishment is done using a signaling path, which is an indirection infrastructure maintained by proxies, forwarding the connection details between the caller and the callee. By this a direct media path is built up between the two parties. VoCCN arguments to simplify this process by merging the signaling and the media path. A callee Bob simply announces to be the content source for the call, e.g. using the name '/tum.de/sip/bob/invite'. The caller Alice then sends out an interest for the SIP invite message with the same prefix, which is served by Bob. There he defines the connection details, i.e. the detailed name under which to route the mediastream. Thus, VoCCN makes use of the active names, described in Section 3.1, to serve unpublished content. This then sets up a simple bidirectional media-stream by the means of delivery depicted in the end of Section 3.3

3.6.2 Feasibility

The CCN design removes intelligence from the edges, the endpoints of connections, to the core of the network itself. This results in higher resource demands particularly considering the content caching functions, as well as an increased size of the routing tables. There is a general consent in that content caching poses high potential in increasing the efficiency of networking, by eliminating duplicate transmissions in the delivery tree. Yet, today's internet infrastructure is highly optimized on low levels to support packet forwarding mechanisms in host-to-host connections. Unfortunately,

doi: 10.2312/NET-2013-02-1_06

this means the underlying functions and data structures of the new proposals cannot simply be deployed on the existing infrastructure. A new hardware design for content routers is necessary. The question answered by [1] and [11] is, whether present-day technology is capable of processing the different workload. Naturally, due to the early stage of the research, those answers rely only on vague estimations regarding the dimensions of the requirements and are supported by simulation. Eventually, they point out that depending on the size of the implementation a deployment is not unrealistic. Up to the size of CDNs or ISPs the concept might very well be feasible. The scalability up to the size of the whole internet, although, is with current possibilities to be doubted.

3.6.3 Evaluation

Content-centric networking is a very radical approach to change the modern internet architecture. Yet, at its current stage it is a research effort, which demands a lot more testing and refining. Prototype deployments in real testbeds yet have to prove the correct functioning and scalability. Particularly the simplistic routing mechanism is a reason to doubt the practicability of CCN. Furthermore, the security aspect of the proposal is a crucial element. If the mechanism to provide authentication and integrity is corrupted, the architecture is defect in its substance. Delivery of the correct content could not be guaranteed anymore, potentially rendering the architecture useless. Another aspect are new threats rising from the mechanism itself, such as the possibility to render denial of service attacks by flooding interest packets, which can only be mitigated by heuristic countermeasures.

In summary, CCN is a yet very theoretic approach, displaying a certain elegance, which leads to a number of desirable advantages. However, it is unlikely to ever be deployed in its initial form. Nonetheless, it is a good research effort for the evolution of the internet. In what ways it will have an impact, will ultimately be decided by its economic benefits.

4. RELATED WORK

There are several proposals, which were designed to replace the current internet architecture. The one closest to the one of CCN is the Data-Oriented Network Architecture DONA [10]. It introduces a new network entity, the resolution handlers (RH), one for every autonomous system. Content is registered at those using flat, human-unreadable names, that are self-certifying. Publishers sign the data, that a consumer can verify with their public key, the signature and the name. Consumers issue find commands towards their RH, which locates the content. It is then delivered using a regular IP connection.

The internet indirection infrastructure (i3) [14], based on early experiences with distributed hash tables (DHT), similarly proposes to form an overlay network by a DHT. The DHT organizes nodes to take care of content identifiers. A node interested in such content places a trigger at this node. Publishers send packets for that identifier to that node, which then forwards the data to the addresses, who registered a trigger for the identifier. The data connection is done again using IP.

In the Publish-Subscribe Internet Routing Paradigm (PSIRP) project[2] [13] follows to implement a pure publish subscribe model as mentioned in Section 2. Much like DNS a directory structure is introduced mapping human-readable names on content-labels. The architecture would then rely on three Network modules. *Rendezvous*, which matches publications and subscriptions, *Topology*, which forms content-delivery-trees and *Forwarding* of labels. *Mediation* depicts the physical data transmission between nodes.

The Network of Information (NetInf) project[3] [4] also makes use of DHT for a name resolution system. While it also implements a publish-subscribe interface, its focus lies more on giving an abstract information model and naming framework.

5. CONCLUSION

The internet architecture as it evolved until today does not reflect the standard usage scenarios of the majority of its users anymore. Content-based networking seems to be a promising approach to simplify networking operations, reflecting today's requirements towards the internet in a more natural manner. A long list of potential enhancements could significantly increase the efficiency of the internet, while simplifying it at the same time.

CCN is a thoroughly designed approach towards the new paradigm. It is in an advanced status allowing ongoing research, which is necessary to reveal unthought of consequences and challenges. Nonetheless it shows possible flaws in respect to scalability when facing the internet's size. Moreover, by this example can be seen, on how many levels considerations have to be taken into account, when tackling the challenge of changing a huge system like the internet at its core.

Aside the technological consequences one may also not forget the surrounding implications the architectural change may have. Breaking up the end-to-end argument of [12] by shifting the computations from the end-hosts to the core of the network, is also a shift in which party has power over the network. As pointed out in [2], the end-to-end argument is a big reason for the internet's flexibility and base for innovation. A major redesign, thus, also has to be taken under careful consideration not only regarding its economic, but also its social benefits. The internet's stakeholders are numerous and represent opposing interests regarding its open nature. Thus, future developments of the internet should also be viewed cautiously from a non-technical perspective.

6. REFERENCES

[1] S. Arianfar, P. Nikander, and J. Ott. On content-centric router design and implications. In *Proceedings of the Re-Architecting the Internet Workshop*, ReARCH '10, pages 5:1–5:6, New York, NY, USA, 2010.

[2] M. S. Blumenthal and D. D. Clark. Rethinking the design of the internet: the end-to-end arguments vs. the brave new world. *ACM Trans. Internet Technol.*, 1(1):70–109, Aug. 2001.

[2]The PSIRP project has ended and is today followed by the PURSUIT project, http://www.fp7-pursuit.eu

[3]http://www.netinf.org

doi: 10.2312/NET-2013-02-1_06

[3] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, and I. Stoica. Rofl: routing on flat labels. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, pages 363–374, New York, NY, USA, 2006.

[4] C. Dannewitz. NetInf: An Information-Centric Design for the Future Internet. In *Proceedings of the 3rd GI/ITG KuVS Workshop on The Future Internet*, Munich, Germany, 2009.

[5] C. Esteve, F. L. Verdi, and M. F. Magalhães. Towards a new generation of information-oriented internetworking architectures. In *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, pages 65:1–65:6, New York, NY, USA, 2008.

[6] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, June 2003.

[7] V. Jacobson and D. K. Smetters. Securing network content. Technical report, PARC, 2009.

[8] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard. Voccn: voice-over content-centric networks. In *Proceedings of the 2009 workshop on Re-architecting the internet*, ReArch '09, pages 1–6, New York, NY, USA, 2009.

[9] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, CoNEXT '09, pages 1–12, New York, NY, USA, 2009.

[10] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '07, pages 181–192, New York, NY, USA, 2007.

[11] D. Perino and M. Varvello. A reality check for content centric networking. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, ICN '11, pages 44–49, New York, NY, USA, 2011.

[12] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-end arguments in system design. *ACM Trans. Comput. Syst.*, 2(4):277–288, Nov. 1984.

[13] M. Särelä, T. Rinta-aho, and S. Tarkoma. RTFM: Publish/subscribe internetworking architecture. ICT Mobile Summit, 2008.

[14] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. In *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '02, pages 73–86, New York, NY, USA, 2002.

[15] D. Trossen, M. Sarela, and K. Sollins. Arguments for an information-centric internetworking architecture. *SIGCOMM Comput. Commun. Rev.*, 40(2):26–33, Apr. 2010.

doi: 10.2312/NET-2013-02-1_06

Alternatives to X.509

Michael Gielesberger
Supervisor: Ralph Holz
Future Internet Seminar - Winter Term 2012/2013
Chair for Network Architectures and Services
Faculty of Computer Science, Technische Universität München
Email: gielesbe@in.tum.de

ABSTRACT

Secure connections on the Internet are crucial to many applications. Nowadays infrastructure for securing connections on the Internet heavily relies on the X.509 public-key infrastructure (PKI), which is maintained by Certificate Authorities (CAs). Recent attacks on CAs showed the fragility of the current X.509 system.
In order to alleviate or entirely fix problems with the X.509 system, different techniques and systems have been proposed. The proposed systems can be categorized into 4 different concepts: DNS-based, Pinning-based, Notary-based, and Transparency-based approaches. This paper gives an overview over these concepts and outlines their advantages and disadvantages. It also covers some of the best-known representatives of the concepts.
Sovereign Keys, a Transparency-based approach, is discussed in detail.

Keywords

X.509, PKI, Sovereign Keys, Certificate Transparency, Convergence, Perspectives

1. INTRODUCTION

In nowaday's widespread use of the Internet, problems have arisen which were unforeseen when the core infrastructure was developed between the 60s and the 80s. In the small networks of those days attacks on the confidentiality or authenticity of connections were at most a theoretical issue. Today's usage of the Internet includes applications with high security requirements, such as e-commerce applications. Because of the ever growing need for secure communication, different protocols were developed to allow secure connections over the Internet. In particular the Secure Socket Layer/Transport Layer Security protocol suite (SSL/TLS) is widely used nowadays to guarantee authentication, data confidentiality, and data integrity. SSL/TLS is used in conjunction with many different protocols but the best-known domain is securing the Hypertext Transfer Protocol (HTTP) with HTTPS.
Like most protocols which are used at the moment to secure connections over the Internet, SSL/TLS is based on asymmetric cryptography, at least for key exchange purposes. This leads to a necessity of distributing public keys. As an approach in which every host exchanges its public key with every other host does not scale (there would be $O(n^2)$ needed key transfers) a hierarchical approach for key distribution was introduced. The X.509 public-key infrastructure (PKI) offers a solution where only a few root public keys need to be transferred to the hosts. The root keys belong

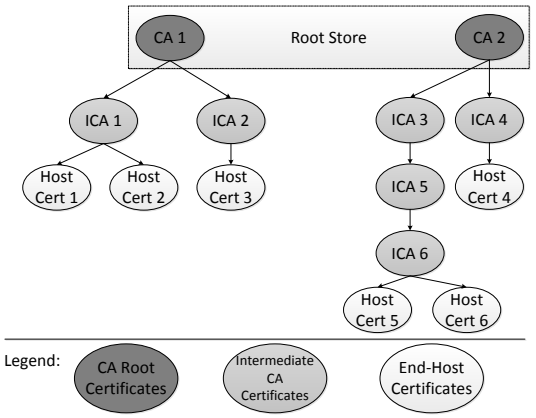


Figure 1: The X.509 hierarchy

to Certificate Authorities (CAs). The CAs issue certificates to other entities, in which they assert the binding of the entity's public key to their identity, in case of the WWW a domain name. CAs may also delegate signing privileges to other entities, which constitute an Intermediate Certificate Authority. X.509 builds up a certificate hierarchy with the CAs' certificates as roots, some optional intermediate certificates, and the hosts' certificates as leaves. The hierarchy is depicted in Figure 1. This results in trust chains. Clients, e.g. web browsers, have a so called root store, where they store root certificates which they trust. With these trusted root certificates, clients are able to verify a host's certificate by going through the chain of certificates until they reach one which is in their root store. Commonly clients ship with many root certificates: A common web browser trusts over 600 entities which are able to issue certificates [2].
The biggest problem with this approach is that every CA is equally able to issue certificates for every entity. When an attacker makes a CA assert the binding of the attacker's public key with another entity's identity, he gains a rogue certificate, which cannot be distinguished from real ones and thus the whole security system is rendered useless. This makes the whole system as weak as the weakest CA in the hierarchy. Recent attacks on the existing PKI have shown the fragility of the current system. With attacks on DigiNotar [14] and Comodo [3] certificates for high profile sites such as `addons.mozilla.org` or `login.facebook.com` have been falsely issued and used. Such rogue certificates allow attack-

doi: 10.2312/NET-2013-02-1_07

ers to perform Man-in-the-middle (MitM) attacks. There are also concerns that some state-run CAs, from which certificates are in root stores of clients, may issue rogue certificates for surveillance purposes [17]. Today's approach to deal with such attacks is to revoke compromised certificates, but the detection of a compromise may take several days or may never be detected [14]. And the techniques for revocation either do not scale well in the case of Certificate Revocation Lists (CRLs) or are flawed in the case of the Online Certificate Status Protocol (OCSP) [12], [9].
Many extensions or alternatives to the current X.509 system have been proposed to alleviate and fix the current problems before an attacker is able to use rogue certificates for a longer period of time. There are different approaches, which can be categorized into different categories: DNS-based (Domain Name System based) approaches try to store additional security information directly with the DNS records of the domains in order to prevent rogue certificates. Pinning-based approaches anchor certificates by comparing the certificates of a new connection with either already seen or out-of-band distributed certificates. Notary-based approaches try to expose rogue certificates by viewing connections to a secured server from different locations on the Internet and determine if there is a common consensus on the seen certificates. Transparency-based approaches force the publication of certificates or another validating key in a way that makes it possible for a client to compare and verify that the certificate it sees is the one the domain owner intended to use. Domain owners are also able to check all existing certificates or keys needed for validating a certificate for their domains with that approach.
The Sovereign Keys (*SK*) project by the EFF [6] belongs to the Transparency-based approaches. It has some unique features, which make it interesting to explore more in-depth.

The remainder of this paper is structured as follows: in section 2 the four mentioned approaches are introduced in general and some of the most important systems based on those approaches are presented. The following section 3 presents and analyses the unique features and techniques of the *SK* project more in depth. The final section 4 concludes the findings of this paper.

2. ALTERNATIVE CONCEPTS

In order to fix flaws in the current X.509 PKI system, several different techniques have been proposed. This section gives a brief overview of the most promising and common techniques. The best-known systems, which implement the technique, are named and some interesting specifics are explained for those systems.

2.1 DNS-based

DNS-based techniques take advantage of the fact that for every connection to a domain on the web a DNS-server has to be contacted in the first place to translate the domain name to the corresponding IP. As this connection needs to be established anyway (except for cached or locally assigned domain names), DNS servers seem to be a good place to store additional security relevant data. To be able to securely transfer data with the DNS the use of DNNSEC, which is still not wide-spread, is a prerequisite for most of these systems. With the use of DNS servers it is unnecessary to build up a whole new infrastructure to distribute certificates. The general idea is to publish additional security information as a new record type along with the other records in the DNS. One can differentiate between client side checked records and CA checked concepts:
For the client side concept a hash or a fingerprint of a public key is stored in the DNS. The hash or fingerprint is then compared with the actual certificate a client gets when connecting to the secured server. With this there is another channel which needs to be compromised by an attacker in order to conduct a MitM attack. The domain owner may decide if the single hash of the certificate, which belongs to the server's domain should be directly stated or the hash of a certificate further up on the certificate chain. By using the latter it is possible to state the fingerprints of the public key of the CA which issued the domain's certificate. This allows a domain owner to use different certificates but still narrow down which CA may issue certificates for the domain. A connecting client would check if there is a certificate along the certificate chain which matches the hash/fingerprint stated in the DNS and just allow the connection if this check succeeds. If an attacker is able to compromise a CA it is just possible to generate certificates for domains which have the fingerprint of the compromised CA stated in their DNS security record. When the attacker generates a rogue certificate for a domain, which states another, not compromised CA's public key fingerprint in the DNS security record, the certificate will be detected as invalid.
For CA side checked concepts the information is just checked by a CA which is asked to issue a certificate. If an attacker tries to get a certificate issued from a CA, the CA should look up the security information in the DNS for that domain and check if it is allowed to issue a certificate for it. Such a system does not offer any protection against CAs which just do not check those entries as there is no concept how to force CAs to check the entries.
DNS-based systems help against common attacks but have the disadvantage that they offer no protection against state-run attacks or surveillance activities. As states normally control their country-code top-level domain (ccTLD) they can completely change the DNS entries for domains in their ccTLD. E.g. Libya controls the `.ly` ccTLD, and thus can gain control over the domain `bit.ly`.
The best-known system, which belongs to the category of client side checked DNS-based concepts is DANE (The DNS-Based Authentication of Named Entities) [7].

2.2 Pinning-based

The general idea of pinning is to compare the certificate a client sees during a current connection attempt with other already seen certificates or hashes of certificates. From the first connection on it should additionally be required to use a secured connection, as otherwise simple attacks are possible, where a secured connection just gets redirected to an unsecured one. The easiest implementation is to store the certificates when a client first connects to a SSL/TLS secured server and compare the certificates seen in subsequent connections with the already seen certificate. There are different systems which use simple methods like this, e.g. Certificate Patrol [16]. Another approach to pin certificates is to send an additional header when connecting over SSL/TLS to a server, which contains a hash of a public key in the certificate chain of the real certificate. This pin has a lifetime, which is specified by the server the client got the pin from.

doi: 10.2312/NET-2013-02-1_07

When a client connects to a server, it compares the certificates in the trust chain with the hash of the pin and just allows the connection if the hash matches some certificate in the chain. By using the hash of a certificate which is not the certificate of the domain itself, but a certificate which is further up the certificate chain, a server is able to just allow certificates which belong to a specific intermediate CA or a specific root CA.
A specific problem of this concept is the distribution of the pins. As pre-sharing pins for all SSL/TLS secured domains is not practical, the pins are normally sent to the client when it first connects to the server. Thus an attacker which controls the connection at the moment a client first connects to a secured service will still be able to compromise the connection.
Google's Chrome browser uses public-key pinning by already shipping some hard-coded pins for domains of Google and the anonymization service Tor.

2.3 Notary-based

The concept of a Notary-based approach is to look at a secured server from many different viewpoints on the network.

2.3.1 Concept

A way to find out about MitM attacks is to compare the connection to a specific server with connections to the same server from the perspective of different clients at different positions on the Internet. Those other clients, which also try to connect to the secured server, are called notaries. This includes not just notaries which are just connected to other Internet Service Providers (ISPs), but also in other countries, or even on other continents. The reasoning is the following: when many notaries on very different positions on the network see the same certificate, it is very likely the certificate the domain owner really intended to use. As attackers are normally not able to compromise the whole network, but just small parts of it, a rogue certificate used for a MitM attack in just a fraction of the whole network will lead to different views on the compromised domain and can thus be detected by the notary concept. With the Notary-based concept the whole checking of certificates boils down to finding out if there is a consensus on the certificate of a server throughout the whole network.
There are some problems with this concept: there are some sites which have different certificates for each single server. The best-known site employing such a scheme is Citibank [16]. Sites like that will raise alerts with a Notary-based approach as there will be no consensus on the seen certificates.
Another deficiency is that the Notary-based approaches cannot detect global attacks. An attack which would compromise every connection to a server may be possible when there is just one gateway connecting the server to the global network. When this single connection point to the network is controlled by an attacker, the attacker is able to compromise every connection to the server. This is a general problem with the concept as it does not check if the used certificate is actually the one the domain owner uses but just if everyone sees the same certificate [18].
Notary-based systems do not use an already established channel such as DNS. This is why they need to open a new channel, a side-channel, for their communication with the notary servers. As with every technique which uses side-channels there is also the practical problem with captive portals: normally when connecting through a pay-to-use portal to the Internet, e.g. at a hotel, all requests get directed to the login site and all services besides DNS are blocked until the user paid for the usage and the connection is opened up for normal use. As the notary service runs through a side-channel, at the time of connecting to the login site a user cannot validate this secured connection with a notary system.
Unlike some other approaches, Notary-based approaches are able to deal with self-signed certificates as it does not matter for the notaries if there is a path to a (trusted) CA or not.

2.3.2 Perspectives

One of the first and best-known systems using the Notary-based approach, is the Perspectives project by Carnegie Mellon University [19]. Perspectives was already published in 2008 and came with a proof of concept in the form of a Firefox Add-On.
The notary servers are the most important entities of Perspectives. Notary servers are decentrally organized and generally independent of each other. Notaries either already have a key history of the requested service and are able to answer with that key history or if the client requests the key history for a service the first time, the domain is added to the list of monitored services. Either way, the notary will get the certificate from the domain's service and answer with that key. A host running the Perspectives client asks a number of notaries when connecting to a secured service and asks them for the key history they have seen for the server. The list of monitored servers is periodically probed by each notary and all witnessed certificates are stored with a timestamp. This results in a key history for all monitored services. The key history includes all previously witnessed certificates, and with the timestamps, how long they have been seen. The connection to the notary servers is secured by public key cryptography where the notary servers' public keys need to be distributed to the Perspectives clients by an out-of-band mechanism.
As a notary server knows about all connection attempts of a client to secured services, the Perspectives system raises some privacy issues. This is one of the issues of Perspectives, which Convergence tries to fix.

2.3.3 Convergence

Moxie Marlinspike presented Convergence at the BlackHat USA conference in 2011 [13]. Convergence uses the same idea as the Perspectives project, but it tries to tackle a few deficiencies of the Perspectives system. In order to fix the privacy issues, Convergence locally caches certificates which have already been validated by the Convergence system for some time. This brings the advantage that notary servers will not find out about every connection attempt to a secure service. But on the other hand it introduces a time frame, where a previously validated certificate may get compromised, but still be trusted by the Convergence client. Another mechanism for the improvement of a client's privacy is the introduction of notary bouncers. A client may ask a notary to proxy its request to another notary server. This is an onion-style routing mechanism where the proxying notary knows which client requested a key history, but as it cannot read the request message it does not know for which service. The notary server, which gets contacted by the proxy notary, knows for which service the request was for but does not know which client requested that information.

doi: 10.2312/NET-2013-02-1_07

However as soon as an attacker controls the proxying notary and the second notary the client's requests are known to the attacker.
In contrast to Perspectives, Convergence is designed to be highly extensible. While Perspectives uses the current X.509 PKI to validate certificates, Convergence is able run notaries with different ways of validating certificates. It is possible to run notary servers which validate the certificates with just a certain set of trusted CAs or even with a completely different system, e.g. DANE or the OpenPGP Web of Trust. Convergence is the only system presented in this paper which views itself as a replacement rather than an extension for the existing X.509 PKI.

2.4 Transparency-based

The recent attacks on the X.509 PKI took a few days to be detected. This is due to the fact that there is no mechanism to look up all certificates issued for a certain domain. A domain owner may never find out about the existence of rogue certificates for the domain which were generated by some attacker. Such rogue certificates are just detected when a suspicious user finds out about such a certificate, or when a CA finds out that it got compromised and issued rogue certificates. The concept of Transparency-based approaches is to fix the existing X.509 PKI by publishing issued certificates or keys needed to validate a certificate in such a way that it is possible to find all valid certificates or keys needed to validate certificates for a certain domain. Transparency-based systems are a very promising approach, as they focus on the idea of clients being able to check if the certificate they see is really the one the service operator intends to be seen.

2.4.1 Concept

The general concept involves an infrastructure in which a certificate must be published before it becomes valid. One could describe the Transparency-based approaches as public log based approaches. This relatively easy idea brings up technical problems, which need to be solved to be usable in real world environments. Public logs need not just serve the currently valid certificates or keys needed to validate certificates but need to be able to give a history of the used certificates or keys for a specific domain. Otherwise trivial attacks are feasible where an attacker adds a rogue certificate to the log for a short period of time and deletes it from the log when the attack is over. Because of that Transparency-based systems use append-only data structures which ensure that all certificates ever issued for a specific domain are stored in the log. The result is that domain owners and other interested parties are always able to find out about all valid certificates or keys needed to validate a certificate for their domains. If an attacker is somehow able to add a rogue certificate to the log the real domain owner is at least able to easily find out about it and revoke it.

2.4.2 Certificate Transparency

Certificate Transparency [11] is a system currently developed by Google. Since the original proposal some details have been altered and the current design paper version 2.1a was recently released [10]. With Certificate Transparency all certificates have to be registered with public log servers. The log servers save the domains' certificates in an append-only data structure. This is done using an ever-growing Merkle Tree which is a cryptographic primitive offering the possibility to easily and continuously check the append-only property of the data structure and to ensure integrity.
When a domain owner registers a certificate with a log server, the hash of the certificate together with a timestamp of the registration is signed by the log server and sent back to the domain owner. This signature is the so called Signed Certificate Timestamp (SCT). The secured server has to send the SCT together with the actual certificate at every SSL/TLS connection attempt to the connecting client. For clients the SCT works as a proof that the certificate is registered with a log server. In addition to validating the certificate itself, like it is done in the current X.509 PKI system, the client checks if the SCT is correctly signed by a log server. Clients will get the log servers' public keys using out-of-bands mechanisms, like shipping them hard-coded in the client software. The system ensures that every certificate has to be registered with the log servers, because otherwise clients will not accept the connection. Domain owners should monitor the certificates which are registered for their domains and make sure that the rogue certificates get revoked.
To ensure that the log servers are not compromised and work honestly there are monitors and auditors. Monitors check that logged certificates are promptly visible on the log and that they are on the log legitimately. Domain owners should either take this role or use some service, which does it on their behalf. A domain owner is able to check if a new certificate gets added to the log after submission and to check if there are any certificates for the domain not created by himself.
Auditors can check that partial data on the logs is consistent with the current state of the log. E.g. clients can implement an auditor, which checks that all certificates they encounter are visible in the log. Both can be efficiently implemented because of attributes of the Merkle Tree.
Certificate Transparency has the advantage that it does not depend on side-channels, like many other systems do. The secured servers themselves send the needed data to the clients and they are able to verify the signature of a log server offline. The usage of side-channels is limited to additional checks, which are not necessary for every client for the system to work. As of today the proposal does not include any mechanisms for revocation of keys. However, the initial proposal brought up the idea of having another log for revocations together with a proof of non-existence similar to DNSSEC's NSEC records.
Although Certificate Transparency is meant to be deployed over a period of time, it is meant to be required for every domain certificate in the end. The authors suggest that a complete transition may be enforced by not allowing the issuance of any new certificates without their publication by the Certificate Transparency system after a certain date [8]. As certificates automatically expire and need to be renewed, this enforcement would gradually reach every certificate.

3. SOVEREIGN KEYS

Sovereign Keys is a project by the Electronic Frontier Foundation (EFF) [6] and is categorized as a Transparency-based system. Unlike Certificate Transparency, the *SK* system does not publish the certificates themselves but another type of key, the Sovereign Key. It still fulfills the classic definition of Transparency-based approaches as no certificate is considered valid without being cross-signed with a Sovereign Key

doi: 10.2312/NET-2013-02-1_07

and the Sovereign Keys themselves are indeed published. Note that there is no working prototype yet. This paper takes information from the proposal, which was first published in November 2011 and last updated in June 2012 [4] and the specifications writeup from July 2012 [20]. Those are still subject to change and a final implementation may differ in specifics. However, source code is already publicly available in the project's git repository.

3.1 The basic design

The major problem with the existing X.509 system is that CAs are able to issue valid certificates for a domain without needing any information or consent of the real domain owner. With *SK*, certificates are not valid without being cross-signed with a special private key of the domain owner, the so called Sovereign Key. This private key is supposed to

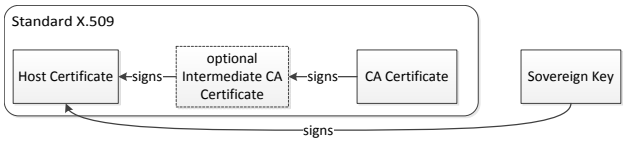


Figure 2: Relation between Certificates and Sovereign Keys

be secret and just known to the real domain owner. Because of that no one besides the real domain owner is able to carry out the final step of making a certificate valid.
In order for clients to be able to check the cross-signatures, the corresponding public keys need to be made available in some form. This is where the transparency aspect of the *SK* design comes in: all public keys of domain owners' Sovereign Keys need to be published in so called Timelines in order to be valid.
Timelines offer an append-only data structure for storing all Sovereign Keys ever used for domains. The integrity of the timelines can be cryptographically checked. When registering a domain name as a Sovereign Key in a Timeline, the Timeline server checks if the registration is authorized (see section 3.2.1). This relatively easy concept becomes quite complex when one addresses some real-world problems like the possibility of compromised Sovereign Keys or scalability.

3.2 Architecture

SK sees itself as an extension to the current X.509 PKI and keeps that infrastructure at least for bootstrapping. The system however introduces some new entities: Timeline servers, Mirrors and clients.

3.2.1 Timelines

The Timelines are the backbone of the *SK* system. They hold mappings between a (domain) name and Sovereign Keys. The concept envisages a semi-centralized data structure with the use of 10 to 30 Timeline servers [4]. The reason to use not just a single centralized server is to provide a Śdiversity of jurisdictions, operational philosophies and security implementations, and ideally to provide these properties even if some of the servers are compromised/disabledŠ [4]. Timeline servers may be identified with their Timeline Address (TADDR). The TADDR consists of the domain name and a port, by which they can be reached, and

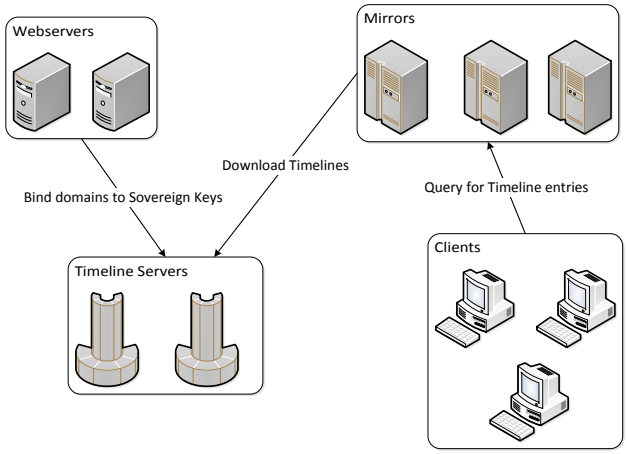


Figure 3: The Sovereign Keys architecture

a public key, which is used for secure connections to them using SSL/TLS. Besides that they have a unique Timeline ID (TID), which should only be used for a specific Timeline server. Even when a Timeline server gets shut down and removed from the system, the TID must never be assigned to another Timeline server. Each entry in the Timeline includes the Timeline server's TID so that it is possible to distinguish from which Timeline server an entry came from when they are merged into a single dataset.

Registration of a Sovereign Key: When a domain owner registers a Sovereign Key, a master-write is done to one Timeline server. However entries should be written to many Timeline servers, preferably in different geographic regions, to be safe in case a Timeline fails. The registrations to additional Timeline servers is done by reference where just a hash of the entry at the other Timeline server and that server's TID is saved. However when a Timeline server just adds a reference it has to cache a copy of the entry from the other Timeline server. This is done in order to have copies in case of a failure of the referenced Timeline server but to still have a globally unique entry.
In order to register or bind a domain name to a Sovereign Key, a domain owner sends a `Bind` message to a Timeline server. A `Bind` message includes the domain name for which the entry should be created. To prove the authorization, a domain owner must present a claim proof for the domain. This may be a CA-signed certificate from the current X.509 system or a key published with DNSSEC using DANE [7]. This proof of claim includes the whole certificate chain from the host certificate to the trusted CAs certificate, in order to be able to reproduce the trust decision later. The claim proof is also appended to the `Bind` entry in the Timeline. In order to not be fooled by compromised, but revoked keys Timeline servers have to verify the OCSP status of the certificates.

Cryptographic features of the Timelines: Timeline servers are not able to create fake entries as the data structure of the Timelines allows a cryptographic verification of the entries. All entries have an incrementing serial number which is local to every Timeline server and a monotonically non-decreasing timestamp. Every Timeline server has to

doi: 10.2312/NET-2013-02-1_07

sign every entry in its Timeline with its private key. All the listed properties may be checked easily:

- If there is an entry which is not correctly signed with the Timeline's private key, every checking party will notice and the entry will not be processed.
- If two correctly signed entries with the same serial number exist, a checking party can notice and has signed evidence of the violation of the incrementing serial number property.
- If there is a signed entry with a lower serial number but a higher timestamp, a checking party has signed evidence of the violation of the monotonicity of the timestamps.
- If a correctly signed entry with non-self-consistent data exists (like a non-verifiable certificate chain for a CA signed domain claim proof), a checking party has signed evidence of that inconsistency.

Because there is always signed proof for dishonest entries it is easy to verify the claim of a Timeline's misbehaviour. The design offers a mechanism to distribute the information of Timelines' misbehaviour and a Timeline server which distributes false data will be blacklisted in the system.

3.2.2 Mirrors

To have a scalable system and in order to decrease privacy issues, clients do not directly query the Timeline servers themselves, but Mirrors.
Mirrors store the whole Timelines data structure, collected from all trusted Timeline servers. For an easy initial deployment process Mirrors may be bootstrapped by downloading a set of Timeline entries up to a specific timestamp. As this might be a huge dataset (up to a few hundred gigabytes [5]) it is envisaged to be done with bittorrent or sending a hard disk with the dataset on it via conventional parcel services (disk-over-snailmail). The dataset is then updated to the current state by querying each trusted Timeline server for all entries since the last serial number in the dataset, that the mirror already holds. With this process, the mirrors gather the complete Timelines data structure, which is semi-centrally spread across the Timeline servers. Using this data set, the Mirrors are also able and supposed to check if all Timeline servers are behaving correctly. In case of a misbehaviour the misbehaving Timeline server gets flagged as rogue and this information is then passed to clients and other mirrors. In the end, misbehaving servers get removed from the list of trusted Timeline servers.
To create a complete view of the semi-centralized Timeline data structure, mirrors combine all entries from the different Timeline servers to produce a complete view on the Timelines. This is the view a client gets when querying Timelines for a specific domain. This set of entries is first ordered by their timestamps and, for entries which have the same timestamp, by the entry's TID.
Clients are able to choose which Mirror they want to use, just like DNS servers. The EFF envisages that Mirrors will also be run by ISPs just like they run DNS servers today. This would decrease the risk of privacy concerns as the queries would not get out of the ISP's network and the ISP is supposed to know to which domains a client connects to anyway because of DNS lookups. In addition, the design introduces an onion-routing style proxy feature for mirrors, where a client can ask a mirror to proxy a query to another mirror. Although the mirror knows which domains a client connects to, no other party on the network is able to read the communication between the client and the mirror as this connection is secured with SSL/TLS.

3.2.3 Clients

When a secured server is uses *SK*, the client connecting to it will bypass the X.509 certificate chain checking, as this has already been done by the Timeline server. The client will just need to get the Sovereign Key for that server from a Mirror and then check if the server's certificate has correctly been signed with the current Sovereign Key.
One thing to note about the cross-signature of the Sovereign Key is that the X.509 data format does not offer the ability to sign a certificate with more than one key. To be able to add the Sovereign Key signature to a host's certificate, which should already be signed by a CA, an extension to the X.509 data format is needed. However, clients which do not know about *SK* should not be affected because the extension is done in a way that legacy clients will ignore the *SK* part.

Trust towards Mirrors: When connecting to a secured server, clients query Mirrors for all Timeline entries regarding the server's domain name since the last entry that the client still has in its cache. Mirrors answer those requests with a list of all entries since the last entry of the client's cache together with a Timeline Freshness Message (TFM) of all involved Timeline servers. A TFM includes a timestamp, the highest serial number to that point in time, the serial number of the last CA update entry to that time, and a signature of the timeline server. All this is additionally signed by the mirror so that the mirror is liable for it. The TFMs are used to check the honesty of Mirrors. Like Mirrors check Timeline servers' integrity and honesty, clients check Mirrors: When a client gets an entry which has a higher serial number or a higher timestamp than stated in the TFM, it knows that the Mirror is misbehaving. When such a dishonest Mirror is found, the evidence of their misbehaviour will be reported and in the end those Mirrors will be flagged as bad and not used anymore.

Connection to a secured server: When a client connects to a SSL/TLS secured service it will query a Mirror for the Sovereign Key for that domain and service. The Mirror will answer with a timestamp-sorted list of entries for that specific domain. As the next step the client goes through the timestamp sorted list of entries it received from the Mirror. The first `Bind` entry is trusted, from there on every additional entry to the timeline must be signed with the bound Sovereign Key, unless a signed `Unbind` entry exists. If there is an `Unbind` entry, the chronological next valid `Rebind` entry will be trusted next (see section 3.4 for details). All correctly signed entries will be processed in this way and ultimately the client knows which Sovereign Key should be trusted for the connection to a specific service on a specific domain.
This Sovereign Key is then used to validate the domain's certificate by checking if it has been signed with the Sovereign

doi: 10.2312/NET-2013-02-1_07

Key. In order to offer more privacy and to be more robust against Mirror outages, clients can cache Sovereign Keys.

Caching: As Sovereign Keys are supposed to be valid for long periods of time, clients cache the keys locally. However, revocation checks will be done occasionally. Caching will also lead to a very small amount of data which needs to be transferred when the Timeline for a certain domain has already been cached as there are not be many updates to be expected.

3.3 Other features

SK offers some additional features. The domain owner may declare for each domain which services should use the *SK* system or if every (secured) service should use *SK*. By using this feature, a server is able to secure some services with *SK* but keep on using the standard X.509 system for some other services.
There is also a feature which allows domain owners to state an alternative route to a domain's service in case the normal route is blocked or unavailable. Such an alternative route would be tried when the normal route is not available or seems to be under attack. Some proposed alternative route modes are the use of proxies, VPNs or, especially promoted by the authors of *SK*, the anonymization service Tor. This allows an automated fallback, which may be useful e.g. in restrictive countries where the government tries to block access to certain domains.

3.4 Key management

The *SK* design already includes several solutions for key management use cases:

Revocation: Unlike Certificate Transparency, *SK* already has a concept to revoke compromised keys. In order to be able to rebind a previously unbound Sovereign Key, a domain owner may state the domain names of so called Rebinders. First, a compromised Sovereign Key needs to be unbound. This is done using an `Unbind` message, which needs to be signed with the Sovereign Key. An attacker which compromised a Sovereign Key could do that as well, but because of the Rebinder concept the attacker is not able to rebind a new Sovereign Key to it.
When one wants to bind a new Sovereign Key to the domain, this may just be done with a `Rebind` message, which needs to be signed with the Sovereign Key of one of the domain names previously stated as a Rebinder.

Change of domain owner: When a domain is transferred to another owner, the same mechanism as with revocation is used. First the current owner unbinds the current Sovereign Key and then the new owner will be able to rebind the new Sovereign Key through one of the Rebinders.

Expiry: To not block a domain by binding a Sovereign Key with a domain forever (see next item), Sovereign Keys expire and need to be renewed just like certificates in the X.509 system. One can never bind a Sovereign Key for longer than one holds ownership of the domain. This property is checked when binding a domain name to a Sovereign Key. However, it is possible to get domains for a hundred years, and thus it is possible to bind a Sovereign Key to a domain for that period of time.

Losing a Sovereign Key: One problem with the Sovereign Key concept is that if a domain owner loses the Sovereign Key, no changes to the entries are possible anymore. This may even lead to losing control over the domain until the Sovereign Key expires. The domain owner is not able to add any entries for the domain in the *SK* system anymore, as all entries must be signed with the now lost *SK*.
A Sovereign Key may be compared to physical property in this case: Once it is lost, it cannot be regenerated. Revocation of a lost key is impossible as the system requires an `Unbind` message for a revocation, which needs to be signed with the Sovereign Key. Thus Revocation is really just useful in the case of a compromised key.

3.5 Discussion

SK fixes some issues which are prevalent in other concepts or systems. The system is not affected by state-driven attacks on the DNS system, like DNS-based approaches are. When a government abuses its control over the DNS system and changes entries in there, the rightful domain owners are still in power over their Sovereign Key and the government would not be able to forge connections to the secured services running on that domain. In contrast to normal Pinning-based approaches, even the first connection of a client to a service running *SK* is secured. Also, *SK* is generally compatible with every X.509 PKI based protocol. In comparison to Notary-based systems, *SK*, just like every other Transparency-based system, does not just rely on what other hosts see on a network. Clients are able to check if the certificate they see is really signed by the rightful domain owner, which in the end is the only party able to decide if a certificate is the right one or not. Unlike Certificate Transparency, the *SK* project already has a concept for a revocation system.
However, there are some issues, which will take some effort to be fixed. Problems with captive portals should not be a problem for *SK* in general. But as *SK* needs a side-channel to check if a domain name is secured with a Sovereign Key, this will need a workaround: The authors propose to collect a list of known captive portal domains and then whitelist them [1]. This collection is proposed to be done with the help of a common web browser or a plugin for one. As the domains of the captive portals do not use *SK*, this does not affect the security. When a captive portal provider decides to incorporate *SK*, the captive portal will be removed from the whitelist.
Another issue are Denial-of-Service attacks on the Timelines data structure by writing a big amount of entries to it. This is proposed to be fixed with special rules for TLDs, which need to be paid for and rate-limiting for all domain spaces, which may be registered for free.
It will also be needed to take precautions that it is not possible to register a Sovereign Key for a domain not owned by the registering party. This may even render domains useless (see section 3.4). The authors propose to have a multi-staged process, which includes some explicit changes to the services run on the domain over a longer period of time [1].
However one issue is immanent to the *SK* system: a Sovereign Key may be seen as physical property. When it is lost it cannot be recovered nor revoked. This may even lead to loss of the domain until the Sovereign Key expires.

doi: 10.2312/NET-2013-02-1_07

4. CONCLUSION

The 2011 attacks on CAs alerted the Internet community that the current X.509 PKI is in a fragile state. There are many different approaches to fix the current system. As each system has its weaknesses, it may take the combination of some of those approaches to build a system which is able to fix all problems while still being deployable.

SK in its current proposed form is quite comprehensive and may as well be combined with DNS-based approaches. As there is no working prototype yet, it has to be seen if the concept will hold up to its promises or if some unforeseen technical problems get in the way.

Some of the outlined security concepts require quite big changes to be made to clients, servers, or even to the operational practices of CAs. Those changes will take quite some time and e.g. the EFF thinks that it will take a couple of years to build up the Sovereign Key system [5]. Some of the systems could be faster deployed, such as Pinning-based approaches, which are already used by Google for its own domains within the Chrome browser. Notary-based systems are also already available for end users. Even if some of the concepts will not establish themselves, they might be used in the period before a more comprehensive system is deployed. There are also other applications for some of the concepts: e.g. Notary-based systems can already be used to identify and track down the source of MitM attacks [15].

Time will tell which concepts will be widely accepted and if they hold up to their security promises.

5. REFERENCES

[1] P. Eckersley. 28C3: Sovereign Keys - A proposal for fixing attacks on CAs and DNSSEC. `https://www.youtube.com/watch?v=18pFTo3zVxk`, Dec. 2011. [last retrieved in September 2012].

[2] P. Eckersley. How secure is HTTPS today? How often is it attacked? `https://www.eff.org/deeplinks/2011/10/how-secure-https-today`, 2011. [last retrieved in September 2012].

[3] P. Eckersley. Iranian hackers obtain fraudulent HTTPS certificates: How close to a Web security meltdown did we get? `https://www.eff.org/deeplinks/2011/03/iranian-hackers-obtain-fraudulent-https/`, 2011. [last retrieved in September 2012].

[4] P. Eckersley. Sovereign Key Cryptography for Internet Domains. `https://git.eff.org/?p=sovereign-keys.git;a=blob;f=sovereign-key-design.txt;hb=master`, June 2012. [last retrieved in September 2012].

[5] Electronic Frontier Foundation. Sovereign Keys: A Proposal to Make HTTPS and Email More Secure. `https://www.eff.org/deeplinks/2011/11/sovereign-keys-proposal-make-https-and-email-more-secure`, Nov. 2011. [last retrieved in September 2012].

[6] Electronic Frontier Foundation. The Sovereign Keys project. `https://www.eff.org/sovereign-keys`, 2011. [last retrieved in September 2012].

[7] P. Hoffman and J. Schlyter. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698, IETF, August 2012.

[8] A. Langley. Certificate Transparency. `http://www.imperialviolet.org/2011/11/29/certtransparency.html`, Nov. 2011. [last retrieved in September 2012].

[9] A. Langley. Revocation doesn't work. `http://www.imperialviolet.org/2011/03/18/revocation.html`, 2011. [last retrieved in September 2012].

[10] B. Laurie and E. Kasper. Certificate Transparency v2.1a. `http://www.links.org/files/CertificateTransparencyVersion2.1a.pdf`, Sept. 2012. [last retrieved in September 2012].

[11] B. Laurie and A. Langley. Certificate Transparency. `http://www.certificate-transparency.org/`, 2012. [last retrieved in September 2012].

[12] M. Marlinspike. Defeating OCSP With The Character '3'. `http://www.thoughtcrime.org/papers/ocsp-attack.pdf`, 2009. [last retrieved in September 2012].

[13] M. Marlinspike. SSL And The Future Of Authenticity. `https://www.youtube.com/watch?v=Z7Wl2FW2TcA`, Aug. 2011. [last retrieved in September 2012].

[14] Mozilla Security Blog. DigiNotar removal follow up. `https://blog.mozilla.com/security/2011/09/02/diginotar-removal-follow-up/`, 2011. [last retrieved in September 2012].

[15] R. Holz, T. Riedmaier, N. Kammenhuber, and G. Carle. X.509 Forensics: Detecting and Localising the SSL/TLS Men-in-the-middle. In *Proc. 17th European Symposium on Research in Computer Security (ESORICS 2012)*, volume 7459/2012 of LNCS, pages 217–234, Pisa, Italy, Sept. 2012. Springer Verlag.

[16] T. Ritter. New Standards for browser-based trust - The recent acceleration of improvements. `http://ritter.vg/p/2012-TLS-Survey.pdf`, March 2012. [last retrieved in September 2012].

[17] C. Soghoian and S. Stamm. Certified lies: Detecting and defeating government interception attacks against SSL. In *Proc. 15th. Int. Conf. Financial Cryptography and Data Security (FC'11)*, Mar. 2011.

[18] A. Steingruebl. Perspectives on "perspectives". `http://www.thesecuritypractice.com/the_security_practice/2010/04/perspectives-on-perspectives.html`, April 2010. [last retrieved in September 2012].

[19] D. Wendlandt, D. Andersen, and A. Perrig. Perspectives: Improving SSH-style host authentication with multi-path probing. In *Proc. USENIX Annual Technical Conference*, Boston, MA, June 2008.

[20] J. Wierzbicki. Sovereign Key Draft Specification. `https://git.eff.org/?p=sovereign-keys.git;a=blob_plain;f=spec.txt;hb=master`, July 2012. [last retrieved in September 2012].

Rootkits

Stefan Liebald
Betreuer: Simon Stauber
Seminar Innovative Internet Technologien und Mobilkommunikation WS2012
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: liebald@in.tum.de

KURZFASSUNG

Rootkits sind eine Sammlung von Softwaretools, welche es Außenstehenden erlauben, unerkannt die Kontrolle über ein Computersystem zu behalten und dieses zu manipulieren, ohne dass legitimierte Nutzer in der Lage sind dies zu erkennen. Gerade Schadsoftware verwendet Rootkittechnologie zur Tarnung, diese sind also eine allgegenwärtige Gefahr.
Dieses Paper dient dazu, dem Leser die Eigenschaften und Gefahren von Rootkits näher zu bringen, so dass dieser in der Lage ist, die Grundlegenden Konzepte zu verstehen. So wird auf zwei verschiedene Rootkit-Typen eingegangen und speziell Kernel-Mode Rootkits einer genaueren Betrachtung unterzogen. Abschließend wird erläutert auf welche Weise ein Rootkit das Betriebssystem manipulieren kann und welche Möglichkeiten zu Abwehr und Erkennung existieren.

Schlüsselworte

User-Mode Rootkits, Kernel-Mode Rootkits, LKM, Linux Kernel, Kernel Manipulation

1. EINLEITUNG

Angriffe auf Rechensysteme sind eine allgegenwärtige Gefahr. Neben dem Ausspionieren von sensitiven Daten (wie z.B. Kreditkartendaten) wollen Angreifer den Zugriff auf ein System oft für eine längere Zeit behalten (z.B. für Botnetze). Hierzu versuchen sie die Kompromittierung des Rechners vor dem Benutzer zu verschleiern um Gegenmaßnahmen zu vermeiden.
Ein Mittel hierzu stellen Rootkits dar. Diese können, nach dem erstmaligen Erlangen von Rootrechten, auf einem System platziert werden und gewähren dem Angreifer ab diesem Zeitpunkt fortgesetzten privilegierten Zugriff durch Schaffung einer Hintertür. Des Weiteren sind sie in der Lage sich selbst und andere Schadsoftware vor Entdeckung zu schützen indem sie sich im System verstecken.
Ein prominentes Beispiel für den Einsatz von Rootkits stammt aus dem Jahr 2005. Das Musiklabel “Sony BMG” verwendete heimlich ein Rootkit als Teil seines Kopierschutz [15]. Musik CDs mit diesem “Schutz” wurden millionenfach verkauft und von den Käufern auf dem PC abgespielt, wodurch das Rootkit installiert wurde, welches auch durch andere Schadsoftware verwendet werden konnte.
Dieses Beispiel zeigt, Rootkits sind nicht nur eine theoretische Gefahr für die Sicherheit von Computersystemen, sondern auch eine ganz reale. Dieses Paper soll einen Überblick über Geschichte und Funktion von Rootkits vermitteln sowie einige Schutzmöglichkeiten aufzeigen. Hierbei liegt der Fokus auf Unix Rootkits.
Dazu gibt Abschnitt 2 eine Definition von Rootkits, sowie einem Überblick über ihre Entwicklung und erläutert skizzenhaft verschiedene Rootkit-Typen. Abschnitt 3 zeigt beispielhaft die Funktionsweise eines Kernel-Mode Rootkits, erläutert wie dieses im Kernel integriert werden kann und zeigt eine Möglichkeit diesen zu manipulieren. Abschnitt 4 stellt Optionen zur Erkennung und Abwehr von Rootkits vor. In Abschnitt 5 endet dieses Paper mit einem kurzen Ausblick.

2. HINTERGRUND

Um sich dem Thema Rootkits anzunähern, wird im folgenden Abschnitt eine Definition von Rootkits vorgestellt. Anschließend wird ein Blick auf die Historie der Rootkits geworfen und auf deren Taxonomie eingegangen.

2.1 Definition

Da Rootkits je nach Umfang und Zweck verschiedene Funktionen eines Betriebssystems beeinflussen können ist eine präzise Definition schwer zu finden. Hoglund [11] verwendet folgende grundlegende Definition der Eigenschaften eines Rootkits:

“Ein Rootkit ist dabei ein ’Kit’ aus kleinen, nützlichen Programmen, die einem Angreifer den fortgesetzten Zugriff auf root erlauben, dem mächtigsten Benutzer auf einem Computer.[. . .] Ein Rootkit ist ein Satz von Programmen und Code, der eine dauerhafte und nicht aufzuspürende Präzens auf einem Computer erlauben.”

Ein einmal integriertes Rootkit ist in der Lage sich selbst und andere Software vor dem rechtmäßigen Administrator eines Systems zu verbergen. Im Idealfall ist sich der Benutzer der Kompromittierung nicht bewusst. Nach einer Infizierung des Systems kann ein Rootkit (je nach Umfang) theoretisch alle Interaktion eines Benutzers mit diesem abfangen, kontrollieren und gegebenenfalls manipulieren. In vielen Fällen bietet ein Rootkit außerdem eine Hintertür zum System, mittels derer ein Angreifer wiederholt versteckten Rootzugriff erlangen kann. Ein Rootkit dient **nicht** dem erstmaligem Erlangen von Rootrechten, welche nötig sind um ein Rootkit im System zu integrieren. Diese Rechte müssen vor der Integrierung des Rootkits auf andere Art erlangt werden (z.B. mittels Schadsoftware, Social Engineering).
Aber nicht nur Schadsoftware verwendet Rootkits, sondern auch hostbasierte Überwachungssysteme zum Schutz

doi: 10.2312/NET-2013-02-1_08

eines Computers vor Angreifern können diese Software verwenden.

2.2 Geschichte

Erste Software mit der Eigenschaft zur Tarnung trat erstmals in den späten 1980er Jahren auf. Einer der ersten Viren mit Tarnfunktionen war der Brain Virus für DOS [1], welcher sich im Bootsektor eines im DOS FAT formatierten Speichermediums eingenistet hat. Wann immer versucht wurde aus dem infizierten Boot Sektor zu lesen, hat Brain diesen Zugriff umgeleitet und den originalen Boot Sektor vorgezeigt.
Kurze Zeit später traten die ersten Rootkits auf UNIX Systemen auf. Diese Rootkits ersetzten Benutzerprogramme (z.B. *login* oder *ps*) durch eigene Varianten und sorgten dafür, dass Informationen über ihre Tätigkeit aus den Logs entfernt wurden. Eine erweiterte Funktionalität war das Ersetzen von Bibliotheken, welche von den Benutzerprogrammen verwendet wurden. Diese wurden dadurch indirekt beeinflusst, blieben selbst aber unverändert.
Da ersetzte Dateien relativ leicht aufspürbar waren, traten in den späten 90er Jahren erste Rootkits auf, die den Betriebssystemkern (Kernel) beeinflussten. Auf diese Weise waren sie in der Lage Schutzmöglichkeiten zu unterlaufen und waren auf diese Weise noch schwerer aufzuspüren. In dieser Zeit kamen schließlich auch die ersten Windows Rootkits in größerem Maße in Umlauf.

2.3 Taxonomie

Grundsätzlich existieren zwei verbreitete Typen von Rootkits, die User- und die Kernel-Mode Rootkits. Zur Einordnung werden die verschiedenen Berechtigungsstufen eines Systems betrachtet. Die meisten Prozessoren implementieren Privilegierungslevel, die nutzbare Prozessorinstruktionen oder den Zugriff auf bestimmte Speicherbereiche für einen Prozess einschränken. Oft werden diese Level als Ring 0-3 bezeichnet, wobei Ring 3 auch als User-Mode bezeichnet wird. Normale Benutzerprogramme laufen in diesem Modus. Hier gibt es Einschränkungen auf den Zugriff auf Hardware und Speicher, es darf nur auf den zugewiesenen (virtuellen) Speicherbereich zugegriffen werden und nicht auf Speicherbereiche anderer Programme. Auch der Zugriff auf den Kernel ist beschränkt.
Ring 1 und 2 werden nur in seltenen Fällen benutzt, Ring 0 entspricht der höchsten Privilegierungstufe, in welcher keine Einschränkungen gelten und in welchem der Kernel eines Betriebssystems läuft.

Abbildung 1 zeigt ein grundlegendes Modell eines Computersystems. Das Betriebssystem läuft auf der Hardware und verwaltet Zugriffe auf sie über den Kernel. Im Betriebssystem ausgeführte Programme laufen im User-Mode und können dabei auf Kernel Funktionen zugreifen. Ein Rootkit im User-Mode ersetzt diese Programmen oder Bibliotheken, welche von diesen genutzt werden, um das System zu manipulieren. Dabei könnte beispielsweise *ps* dahingehend geändert werden, dass es bei einem Aufruf vom Rootkit angestoßene Prozesse nicht länger auflistet, um diese zu verstecken.

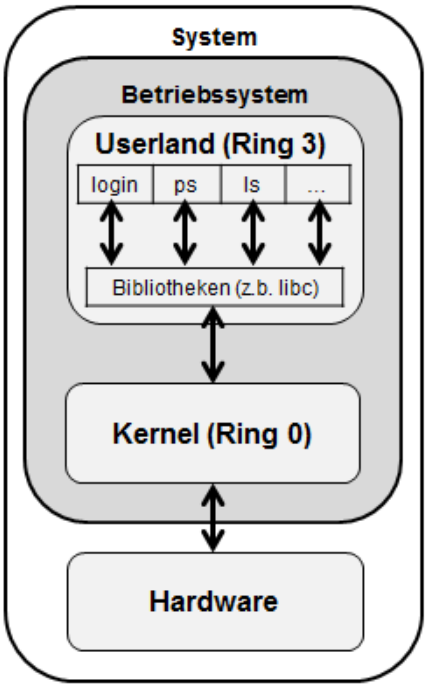


Abbildung 1: Schematische Darstellung der grundlegenden Struktur eines Systems

Ein Beispiel für ein User-Mode Rootkit[1] ist "Jynx2"[3], welches bestimmte Funktionsaufrufe aus Bibliotheken ersetzt, indem es eigene Bibliotheken vor den normalen Bibliotheken lädt ("LD_PRELOAD[2]"). Wird auf diese Art beispielsweise eine eigene Bibliothek mit der Funktion *printf()* vor der Standard C Bibliothek *libc* geladen, so wird die zuerst geladene *printf()* Funktion aus der eigenen Bibliothek verwendet. Nach der Installation versteckt Jynx2 unter anderem alle Dateien und Prozesse, die einer bestimmten Benutzer ID gehören oder deren Name mit einem vorher definierten String beginnt (standardmäßig "XxJynx"). Listing 1 zeigt die Auswirkung beispielhaft:

Listing 1: Verstecken eines Verzeichnisses durch Jynx2

```
1 # ls
2 datei1.txt datei2.txt
3 # mkdir XxJynx_Rootkit
4 # ls
5 datei1.txt datei2.txt
```

Das neu erstellte Verzeichnis *XxJynx_Rootkit* wird durch den Befehl *ls* nicht angezeigt, da es mit dem String "XxJynx" beginnt und deshalb durch das Rootkit verborgen wird.
Die Vielzahl der zu ersetzenden Dateien um alle Benutzerprogramme abzudecken, war einer der Gründe

[1] erschienen Mitte 2012, getestet unter Ubuntu 12.10
[2] LD_PRELOAD ist eine Umgebungsvariable, die eine Liste von Bibliotheksadressen (z.b. "/prelib/libc.so.6") enthält, die beim Starten von Programmen vor allen anderen Bibliotheken geladen werden

doi: 10.2312/NET-2013-02-1_08

für die Entwicklung von Kernel-Mode Rootkits. Wie in Abbildung 1 zu sehen ist, operiert der Kernel in der höchsten Privilegierungsstufe (Ring 0) und regelt den Zugriff von Userland Programmen auf die Hardware. Ein Kernel-Mode Rootkit kann an dieser Stelle ansetzen und das Verhalten des Kernels manipulieren, so dass zwar die Userland Programme wie *ls* nicht ausgetauscht werden und unverdächtig wirken, die von ihnen oder einer eingebundenen Bibliothek verwendeten Kernelfunktionalitäten aber manipuliert sind. Auf diese Weise lassen sich verschiedene Benutzerprogramme zentral manipulieren, ohne diese direkt zu ersetzen.
Ein Beispiel zur Funktionalität und Installation von Kernel-Mode Rootkits wird in Abschnitt 3 beschrieben.

3. KERNEL-MODE ROOTKITS

In diesem Abschnitt werden Kernel-Mode Rootkits näher untersucht. Dazu werden Möglichkeiten aufgezeigt, wie ein derartiges Rootkit in den Kernel eingeschleust werden kann. Hierfür wird die gebräuchliche Vorgehensweise mittels Kernelmodulen untersucht. Im Anschluss wird die Funktionsweise eines Kernel-Mode Rootkits anhand eines Beispiels genauer erläutert.

3.1 Integration in den Kernel

Um ein Rootkit in den Kernel zu bringen, stehen mehrere Möglichkeiten zur Verfügung [16], von denen je nach Kernelversion allerdings nicht immer alle verwendet werden können (siehe Abschnitt 4.1):

- Modifizieren des Abbilds des Kernels im Hauptspeicher (repräsentiert durch /dev/kmem).
- Austausch des Kernelabbilds auf der Festplatte durch eine eigene Version (/boot/vmlinuz).
- Ausführen des Rootkits im User Mode mit Kernel Mode Privilegien (Unterstützung durch Kernel erforderlich, z.B. Kernel Mode Linux [13]).
- Integration in den Kernel durch Kernelmodule (engl.: Loadable Kernel Modules, LKM).

Im Folgenden soll die gebräuchliche Methode der Integration von Kernel-Rootkits in den Kernel mithilfe von LKMs detaillierter betrachtet werden. LKMs sind als Binärdatei gespeichert und erweitern den Kernel um zusätzliche Funktionalitäten (z.B. Gerätetreiber). Sie können während dem laufenden Betrieb in den Kernel mit aufgenommen und von diesem benutzt werden. Der Code wird in der selben Privilegierungsstufe wie der Kernel ausgeführt (Ring 0). Mit LKMs ist es möglich, tiefergreifende Veränderungen im Betriebssystemkern zu bewirken, ohne das System dabei neustarten zu müssen. Der Code eines LKMs muss dabei zwei bestimmte Methoden beinhalten, *init_module()*, welche ausgeführt wird wenn das Modul in den Kernel geladen wird und *cleanup_module()*, welche beim Entladen aus dem Kernel aufgerufen wird. Mittels dieser Aufrufe integriert sich das Modul in den Kernel, so das diesem die neuen Funktionalitäten zur verfügung stehen (siehe Abschnitt 3.3).

3.2 Systemaufrufe

Zum Verständnis der Funktionsweise von Kernel-Mode Rootkits, ist es nötig zu verstehen, wie die für Rootkits relevanten Teile des Kernels funktionieren, beziehungsweise welche Aufgaben der Kernel für gewöhnliche Userland Programme ausführt.
Funktionen des Kernels sind unter anderem Speicher-, Prozess- und Dateisystem Management, sowie die Verwaltung von Gerätetreibern und des Netzwerkzugriffs. Wenn ein Benutzerprogramm Zugriff auf einen vom Kernel verwalteten Bereich benötigt (z.B. zum Versenden eines Netzwerkpakets), muss es auf entsprechende Funktionen des Kernels zugreifen, welche die Anfrage dann behandeln.
Für die Kommunikation mit dem Kernel stehen, je nach Anfrage verschiedene, Systemaufrufe zur Verfügung. Will ein Benutzerprogramm beispielsweise eine Datei öffnen, so muss es dazu, unter anderem, den *sys_open()* Systemaufruf des Kernels nutzen. Im Normalfall werden für diesen Zugriff von Benutzerprogrammen auf den Kernel Funktionen von Bibliotheken im Userland benutzt, wie beispielsweise *fopen()* aus der Standard C Bibliothek *libc*. Die Bibliotheken rufen dann ihrerseits den Systemaufruf auf.
In der Kernelversion 3.5.0 gibt es über 270 verschiedene Systemaufrufe. Die Speicheradressen dieser Funktionen sind in der sogenannten Systemaufrufstabelle abgelegt und werden über diese verwaltet.
Kernel Rootkits basieren darauf, auf diese Tabelle zuzugreifen und bestimmte Systemaufrufe (je nach Ziel des Angreifers) durch eigene, angepasste Varianten zu ersetzen oder neue Systemaufrufe zu erstellen, welche durch Userland Schadsoftware genutzt werden können (siehe Abschnitt 3.3.2).

3.3 LKM Kernel-Mode Rootkits

Im folgenden Abschnitt wird gezeigt, wie ein Rootkit als Kernelmodul in den Kernel geladen wird und dort Systemaufrufe manipulieren kann.

3.3.1 Zugriff auf die Systemaufrufstabelle

Um Systemaufrufe zu beeinflussen, benötigt ein Kernel-Mode Rootkit Zugriff auf die Systemaufrufstabelle. Um an die Adresse der Tabelle im Speicher zu gelangen, kann aus */boot/System.map-Kernelversion* die Speicheradresse ausgelesen werden an welcher sich die Tabelle befindet, um sie dann im Rootkit direkt zu adressieren.
Abbildung 2 zeigt den entsprechenden Eintrag in der

```
c15d501f r __func__.22707
c15d5028 r print_trace_ops
c15d5040 R sys_call_table
c15d55e0 r k8_nops
c15d5620 r p6_nops
c15d5660 r intel_nops
```

Abbildung 2: Adresse der Systemaufrufstabelle in System.map

System.map. Die Adresse der Systemaufrufstabelle unterscheidet sich je nach Kernelversion. Der linke Eintrag stellt die Adresse im Speicher dar, der rechte gibt den Symbolnamen der Systemvariable an, welche

doi: 10.2312/NET-2013-02-1_08

sich an dieser Stelle befindet (in diesem Fall sys_call_table als Symbolname für die Systemaufrufstabelle). Das R weist darauf hin, dass der Speicherbereich, in dem die Tabelle liegt, schreibgeschützt (read-only) ist. Um die Tabelle manipulieren zu können, muss dieser Schutz aufgehoben werden, wie es beispielhaft in Abschnitt 3.3.2 gezeigt wird. Ein funktionsfähiges Beispiel eines Rootkits, welches Systemaufrufe ersetzt, findet sich auch im Blog von Styx [19]. Dieser zeigt auch eine Möglichkeit wie ein Rootkit so erweitert werden kann, dass es die Adresse der Systemaufrufstabelle dynamisch während dem Kompilieren bzw. zur Laufzeit aus der System.map auslesen kann [18]. Auf diese Art und Weise ist es nicht länger nötig den Speicherort der Tabelle für verschiedene Systeme manuell zu suchen.

3.3.2 Ersetzen eines Systemaufrufs

Nachdem man Zugriff auf die Systemaufrufstabelle besitzt, kann man verschiedene Systemaufrufe durch eigene Varianten ersetzen. Im Folgenden soll die Funktionalität eines Kernel-Mode Rootkits anhand der wichtigsten Codeausschnitte eines einfachen Beispiels gezeigt werden. Das Rootkit soll den Systemaufruf *sys_nanosleep()* durch eine eigene Variante ersetze. Normalerweise wird der Systemaufruf beispielsweise von der Funktion sleep() verwendet, welche die Ausführung eines Programms um eine angegebene Zeit verzögert. Die Rootkitversion des Aurufs soll nun bei einer bestimmten zu wartenden Zeit anstatt zu warten dem aufrufenden Prozess Rootrechte verleihen. Listing 2 zeigt den Code der zur Ausführung kommt wenn das Rootkit als Modul in den Kernel geladen wird.

Listing 2: init Funktion des Rootkit Moduls

```
1  unsigned long *syscall_table = (unsigned long *)0xc15d5040;
2  asmlinkage int (*original_nanosleep)(struct timespec *, struct timespec *);
3
4  static int init(void) {
5    write_cr0(read_cr0() & (~ 0x10000));
6
7    original_nanosleep = (void *)syscall_table[__NR_nanosleep];
8    syscall_table[__NR_nanosleep] = new_nanosleep;
9
10   write_cr0(read_cr0() | 0x10000);
11   return 0;
12 }
```

In Zeile 1 wird die Systemaufrufstabelle anhand ihrer vorher aus /boot/System.map ausgelesenen Position im Speicher adressiert, so dass sie im Weiteren manipuliert werden kann. Ab Zeile 4 ist der Code zu sehen, der beim Laden des Moduls in den Kernel ausgeführt wird (vgl. Abschnitt 3.1). In Zeile 5 wird zuerst durch die Funktion *write_cr0()* der Schreibschutz des Speicherbereichs aufgehoben, in dem auch die Systemaufrufstabelle liegt. Dies geschieht durch Setzen des 17. Bits im dafür verantwortlichen Register CR0 auf 0 [12]. Anschließend wird die Originaladresse des zu ersetzenden Systemaufrufs (*sys_nanosleep()*) gesichert und der entsprechende Eintrag durch die Adresse der Rootkit Funktion *new_nanosleep()* ersetzt (Zeile 8), welche ebenfalls als Code im Modul vorliegt. Abschließend wird der Schreibschutz des Speicherbereichs wieder reaktiviert.

Der neue Systemaufruf des Rootkits *new_nanosleep()* ist in Listing 3 zu sehen. Immer wenn der ursprüngliche Systemaufruf angefragt wird, springt stattdessen der Rootkit Code ein.

Listing 3: Rootkitversion des ersetzten Systemaufrufs

```
1  asmlinkage int new_nanosleep(struct timespec *req , struct timespec *rem){
2    struct cred *new;
3    if (req->tv_sec==1234){
4      new = prepare_creds();
5      if ( new != NULL ) {
6         new->uid = new->gid = 0;
7         new->euid = new->egid = 0;
8         new->suid = new->sgid = 0;
9         new->fsuid = new->fsgid = 0;
10        commit_creds(new);
11        }
12       return 0;
13     }
14   else {
15     return original_nanosleep(req, rem);
16   }
17 }
```

Dieser überprüft zuerst wie lange *sys_nanosleep()* schlafen soll (Zeile 3) und verleiht dem aufrufenden Prozess Rootrechte (Zeilen 2,4,6-10) wenn die Wartezeit 1234 Sekunden beträgt. Dazu werden die Werte der Benutzer-ID[3] (=uid) und Gruppen-ID[4] (=gid) auf den Wert 0 gesetzt, was intern dem Root Benutzer entspricht[5]. Für alle anderen Zeiten wird in Zeile 15 der originale Systemaufruf aufgerufen.
Danach besitzt der Prozess der den Systemaufruf ausgeführt hat Rootrechte und kann im Folgenden mit diesen operieren, auch wenn nie das Passwort des eigentlichen Rootbenutzers eingegeben werden musste.
Listing 4 zeigt den C Code für ein (Userland) Programm, bei dessen Ausführung (bei installiertem Rootkit) eine Shell mit Rootzugriff gestartet wird ohne dass eine Authentisierung des Benutzers nötig ist[6].

Listing 4: Programm zum Testen des Rootkits

```
1 int main () {
2  sleep(1234);
3  system ("/bin/sh");
4  return 0;
5 }
```

Oft werden (vor allem in der Forschung) Kernelrootkits zusammen mit einem Userland Steuerprogramm genutzt, mit Hilfe dessen z.B. Funktionen des Rootkits dynamisch (de-)aktiviert werden können. Dieses Steuerprogramm könnte, wie in Listing 4 gezeigt, mittels des *sys_nanosleep()* Systemaufrufs arbeiten und ungenutzte Signale[7] zur Steuerung verwenden, die das Kernelrootkit abfängt. Ein Kernel-Mode Rootkit, welches den Systemaufruf *sys_kill()* nutzt ist KNARK [14] (für die Kernelversionen 2.2 und 2.4), welches standardmäßig mittels *killall -31* die Anwesenheit bestimmter Prozesse versteckt und sie nach dem Kommando *killall -32* wieder anzeigt.

Abbildung 3 verdeutlicht die Funktionsweise des Rootkits und zeigt wie es eine eigene Funktion vor dem eigentlichen Systemaufruf platziert. Alle weiteren Aufrufe laufen nun

[3] repräsentiert intern den Benutzer anstelle des Benutzernamens
[4] Ein Benutzer gehört mindestens einer Gruppe an, welche eine Rolle bei seinen Zugriffsberechtigungen spielt
[5] Die Varianten euid,egidm,suid,sgid, fsuid und fsgid sind zum Verständniss von Rootkits hier nicht weiter relevant
[6] vollständiges Rootkit getestet unter Kernelversion 3.5.0
[7] kurze Nachrichten an Prozesse die bestimmte Reaktionen auslösen sollen und über den Kernel laufen

doi: 10.2312/NET-2013-02-1_08

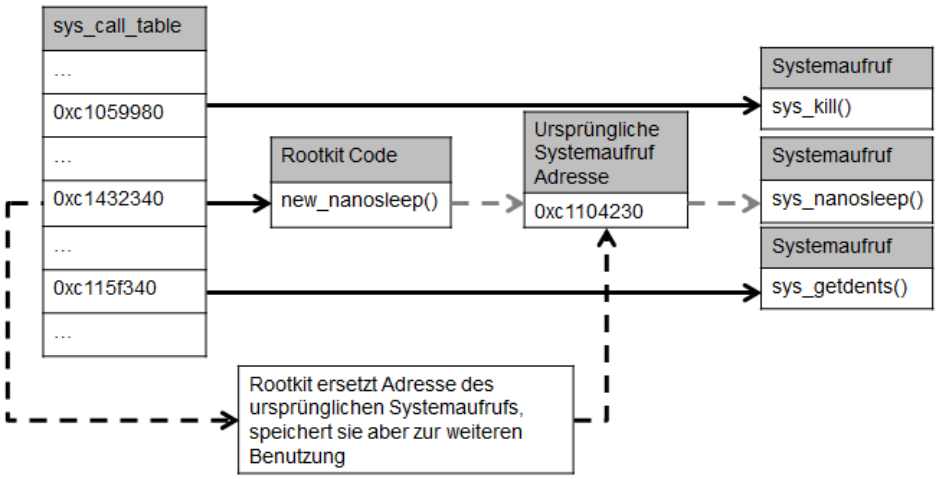


Abbildung 3: Beispiel für das Verhalten eines Kernelrootkits, welches einen Systemaufruf ersetzt

über die zwischengeschaltete Rootkitfunktion und können von dieser manipuliert werden.
Neben der beispielhaft beschriebenen Manipulation des Systemaufrufs *sys_nanosleep()* existieren noch weitere Möglichkeiten, das Systemverhalten zu beeinflussen:

- Verstecken von Dateien, Verzeichnissen und Prozessen: Durch Manipulation des Systemaufrufs *sys_getdents()* können Dateien und Verzeichnisse beispielsweise vor dem Programm *ls* verborgen werden.
- Manipulation eines Programmaufrufs: Mit Hilfe eines angepassten *sys_execve()* kann anstelle eines aufgerufenen Programms ein anderes ausgeführt werden.
- Benutzerrechte verändern: Rootrechte mittels *sys_setuid* verleihen
- Netzwerkverbindung verstecken
- Netzwerkpakete abfangen
- ...

Die aufgeführten Möglichkeiten zeigen einen Überblick dessen, wozu Kernel-Mode Rootkits fähig sind. Weitere Möglichkeiten und detailliertere Erläuterungen zeigt beispielsweise Peláez [16].

4. SCHUTZMÖGLICHKEITEN

Es zeigt sich, dass Rootkits eine große Bedrohung für die Sicherheit eines Systems darstellen. Entsprechend müssen Gegenmaßnahmen getroffen werden, um Rootkits die Manipulation des Systems zu erschweren und schon installierte Rootkits zu erkennen. Ein allgemeiner Überblick dazu zeigt z.B. Hannel [8]. Einige Methoden zu Erkennung und Abwehr werden in diesem Kapitel exemplarisch angesprochen.

4.1 Abwehr

Generell ist es wichtig, das System vor dem ersten Angriff zum Erlangen von Rootrechten zu schützen, welche nötig sind, um ein Rootkit zu platzieren.
Zusätzlich können auch Maßnahmen ergriffen werden, um das System zu härten. Dabei soll dem Angreifer die Installation des Rootkits zu erschwert, beziehungsweise die Funktion des Rootkits behindert werden. Nachfolgend sind einige Möglichkeiten aufgezeigt, die teilweise auch schon standartmäßig in Linux Systemen umgesetzt werden.
Um wichtige Dateien vor dem Ersetzen oder Verändern zu schützen, kann durch einen Benutzer mit Rootrechten ihr so genanntes immutable Flag gesetzt werden. Solange das Flag gesetzt ist, kann niemand diese Datei manipulieren bis das Flag wieder entfernt wird. Dadurch wird auch verhindert, dass die Datei durch ein Userland Rootkit ersetzt wird. Allerdings ist das für den Kernel selbst nicht möglich, wodurch Rootkits mittels LKM weiterhin den Kernel um ihren eigenen Code erweitern können.
Um einen Angriff durch LKM Kernel Rootkits zu vermeiden besteht die Möglichkeit den Kernel ohne die Unterstützung von Kernbelmodulen zu kompilieren. Diese Variante hat aber den Nachteil, dass alle benötigten Gerätetreiber (welche meist auf LKMs basieren) direkt in den Kernel integriert sein. Ein dynamisches Nachladen mittels Kernelmodulen ist dann nicht möglich. Systeme werden durch diese Einschränkung sehr unflexibel, da der Kernel für Änderungen immer neu kompiliert werden müsste.
Zusätzlich ist zu beachten, dass Rootkits nicht nur mittels LKMs in den Kernel gelangen können, sondern z.B. auch durch die Veränderung des Kernels im Speicher mittels /dev/kmem. Über diese spezielle Datei besteht ein Schreibzugriff auf den Speicherbereich des Kernels. Diese Möglichkeit ist aber in vielen Linuxdistributionen wie Ubuntu mittlerweile deaktiviert [5], da ein Zugriff über kmem in erster Linie eine Sicherheitsgefährdung darstellte, aber keinen sinnvollen Nutzen für das System mehr erfüllte. Eine mit der Linux Kernel Version 3.7 neu eingeführte Möglichkeit zum Schutz vor schädlichen Kernelmodulen ist die Verwendung von kryptographisch signierten Modulen [7]. Wird diese Option im Kernel aktiviert kann auch ein

doi: 10.2312/NET-2013-02-1_08

Benutzer mit Rootrechten keine Module mehr in den Kernel laden, welche nicht mit dem richtigen Schlüssel signiert wurden.
Wie erläutert wurde, verwenden Kernel Rootkits die Tabelle der Systemaufrufe um diese zu manipulieren. In früheren Kernelversionen (vor 2.6) wurde diese Tabelle noch vom Kernel exportiert und erlaubte so einen einfachen Zugriff für Kernelmodule mittels *extern void* sys_call_table[].*
In neueren Kernel Versionen wird die Tabelle standartmäßig nicht mehr exportiert und ist schreibgeschützt. Dadurch wird dem Rootkit der Zugriff auf die Adresse an der die Tabelle im Speicher liegt erschwert. Einige der möglichen Wege zu der Adresse liegen in der erwähnten System.map, die allerdings für den laufenden Betrieb nicht nötig ist und daher (nach einem Backup) theoretisch gelöscht werden kann[16][S.143].

4.2 Erkennungsmechanismen

Auch wenn ein System entsprechend geschützt bzw. gehärtet wurde, existiert keine Garantie, dass keine Kompromittierung vorliegt. Daher sollte regelmäßig überprüft werden, ob das Systemverhalten beeinflusst wird. Dazu existieren verschiedene Möglichkeiten.
Durch gezieltes Überprüfen der Rückgabe verschiedener Benutzerprogramme eines Systems, welche normalerweise gleiche Ergebnisse liefern (z.B. *ls* und *echo **), können Unterschiede aufgespürt werden, falls nur eines der Programme eine falsche Ausgabe liefert. Dieser Vergleich verschiedener Sichten wird als "Lie Detection" bezeichnet.
Ein weiteres Beispiel aus diesem Bereich ist das Aufspüren von Ports, welche von Rootkits als Hintertür geöffnet wurden und vor dem Benutzer versteckt werden. Dazu kann man die Ausgabe des Systems (*netstat -nlp*) mit einer externen Sicht vergleichen, die ein Portscanner wie zum Beispiel nmap liefert (*nmap -sT -p 1-65535 IP-Adresse*).
Chkrootkit [2] ist ein Tool, welches unter anderem überprüft, ob Prozesse versteckt werden, indem es Lie Detection anwendet. Prozessinformationen werden in Linux durch ein spezielles Dateisystem im Verzeichnis /proc repräsentiert [8]. Chkrootkit überprüft /proc/ und vergleicht den Inhalt mit dem Ergebnis des *ps* Befehls, um verborgene Prozesse aufzuspüren.
Ein weiteres Verfahren von Chkrootkit ist das Aufspüren von Konfigurationsdateien, welche von einigen Rootkits genutzt werden, um ihr Verhalten flexibel anpassen zu können (z.B. bei Jynx2 zum Wählen des Schlüsselworts und der ID die zum Verstecken einer Datei führt).
Eine weitere Möglichkeit zur Rootkiterkennung bietet eine regelmäßige Durchführung einer Integritätsprüfung von wichtigen Programmen oder Dateien. Dadurch können vor allem Userland Rootkits aufgespürt werden, da diese auf dem Ersetzen von Binärdateien basieren. Zu dieser Überprüfung wird die Datei mit einer vertrauenswürdigen Datei des Benutzerprogramms zu verglichen, die entweder früher als Backup gesichert wurde, oder beispielsweise aus einer vertrauenswürdigen, offiziellen Datenbank entnommen wird. Um den Vergleich zu vereinfachen wird meist ein Hashwert der Datei benutzt.
Rkhunter [4] ist ein Schutzprogramm, welches unter anderem dieses Verfahren einsetzt, um mittels MD5/SHA1 Hash Berechnung und einer aus dem Internet aktualisierbare Datenbank von Hashwerten veränderte Binärdateien aufzuspüren.
Einige Rootkits weisen spezielle Eigenschaften auf, anhand derer sie leichter zu identifizieren sind (Rootkit Signatur). KNARK zum Beispiel verwendet ungenutzte Signale an das System (*sys_kill(-31)*), um Prozesse zu verstecken. Diese können überprüft werden und bei unerwarteten Ereignissen (z.B. Verschwinden des Prozesses) eine Meldung ausgegeben werden.
Im Vergleich zu anderer Rootkiterkennungssoftware ist das letzte rkhunter Update relativ aktuell (Mai 2012), was für Rootkiterkennung eine wichtige Rolle spielt, da Rootkitersteller versuchen ihre Software vor Entdeckung zu schützen.
Einige weiterführende Tools zur Systemanalyse und -härtung beschreibt Claudia Eckert [6][Seite 179ff].

Gerade bei Kernel-Mode Rootkits läuft die Erkennung von Rootkits auf ein Wettrüsten zwischen Sicherheitssoftware und Rootkits hinaus. Rootkits versuchen sich möglichst komplett vor Erkennung zu verbergen, Sicherheitssoftware versucht Lücken in diesem "Versteck" zu finden. Wenn Erkennungstools versuchen eine Veränderung der Systemaufrufstabelle zu erkennen (z.B. durch Vergleich mit einem Snapshot aus einem früheren, unveränderten Zustand), können Rootkits darauf verzichten, auf die Originaltabelle zuzugreifen und beispielsweise mit einer Kopie der Tabelle arbeiten und Aufrufe an diese umleiten. Darüber hinaus kann auf die Manipulation der Systemaufrufstabelle verzichtet werden. Stattdessen überschreiben ein Rootkit die ersten Bytes der Systemaufrufe mit einem Sprungbefehl an die Adresse ihrer eigenen Variante.
Um den als Beispiel für Lie Detection erwähnten Portvergleich zu umgehen, kann ein Rootkit beispielsweise diesen erst dann öffnen, wenn davor versucht wurde in einer bestimmten Reihenfolge verschiedene Ports zu adressieren. Spricht ein externer Portscanner die Ports dann nicht zufällig in dieser Reihenfolge an, oder wird der Port gerade von dem Rootkit verwendet, ist er geschlossen und wird nicht erkannt.

5. ZUSAMMENFASSUNG UND AUSBLICK

Rootkits stellen eine große Sicherheitsgefährdung dar, welche sich seit den 80er Jahre kontinuierlich weiterentwickelt haben. Sie sind in der Lage unerkannt die Herrschaft über ein System zu behalten und dieses in dem Sinn ihres Erstellers zu beeinflussen, um diesem heimlichen Root Zugriff zu gestatten, sowie bestimmte Prozesse und Dateien vor Entdeckung zu schützen.
Grundlegend kann zwischen zwei Rootkit Typen unterschieden werden. User-Mode Rootkits setzen auf der Benutzerebene an und ersetzen komplette Programme bzw. Bibliotheken, während Kernel-Mode Rootkits Systemaufrufe des Kernels manipulieren. In diesem Paper wurde ein exemplarisches Kernel-Mode Rootkit vorgestellt, welches in der Lage ist einem gewöhnlichen Benutzer Rootzugriff auf das System zu gewähren.
Außerdem wurden einige Möglichkeiten aufgezeigt, wie basierend auf Lie Detection, Integritätsprüfung bzw. Signaturerkennung infizierte Systeme erkannt

[8]Für den Prozess mit der ID XXXX existiert das Verzeichnis /proc/XXXX/

doi: 10.2312/NET-2013-02-1_08

werden können. Je nach Rootkit können sich die Erkennungsmerkmale allerdings unterscheiden. Neuere Rootkitversionen sind in der Lage bekannte Erkennungsmechanismen zu umgehen, wie das in diesem Paper erwähnte User-Mode Rootkit Jynx2, welches weder von chkrootkit noch von rkhunter, zwei Rootkiterkennungstools, erkannt wurde.

Neuere Entwicklungen zeigen, dass sich Rootkits noch tiefer in einem System einnisten können, als es Kernel-Mode Rootkits tun.
Rootkits, die die Möglichkeiten der Virtualisierung benutzen, verschieben dabei das komplette Betriebssystem in eine virtuelle Umgebung. Sie werden als "virtual machine based Rootkit", kurz VMBR bezeichnet und setzen sich zwischen Hardware und Kernel und täuschen diesem vor, weiterhin volle Kontrolle über die Hardware zu besitzen. Beispiele sind Subvirt[10] oder Bluepill[17], beide vorgestellt im Jahr 2006.
Eine weitere Variante sind Firmware-Rootkits, welche sich direkt in der Hardware Ebene festsetzen und die Firmware beeinflussen. Ein solches Rootkit von John Heasman [9] überlebte sogar eine komplette Neuinstallation des Betriebssystems des kompromittierten Systems, in dem es sich im BIOS festsetzte.
Rootkits sind ein aktuelles Thema, welches gerade wegen den Eigenschaften sich vor Benutzern zu verbergen aufmerksam verfolgt werden sollte, um den Sicherheitsanforderungen von heutigen Systemen gerecht werden zu können.

6. LITERATUR

[1] Brain virus. Webseite. http://www.f-secure.com/v-descs/brain.shtml; besucht am 19.Dezember 2012.

[2] chkrootkit. Webseite. http://www.chkrootkit.org/; besucht am 18.Dezember 2012.

[3] Jynx2 rootkit. Webseite. http://www.blackhatlibrary.net/Jynx; besucht am 18.Dezember 2012.

[4] The rootkit hunter project. Webseite. http://rkhunter.sourceforge.net/; besucht am 18.Dezember 2012.

[5] Ubuntu security features. Webseite. https://wiki.ubuntu.com/Security/Features#dev-kmem; besucht am 19.Dezember 2012.

[6] C. Eckert. *IT-Sicherheit*. Oldenburg Wissenschaftsverlag GmbH, 2012. 7. Auflage.

[7] F. Grosshans. Linux kernel 3.7. Webseite, 2012. http://kernelnewbies.org/Linux_3.7; besucht am 2.Januar 2013.

[8] J. Hannel. Linux rootkits for beginners - from prevention to removal. Technical report, SANS Institute, Januar 2003.

[9] J. Heasman. Implementing and detecting a pci rootkit. *Black Hat DC 2007*, 2007.

[10] heise online. Microsoft demonstriert virtualisierungs-rootkit. Webseite. http://www.heise.de/newsticker/meldung/Microsoft-demonstriert-Virtualisierungs-Rootkit-110190.html; besucht am 17.Dezember 2012.

[11] G. Hoglund and J. Butler. *Rootkits*. Addison-Wesley Verlag, 2006.

[12] Intel. Intel® 64 and ia-32 architectures software developer's manual. Technical report, Intel, 2012.

[13] T. Maeda. Kernel mode linux. Webseite. http://web.yl.is.s.u-tokyo.ac.jp/~tosh/kml/; besucht am 20.Dezember 2012.

[14] T. Miller. Analysis of the knark rootkit. Webseite. http://www.ouah.org/tobyknark.html; besucht am 16.Dezember 2012.

[15] D. K. Mulligan and A. K. Perzanowski. The magnificence of the desaster: Reconstructing the sony bmg rootkit incident. *Berkeley Technology Law Journal [Vol. 22:1157 2007]*, pages 1158–1189, 2007.

[16] R. S. Peláez. Linux kernel rootkits: protecting the system's "ring-zero". Technical report, SANS Institute, Mai 2004.

[17] J. Rutkowska. Introducing blue pill. Webseite. http://theinvisiblethings.blogspot.de/2006/06/introducing-blue-pill.html; besucht am 19.Dezember 2012.

[18] styx. Syscall hijacking: Dynamically obtain syscall table address (kernel 2.6.x). Webseite. http://memset.wordpress.com/2011/01/20/syscall-hijacking-dynamically-obtain-syscall-table-address-kernel-2-6-x/; besucht am 13.Dezember 2012.

[19] styx. Syscall hijacking: Simple rootkit (kernel 2.6.x). Webseite. http://memset.wordpress.com/2010/12/28/syscall-hijacking-simple-rootkit-kernel-2-6-x/; besucht am 13.Dezember 2012.

doi: 10.2312/NET-2013-02-1_08

Software Defined Networking mit OpenFlow

Josias Montag
Betreuer: Daniel Raumer, Florian Wohlfart
Hauptseminar: Innovative Internettechnologien und Mobilkommunikation
WS 2012/2013
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Mail: montag@in.tum.de

KURZFASSUNG

Diese Ausarbeitung behandelt das Software Defined Networking (SDN) mit OpenFlow. Das SDN lagert die Kontrollinstanz des Netzwerkes aus und ermöglicht die Virtualisierung auf Netzwerkebene, welche langfristig durch sich veränderte Anforderungen an die Netzwerk Infrastrukturen nötig wird und Einschränkungen von klassischen Netzwerken entgegenwirkt. Technisch gesehen handelt es sich bei OpenFlow um ein Layer 2 Protokoll. Für die Nutzung von OpenFlow ist ein vom Hersteller angepasster Switch nötig, welche inzwischen zahlreiche Hersteller anbieten. Praktische Anwendungen, die die Vorteile des SDNs ausnutzen, sind Experimente in Produktivnetzwerken, Load Balancing, Clouds, transparente Dienste oder große Datencenter. Für die Programmierung solche virtueller Netzwerke bietet sich die dafür spezialisierte Programmiersprache Frenetic an. Obwohl das SDN und OpenFlow viele praktische Anwendungszwecke bietet, kommt es bisher nur in sehr speziellen Szenarien zum Einsatz.

Schlüsselworte

Software Defined Networking, OpenFlow, Netzwerk Virtualisierung, Frenetic

1. EINLEITUNG UND DEFINITION

„Virtualisierung“ und „Cloud“ sind momentan die Schlagwörter der IT-Industrie und haben sich von einem Trend zu einem nicht mehr wegzudenkenden Standard im Bereich von IT-Infrastrukturen entwickelt. [10]

Unter Virtualisierung versteht man „Technologien, die entworfen wurden, um eine Ebene der Abstraktion zwischen den Hardware Systemen und der darauf laufenden Software zu bereitzustellen“ [20]. Im Bereich der Server Virtualisierung bedeutet dies, dass die physisch vorhandenen Ressourcen der Hardware in verschiedene virtuelle Pools aufgeteilt oder zusammengefasst werden. Dadurch wird eine Abstraktion erreicht, die es ermöglicht die einzelnen virtuellen Maschinen unabhängig von der eigentlichen Hardware zu betreiben, zu duplizieren oder zu verschieben.

Wendet man die obige Definition von Virtualisierung auch auf Netzwerk Infrastrukturen an, ergeben dich die Grundzüge des „Software Defined Networking“ (SDN). Im Deutschen könnte SDN am ehesten als „durch Software gesteuerter Netzwerkbetrieb“ übersetzt werden. Unter Software Defined Networking versteht man das Entkoppeln der Kontroll-Ebene von der Daten-Ebene in der Netzwerk Architektur. Durch das Verschieben der Kontrollinstanz wird eine Abstraktion und Virtualisierung des Netzwerkes erreicht [12, S. 7].

Ziel der SDN Architektur ist es, unabhängig vom Hersteller der einzelnen Netzwerkgeräte, Netzwerk Ressourcen mittels dynamischer und automatisierter Software einzurichten, zu steuern, abzusichern und zu optimieren [12, S. 8].

Diese Ausarbeitung wird zunächst die Möglichkeiten des SDN gegenüber klassischen Netzwerken sowie die Funktionen, Ziele und Absichten von OpenFlow erklärt. Dabei werden unterschiedliche praktische Anwendungsfälle für die Nutzung SDN Architekturen dargelegt. Außerdem wird die Programmierung von SDN-Controllern mit der Programmiersprache Frenetic kurz aufgezeigt. Abschließend setzt sich diese Ausarbeitung kritisch mit den Limits und der Zukunft von OpenFlow auseinander.

2. EINSCHRÄNKUNGEN VON KLASSISCHEN NETZWERKEN

Die Anforderungen an die Netzwerk Infrastrukturen haben sich, besonders in den letzten Jahren, stark verändert. Gründe dafür sind unter anderem neue „On-Demand“ Geschäftsmodelle, die zunehmende Anzahl von Geräten, wie zum Beispiel durch Smartphones, und das Verlangen nach immer größeren Bandbreiten. Klassische Netzwerkstrukutren, dessen grundsätzlichen Strukturen noch aus den frühen 90er Jahren stammen, sind dafür teilweise nicht ausgelegt worden.

Verändertes Nutzerverhalten und größere Bandbreiten: Klassische Netzwerke sind starr und wurden für Client-Server Verbindungen konzipiert, die für eine bestimmte Zeit aufgebaut werden und anschließend wieder getrennt werden. Diese Architektur passt teilweise nicht mehr zu modernen Anwendungen, die auf viele verschieden Server gleichzeitig zugreifen und beispielsweise „Push“-Dienste nutzen. Der Trend geht zu „Always-On“ und der Nutzer möchte von überall auf der Welt jederzeit auf seine Daten zugreifen können. Um diese Bedürfnisse zu befriedigen, muss das Netzwerk flexibel werden und fähig sein sich in Echtzeit an das Nutzerverhalten anzupassen: Beispielsweise müssen die immer größer werdenden Bandbreiten dynamisch Anwendungen oder Nutzern zugewiesen werden können oder das Netzwerk muss selbstständig Engpässe erkennen

doi: 10.2312/NET-2013-02-1_09

und darauf reagieren können. [12, S.3]

Komplexität und Verwaltbarkeit: Die zunehmende Anzahl der Hosts in den, meist historisch gewachsenen, Netzwerken führt zu einer hohen Komplexität bei der Verwaltung. Durch die zunehmende Server-Virtualisierung vervielfacht sich zusätzlich die Anzahl der virtuellen Hosts. Diese Hosts in die unterschiedlichen Konfigurationsinterfaces der verschiedenen Switches und Router für beispielsweise ACLs (Access Control Lists), VLANs (Virtuelle Subnetze) oder QoS (Quality of Service) einzutragen ist ein hoher Verwaltungsaufwand. Diese Aufgaben können durch SDN automatisiert werden und damit die OPEX Kosten (Betriebsausgaben) gesenkt werden. [12, S.4,5]

Abhängigkeit von den Geräteherstellern: Hersteller von Switches oder Routern bieten oft zentrale Verwaltungsmöglichkeiten für ihre Geräte an. Diese funktionieren jedoch nicht herstellerübergreifend, wodurch man von einem bestimmten Hersteller, dessen Support und dessen Produktzyklen abhängig ist. Setzt man auf einen einheitlichen Standard wie OpenFlow ist man unabhängig vom Hersteller und kann beliebige Geräte kombinieren. Zudem können erhebliche Kosten für die Erneuerung von Switches eingespart weden, da die eigene Software auf auf den Switches gegebenenfalls aktualisiert oder angepasst weden kann, ohne dass neue Hardware gekauft werden muss. [12, S.6]

Skalierungs- und Anpassungsmöglichkeiten: Durch „On-Demand“ Geschäftsmodelle muss sich das Netzwerk ebenfalls „On-Demand“ an die Anforderungen anpassen können. Das Hinzufügen und die Konfiguration von neuen Netzwerkgeräten kann durch SDN automatisiert werden und daher in viel kürzerer Zeit erfolgen. [12, S.6]

3. SOFTWARE DEFINED NETWORKING MIT OPENFLOW

OpenFlow ist ein Open Source Projekt hat sich als die Standardschnittstelle für SDN-Architekturen etabliert. Die ursprüngliche Spezifikation von OpenFlow wurde 2008 an der Stanford Universität verfasst. Seit 2011 wird OpenFlow von der Open Networking Foundation (ONF) betreut. Die ONF wurde von der Deutschen Telekom, Facebook, Google, Microsoft, Verizon, und Yahoo! gegründet und zählt inzwischen über 80 Mitglieder. [1, 9]

Abbildung 1: OpenFlow Logo [1]

3.1 OpenFlow Unterstützung

Grundvoraussetzung für die Nutzung von OpenFlow sind mit OpenFlow kompatible Switches bzw. Router. Dabei unterscheidet man zwischen dedizierten OpenFlow Switches („Type 0“) und OpenFlow-Enabeld Switches („Type 1“). Die Type-0 Switches funktionieren nur im OpenFlow-Modus, das heißt sie bieten sonst keine klassischen Switch Funktionalitäten. Bei den OpenFlow-Enabeld Switches dahingegen ist OpenFlow eine Zusatzfunktion. Der Hersteller muss das vorhandene geschlossene System nicht offenlegen und OpenFlow kann trotzdem auf die Funktionalitäten des Switches zurückgreifen. (Kapitel 3.2) [16, S. 71]

Alle größeren Hersteller bieten solche Geräte bzw. passende Firmware-Updates an: IBM galt als Vorreiter und bot als einer der Ersten OpenFlow enabled Switche und sogar OpenFlow Controller an [7]. HP ist Mitglied der Open Networking Foundation und bietet derzeit 16 OpenFlow fähige Switche an [4]. Cisco setzt auf die eigene Lösung „Cisco Open Network Environment“, einem Framework mit verschiedenen APIs zur Steuerung von Cisco-Geräten, das jedoch ebenfalls eine OpenFlow Schnittstelle anbietet [2].

Alternativ besteht die Möglichkeit OpenFlow auf Software Routern zu betreiben, wofür Images für beispielsweise Debian und sogar ein Modul für die Heim-Router Software DD-WRT und OpenWRT zur Verfügung stehen [6].

3.2 Funktionsweise von OpenFlow

Bei OpenFlow handelt es sich um ein Layer 2 Protokoll mit der grundsätzlichen Zielsetzung, die Steuerung des Netzwerkes von den einzelnen Netzwerkgeräten von unterschiedlichen Herstellern in einen zentralisierten Software Controller zu verschieben. Abbildung 2 zeigt diese Architektur:

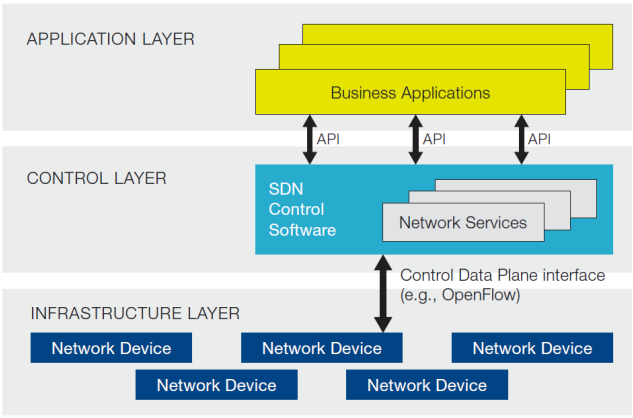


Abbildung 2: SDN Architektur [12, S. 7]

Switche und Router im Netzwerk (die Infrastructure Layer) leiten nur noch die Datenpakete weiter. Die eigentlichen Entscheidungen im Netzwerk trifft ein zentralisierter und programmierbarer SDN-Controller (die Control Layer). Der SDN-Controller erlangt so eine Gesamtsicht über die Netzwerk Infrastruktur. Applikationen und Dienste (die Application Layer) im Netzwerk können nun über eine API des SDN-Controllers Echtzeitinformationen über das Netzwerk bekommen oder sogar die Steuerung der Datenflüsse im Netzwerk direkt beeinflussen. Das OpenFlow Protokoll dient hierbei als Vermittler zwischen Netzwerkgerät und SDN-Controller.

OpenFlow identifiziert einzelne Datenströme („Flows“) und behandelt diese entweder nach statisch definierten oder auch nach dynamisch programmierten Regeln. Ein Flow könnte zum Beispiel eine TCP-Verbindung, alle Pakete von einer

doi: 10.2312/NET-2013-02-1_09

bestimmten IP oder MAC Adresse oder Pakete mit einem bestimmten VLAN-Tag sein. Die OpenFlow Switche prüfen zunächst für jeden neuen Flow, ob sich eine Regel für diesen Datenstrom in der lokalen Flow Table des Switches befindet. Falls für den Datenstrom noch keine Regel vorhanden ist, wird bei der zentralen SDN-Controller-Software nachgefragt, was mit diesem Flow geschehen soll. Die Verbindung zwischen Switch und Controller erfolgt über den Secure Channel mittels des OpenFlow-Protokolls. Diese, so vom OpenFlow Controller erlangte Regel, wird nun in die lokale Flow Table des Switches eingetragen und auf diesen Datenstrom, sowie alle zukünftigen auf diese Regel passenden Flows, angewendet [12, S. 9] [16, S. 71].

Abbildung 3 zeigt ein vereinfachtes Beispiel einer Flow Table eines OpenFlow Gerätes.

OpenFlow Enabled Network Device

MAC src	MAC dst	IP src	IP dst	TCP dport	Action	Count
*	10:20:*	*	*	*	port 1	250
*	*	*	5.6.7.8	*	port 2	300
*	*	*	*	25	drop	892
*	*	*	192.*	*	local	120
*	*	*	*	*	controller	11

Abbildung 3: Flow Table [12, S.9]

Für die definierten Actions in der Flow-Table gibt es verschiedene Möglichkeiten: [16, S. 71]

1. Weiterleiten der Pakete an einen oder mehrere Ports.
2. Verkapseln des Paketes um es dann über den Secure Channel an den Controller zu senden. Normalerweise geschieht dies mit dem ersten Paket eines Flows, um eine Regel für diesen Flow zu erlangen. Außerdem ist es auch möglich das Paket im Controller weiterzuverarbeiten und zum Beispiel zu verschlüsseln oder zu komprimieren.
3. Verwerfen von Pakten, um zum Beispiel Angriffe abzuwehren.
4. Die Pakete an den normalen Layer 2 und Layer 3 Verarbeitungsprozesse des Switches weiterleiten. Dies ermöglicht es bei OpenFlow-Enabeld Switches nur unter bestimmten Umständen die Paketverarbeitung mit OpenFlow zu nutzen und ansonsten auf die reguläre Paketverarbeitung des Switches zurückzugreifen.

3.3 OpenFlow in der Praxis

Um den praktischen Nutzen von OpenFlow zu demonstrieren werden im Folgenden einige Anwendungsfälle in praktischen Szenarios genauer erläutert.

3.3.1 Experimente und Innovation in bestehenden Netzwerken

Forschung und Entwicklung in Netzwerken ist eine schwierige Herausforderung: Einerseits gehören die Netzwerke an Universitäten und in Unternehmen zu der geschäftskritischen Infrastruktur die auf keinen Fall gestört werden darf. Andererseits ist ein Fortschritt in der Netzwerkinfrastruktur nur möglich, wenn man mit neuen Technologien unter realen Bedienungen im echten, Netzwerk experimentiert und anschließend auf diese umstellt. Dazu gehören neue Routing-Protokolle oder der Umstieg auf IPv6. Des Weiteren bieten sich auch Stress-Tests oder Simulationen von Angriffen an, um das Verhalten des realen Netzwerks in diesen Situationen zu untersuchen.

Der Nutzen von OpenFlow für solche Zwecke lässt sich am besten an einem Beispiel aufzeigen: Ein Forscher einer Universität hat das fiktives Routing-Protokoll „X-OSPF“ entwickelt, das langfristig das Routing-Protokoll Open Shortest Path First (OSPF) in dem Universitätsnetz ersetzen soll. Der Forscher will das Protokoll nun von seinem Rechner aus in dem echten Netzwerk ausprobieren, ohne den regulären Netzwerkbetrieb zu beeinträchtigen. Dazu installiert er im OpenFlow Controller des Netzwerkes das „X-OSPF“-Protokoll und hinterlegt die Regel, alle Pakete die von der MAC Adresse seines Rechners stammen zunächst an den SDN Controller weiterzuleiten. Erreicht nun das erste Paket von seinem Rechner einen OpenFlow-Switch, wird das Paket an den Controller weitergeleitet. Der Controller verarbeitet das Paket und sendet die entsprechenden Regeln für das Routing in den Flow-Tables der Switche, die sich auf dem Pfad des Paketes befinden. Dadurch wird das „X-OSPF“-Protokoll nun nur auf die Pakete angewendet, die vom Rechner des Forschers stammen. Dieser Setup ermöglicht nun das Testen des „X-OSPF“-Protokolls im realen Netzwerk ohne den Produktivbetrieb zu beeinträchtigen. [16, S. 72]

Das Netzwerk der Stanford Universität ist inzwischen komplett auf OpenFlow umgestellt, um Experimente dieser Art zu ermöglichen. Dabei kommt, der für solche Zwecke speziell entwickelte, „FlowVisor“ zum Einsatz. Der FlowVisor ist ein spezieller OpenFlow Controller, der als Proxy zwischen den OpenFlow Switches und Controller fungiert und die Verteilung auf verschiedene Controller ermöglicht. Dadurch wird das Netzwerk in sogenannte „Slices“ unterteilt. Ein Slice ist das Produktivnetzwerk und die anderen Slices sind voneinander unabhängige und abgeschirmte Experimente. Jeder Slice hat seinen eignen OpenFlow Controller. Dadurch werden die einzelnen Experimente isoliert und es wird sichergestellt, dass die einzelnen Experimente das Produktivnetzwerk nicht stören können. [19, S. 129]

Beispiel für eines dieser „Slices“ an der Stanford Universität ist das „OpenPipes“-Experiment: Dabei werden Frames von Video-Streams im Netzwerk automatisch erkannt und mit einem bestimmten Filter versehen. Ein weiterer Versuch trägt den Namen „OpenRoads“: Man versucht eine nahtlose Übergabe (Hand Over) zwischen den WLAN und WIMAX Endpunkten der Universität zu ermöglichen. Dabei erkennt das Netzwerk schon im Voraus, wann es zu einer Übergabe kommen kann und leitet die bestehenden Verbindungen an den neuen Endpunkt um. [19, S. 130]

3.3.2 Load Balancing

Load Balancer haben die Aufgabe, den ankommenden Datenverkehr möglichst effektiv zu verteilen und dabei möglicherweise ausgefallene Server zu berücksichtigen. Normaler-

doi: 10.2312/NET-2013-02-1_09

weise werden dazu in regelmäßigen Abständen Anfragen an die einzelnen Server gesendet um deren Status zu prüfen. Die momentane Auslastung des Netzwerkes an sich wird dabei meistens nicht berücksichtigt.

Ein auf OpenFlow basierter und, ebenfalls an der Stanford Universität entwickelt und eingesetzter, Load-Balancer mit dem Namen „Plug-n-Serve“ hat das Ziel die Antwortzeit von HTTP Anfragen zu minimieren und neue Webserver dynamisch hinzuzufügen. Die Web Server sind verteilt im „Gates Computer Science“ Gebäude der Stanford Universität und es kommen verschiedene OpenFlow-Enabled Switche von Cisco, HP und NEC zum Einsatz.

Abbildung 4 zeigt die Architektur des „Plug-n-Serve“ Systems.

Flow Manager: Das Modul innerhalb des OpenFlow-Controllers das dafür sorgt, dass alle Pakete des gleichen Flows auf den gleichen Webserver umgeleitet werden. Dadurch bleiben Sessions bestehen.

Net Manager: Dieses Modul überwacht den Status des Netzwerkes. Dazu werden die OpenFlow Switche regelmäßig nach aktueller verfügbarer Bandbreite und Latenz befragt.

Host Manager: Der Host Manager überwacht die Auslastung der einzelnen Webserver. Außerdem werden neue Webserver automatisch erkannt und zum System hinzugefügt.

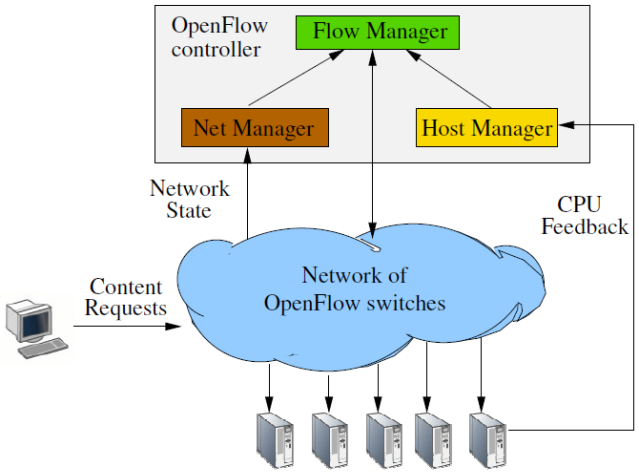


Abbildung 4: Plug-n-Serve Architektur [14, S.2]

Alle Webserver nutzen den gleichen IP-Alias. Wird vom OpenFlow Controller nun ein neuer Flow von einem Client an diese IP erkannt, wird ein passender Webserver aus dem Pool ausgewählt. Dabei werden alle im Controller verfügbaren Daten wie Latenz, verfügbarer Bandbreite, durchschnittliche Antwortzeit und Auslastung der einzelnen Webserver berücksichtigt. Der ausgewählte Webserver wird nun in den Flow-Tables der Switche hinterlegt und alle Pakete des Flows erreichen nun direkt den Webserver. [14]

3.3.3 Virtualisierung und Clouds

Cloud Computing und die Virtualisierung von IT Ressourcen ist inzwischen Standard und wird sicher auch die Zukunft weiter prägen. Dabei muss auch die darunterliegende Netzwerk Infrastruktur an die neuen Gegebenheiten angepasst werden. Um den Anforderungen des Cloud Computings gerecht zu werden, bietet sich eine Virtualisierung der Netzwerk Infrastruktur an:

Größere Datencenter umfassen über 100.000 Server, die je nach Hardware und Anforderungen in bis zu 20-30 virtuelle Hosts eingeteilt werden können. Dies resultiert in 2-3 Millionen MAC und IP-Adressen, dessen Verwaltung eine große Herausforderung ist. [15, S.672] Für die Netzwerk Infrastrukturen ist es wichtig, das Geschäftsmodell „Infrastructure as a Service“ (IaaS) umzusetzen: Das Netzwerk muss in Echtzeit umkonfigurierbar sein und skalieren können; virtuelle Hosts müssen zwischen physischen Hosts ohne Verwaltungsaufwand migriert werden können; Das virtualisierte Netzwerk muss in abgetrennte Subnetze für die einzelnen Nutzer unterteilbar sein. [15, S.672]

Auf der „IEEE CloudCom 2011“, einer internationalen Konferenz rund um Cloud Computing, wurde ein auf OpenFlow basierendes Konzept vorgestellt, dass diesen Aufgaben gerecht werden soll:

Eine Möglichkeit der Unterteilung bzw. Virtualisierung von Netzwerken sind VLANs (Virtual Local Area Networks). Jedes Paket wird mit einem spezifischen VLAN-Tag gekennzeichnet, um die Pakete dem virtuellen Teilnetzwerk zuzuordnen. Diese Methode hat zum einen den Nachteil, dass die Komplexität durch die Verwaltung des VLANs weiter erhöht wird und zum anderen wird durch das Mitsenden der VLAN Tags zusätzlicher Overhead erzeugt.

Die von dem Konzept bevorzugte Unterteilung in Teilnetzwerke erfolgt über die MAC Adressen auf Layer 2: Die MAC Adressen können nach dem IEEE 802.3 Standard lokal zugewiesen und verwaltet werden. Das ermöglicht es, die 48 Bits einer MAC Adresse in einen Host Anteil zur Identifizierung und einer Netz ID zur Einteilung in Subnetze (ähnlich dem VLAN Tag) zu unterteilen. [15, S.674]

Die MAC Adresse von virtuellen Maschinen wird normalerweise zufällig generiert (z.B. bei VMWare, Xen oder VirtualBox) und lässt sich beliebig ändern. Dies ermöglicht es VMs ohne großen Aufwand durch das Ändern des Netz Anteils der MAC Adresse in andere Teilnetzwerke zu verschieben. Da die 48 Bit MAC Adresse nach dem Ethernet Standard in jedem Fall übertragen wird, entsteht kein weiterer Overhead und auch die Verwaltung bleibt übersichtlich und einfach. [15, S.675]

OpenFlow unterstützt ab Version 1.1 auf MAC-Präfixen basierende Regeln. Dadurch kann der OpenFlow Controller anhand des MAC-Präfixes (der Netz ID) sicherstellen, dass der Netzwerkverkehr der einzelnen Teilnetzwerke isoliert wird. So werden z.B. Pakete an die MAC Broadcast Adresse nur an die virtuellen Maschinen mit der gleichen MAC-Präfix zugestellt. Außerdem ist es denkbar, für jedes Teilnetzwerk einen eigenen OpenFlow Controller mit eigenen Regeln einzurichten. [15, S.676]

doi: 10.2312/NET-2013-02-1_09

3.3.4 *Transparente Schaltung von Diensten*

Wedge Networks, eine Sicherheitsfirma aus den USA, hat sich mit der transparenten Einschiebung von zusätzlichen Diensten im Netzwerk mittels OpenFlow befasst. Konkret sollte ein Virenscanner, Anti-Spam System und ein Web Filter in ein Netzwerk integriert werden. Dabei sollten diese unabhängig von der Konfiguration der Clients und Server funktionieren.

Zur Umsetzung wurde ein OpenFlow Switch, ein OpenFlow Controller und ein Sicherheitsserver für den Virenscanner und die Filter verwendet. Im OpenFlow Controller wurden Regeln hinterlegt, die den Datenverkehr auf den Ports für HTTP, HTTPS, IMAP, POP3 und SMTP an den Sicherheitsserver umgeleitet haben. Zusätzlich wurden Regeln aufgesetzt, die die geprüften und gefilterten Datenströme anschließend wieder an das ursprüngliche Ziel weiterleiten.

Das Experiment funktioniere anschließend verlässlich und ohne nennenswerte Einbußen bei der Performance. Das Netzwerk wurde dadurch unabhängig von Server und Client-Konfiguration vor Viren geschützt. Ähnliche Anwendungszwecke zur Filterung, Monitoring und Absicherung des Datenverkehrs bieten sich beispielsweise in großen Campus Netzwerken. [8]

3.3.5 *Big Data Centers*

Das beste Beispiel für den Einsatz von OpenFlow bei riesigen Datenzentren ist Google. Dienste wie die Google Suche, Gmail, Google Maps oder YouTube tauschen täglich Unmengen von Daten zwischen den verschiedenen Google Datenzentren aus.

Googles Netzwerk ist in das Internet-Backbone, das den Traffic zu den Nutzern transportiert, und dem internen Backbone, das für den Traffic zwischen den Datenzentren verantwortlich ist, unterteilt.

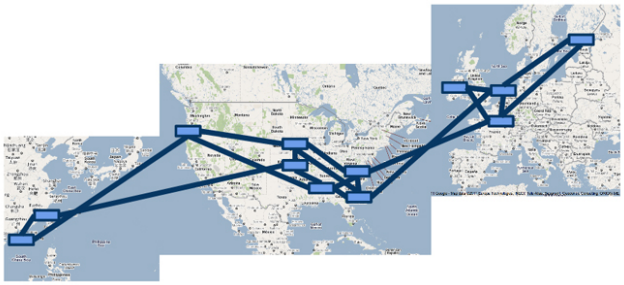


Abbildung 5: Googles internes Backbone [3]

Die Anforderungen für das interne Backbone von Google wurden mit steigendem Traffic immer größer: Der Datenverkehr muss zwischen den einzelnen Links intelligent verteilt werden, neue Dienste müssen schnell und einfach realisierbar sein und das Netzwerk muss auf ausfallende Links schnell reagieren [3].

Als Google deswegen im Jahr 2010 anfing, das interne Datencenter auf OpenFlow umzustellen, waren auf dem Markt kaum Geräte verfügbar, die den Anforderungen von Google gerecht wurden. Deswegen hat Google zunächst eigne Switche und Controller gebaut, die bis heute im Einsatz sind. Um eine hohe Fehlertoleranz zu gewährleisten, werden mehrere OpenFlow Controller eingesetzt. Zusätzlich nutzt Google einen zentralisierten „Traffic Engineering“ (TE) Dienst. Dieser sammelt die Daten von den OpenFlow Controllern und erhält so eine Gesamtsicht über das Netzwerk, um dynamisch Bandbreiten für die einzelnen Dienste zuteilen zu können. Inzwischen ist 95% des Google Netzwerkes auf OpenFlow umgestellt und trotz des großen Datenaufkommens kommt es sehr selten zu Engpässen oder gar Ausfällen. [5]

> „Das Software Defined Networking ist ein pragmatischer Denkansatz um die Komplexität und den Verwaltungsaufwand von Netzwerken zu reduzieren. Obwohl es immer noch zu früh ist um den abschließenden Erfolg zu verkünden, zeigen Googles Erfahrungen mit SDN und OpenFlow im Netzwerk, dass es bereit ist für den Einsatz in der realen Welt.“

(Urs Hölzle, Vizepräsident der technischen Infrastruktur von Google, Übersetzt aus dem Englischen [5])

3.4 Programmierung in SDN-Architekturen mit der Programmiersprache Frenetic

Die meisten OpenFlow Controller basieren auf dem Betriebssystem NOX, das für die Verteilung der verschiedenen Regeln an die einzelnen Switches verantwortlich ist. [11, S.1] Die Programmierung des OpenFlow Controllers ist aus verschiedenen Gründen eine Herausforderung:

- Der Controller enthält nur die Pakete bzw. Flows, für die noch keine Regel im Switch hinterlegt ist. Regeln müssen daher mit Bedacht installiert und gegebenenfalls auch wieder gelöscht werden.
- Die Regeln können sich, beispielsweise durch Wildcards, gegenseitig beeinflussen beziehungsweise überschreiben: Für die einzelnen Regeln müssen Prioritäten in den Flow Tables hinterlegt werden.
- Verschiedene Module, wie Routing oder Monitoring, müssen aufeinander abgestimmt sein, wenn sie auf die gleichen Flows zugreifen.

Ein Ansatz um die Programmierung von OpenFlow Controllern zu vereinfachen ist Frenetic. Frenetic verwendet eine Abstraktion, die eine paketbasierte Programmierung erlaubt. Bei Frenetic handelt es sich um eine höhere Programmiersprache, die auf den Ideen der funktionalen Programmierung basiert. Die verschiedenen Regeln werden dann von Frenetic zur Laufzeit erzeugt und auf den Switches und Controller verteilt. Technisch handelt es sich bei Frenetic um eine Sammlung von Python Bibliotheken. [11, S.1]

doi: 10.2312/NET-2013-02-1_09

Die Funktionsweise von Frenetic wird im Folgenden an einigen Beispielen demonstriert.

Der Frenetic Programmcode in Listing 1 implementiert die einfache Funktion eines Switches, alle Pakete die den Switch an einem Port erreichen an den anderen Port weiterzuleiten.

Listing 1: Einfacher Repater

```
def switch_join(switch):
    # Repeat Port 1 to Port 2
    p1 = {in_port:1}
    a1 = [forward(2)]
    install(switch, p1, DEFAULT, a1)
    # Repeat Port 2 to Port 1
    p2 = {in_port:2}
    a2 = [forward(1)]
    install(switch, p2, DEFAULT, a2)
```

Wenn ein neuer Switch an das Netzwerk angeschlossen wird, wird die Funktion `switch_join` aufgerufen und zwei Regeln installiert: Die Variablen `p1` und `p2` beschreiben die Muster (Patterns), wann die einzelnen Regeln angewendet werden sollen. In diesem Fall das Eintreffen von Paketen an Port 1 bzw. 2. `a1` und `a2` stehen für die darauf folgenden Aktionen (Actions) zum Weiterleiten der Pakete an den jeweils anderen Port. Mit `install` werden die Regeln in der Flow Table das Switches hinterlegt. Der dritte Parameter, in diesem Fall `DEFAULT`, beschreibt die Priorität der Regel. [11, S.2] [18, S.5]

Eine erweitertes Programm in Listing 2 soll den kompletten Traffic auf Port 80 beobachten.

Listing 2: Überwachung von Port 80

```
def switch_join(switch)):
    # Web traffic from Internet
    p = {inport:2,tp_src:80}
    install(switch, p, DEFAULT, [])
    query_stats(switch, p)
def stats_in(switch, p, bytes):
    print bytes
    sleep(30)
    query_stats(switch, p)
```

Zunächst wird eine Regel installiert, die bei eingehenden Traffic an Port 2 des Switches und TCP Port 80 ausgeführt wird. Die Funktion `query_stats` zählt den Traffic in Bytes und die Funktion `stats_in` gibt die übertragene Datenmenge alle 30 Sekunden aus [11, S.2] [18, S.6].

3.5 Kritische Betrachtung und Ausblick

OpenFlow und SDN wird oft als die Zukunft der Routing Protokolle angepriesen. Klassische Routing Protokolle stimmen sich nicht aufeinander ab, sie berücksichtigen nicht die Auslastung des Netzwerkes, sie sind statisch und können nicht in Echtzeit verändert und kontrolliert werden. All diese Nachteile können durch ein OpenFlow basiertes Netzwerk ausgeräumt werden. Trotzdem sollte man nicht unberücksichtigt lassen, dass sich die klassischen Routing Protokolle in den letzten 20 Jahren bewährt haben: sie sind verlässlich, geprüft, selbst-heilend, autonom und skalierbar. [17, S.8]

Bei OpenFlow handelt es sich trotz allem um Software, welche Fehler enthalten kann, anfällig ist oder fehlerhaft programmiert werden kann. Man sollte daher sehr sorgfältig abwägen, ob OpenFlow im spezifischen Anwendungsfall wirklich sinnvoll ist und auch einen Mehrwert bietet. Obwohl es viele erfolgreiche experimentelle und durchaus marktreife Projekte mit OpenFlow gibt, hört man recht wenig von der praktischen Nutzung von OpenFlow Netzwerken in der freien Wirtschaft. Zum einem ist die Skepsis groß, Teile der kritischen Netzwerkinfrastruktur Software anzuvertrauen. Zum Anderem lassen sich viele Probleme bereits durch bewährte Techniken wie beispielsweise VLANs oder QoS lösen. Wirklich bewährt hat sich OpenFlow bisher nur bei riesigen Hostern und Universitäten mit speziellen Anwendungszwecken - oder eben bei Google.

Nicht unbedeutend für die Zukunft von SDN und OpenFlow ist die Unterstützung und die Integration von den Hardware Herstellern. Diese läuft relativ schleppend, obwohl die meisten Hersteller inzwischen OpenFlow unterstützen. Oft wird nur die alte OpenFlow Version 1.0 unterstützt. Dies liegt daran, dass die Hersteller wie Cisco nur ungern Software auf ihren Geräten erlauben, die die eigene Software verdrängen könnte. Es besteht die berechtigte Sorge, dass die bisher teuer verkauften Produkte durch günstige OpenFlow Switches mit Open Source Software ersetzt werden. Cisco entwickelt zudem ihre eigenen SDN Lösungen („Cisco One“), bei denen sie an den Lizenzen verdienen möchten [13]. Diese Blockadehaltung gegenüber OpenFlow aber durch den Konkurrenzdruck zwangsläufig weiter aufweichen.

Auf längere Sicht wird sich die Virtualisierung von Netzwerken und damit das Software Defined Networking sicher weiter durchsetzen. Doch ähnlich wie bei der Virtualisierung von Server Infrastrukturen wird dieser ein langjähriger Prozess sein und sehr langsam voranschreiten. Im Gartner Hype Cycle 2012 für Netzwerk und Kommunikation wird SDN und OpenFlow noch im Bereich „Technology Trigger“ kurz vor dem Übergang zum Gipfel der überzogenen Erwartungen („Peak of Inflated Expectations“) angesiedelt und mit einer Marktreife in 5 bis 10 Jahren eingeordnet. [10]

4. LITERATUR

[1] About openflow. `http://www.openflow.org/wp/about/`, September 2012.

[2] Cisco open network environment. `http://www.cisco.com/web/solutions/trends/open_network_environment/indepth.html`, Oktober 2012.

[3] Going with the flow: Google's secret switch to the next wave of networking. `http://www.wired.com/wiredenterprise/2012/04/going-with-the-flow-google/all/1`, November 2012.

[4] Hp simplifies networking with broadest choice of openflow-enabled switches. `http://www.hp.com/hpinfo/newsroom/press/2012/120202a.html`, Oktober 2012.

[5] Inter-datacenter wan with centralized te using sdn and openflow. `https://www.opennetworking.org/images/stories/downloads/misc/googlesdn.pdf`, Oktober 2012.

doi: 10.2312/NET-2013-02-1_09

[6] Openflow wiki. http://www.openflow.org/wk/index.php/Main_Page, September 2012.

[7] Openflow – next-generation networking for a smarter planet. https://www-304.ibm.com/connections/blogs/VMstg/tags/openflow?lang=en_us, November 2012.

[8] Wedge networks whitepaper: Transparent service insertion in sdns using openflow, September 2012.

[9] C. Berndtson. Open networking foundation: About. https://www.opennetworking.org/about/, September 2012.

[10] N. R. Bjarne Munch, Katja Ruud. Hype cycle for networking and communications, 2012. *Gartner*, Juli 2012.

[11] N. Fostery, R. Harrison, M. L. Meola, M. J. Freedman, J. Rexford, and D. Walker. Frenetic: a high-level language for openflow networks. *PRESTO '10 Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow*, (6), 2010.

[12] O. N. Foundation. White paper: Software-defined networking: The new norm for networks, April 2012.

[13] M. Fratto. Prediction: Openflow is dead by 2014; sdn reborn in network management, Mai 2012.

[14] N. Handigol, S. Seetharaman, N. McKeown, and R. Johari. Plug-n-serve: Load-balancing web traffic using openflow. *ACM SIGCOMM Demo*, August 2009.

[15] J. Matias, E. Jacob, D. Sanchez, and Y. Demchenko. An openflow based network virtualization framework for the cloud. *Cloud Computing Technology and Science (CloudCom)*, pages 672–678, December 2011.

[16] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, April 2008.

[17] I. Pepelnjak. Openflow and sdn: hype, useful tools or panacea?, 2012.

[18] J. Rexford. Frenetic: A programming language for openflow networks. ttp://www.frenetic-lang.org/, Oktober 2012.

[19] R. Sherwood, G. Gibb, and M. Kobayashi. Carving research slices out of your production networks with openflow. *ACM SIGCOMM Computer Communication Review*, 40(1):129–130, January 2010.

[20] J. K. Waters. Virtualization definition and solutions, März 2007.

doi: 10.2312/NET-2013-02-1_09

Resource Management Tools

Anh Nguyen
Supervisor: Corinna Schmitt
Seminar: Innovative Internet Technologies and Mobile Communication WS 12/13
Chair for Computer Network Architectures and Services
Department for Computer Science, Technische Universität München
Email: anh.nguyen@tum.de

ABSTRACT
Sensor networks perform and are used in many different kinds of fields, but are especially given tasks, which are difficult and almost impossible for a human to do. Besides monitoring the environment and collecting necessary information, sensor networks have to manage and supply themselves with essential resources to perform the tasks that were allocated to the whole network.
This paper focuses on resource management and the development of effective resource management tools, which has to face many challenges due to resource limitations, such as energy consumption and scalability. Peloton and Tiny Network Manager cover up most of these challenges with different approaches. While Peloton uses a ticket abstraction to distribute allocations and perform optimization on the whole sensor network process, the Tiny Network Manager has an implemented "request-reply-mechanism" to manage its assignments and resources.

Keywords
Wireless sensor network, resource constraints, resource management tools, Peloton, Tiny Network Manager

1. INTRODUCTION
Sensor networks have been developed from innovations in wireless communication within the past few years and are one of the most important technologies in the 21st century. Nowadays sensor networks are used in military for communication and targeting, in health systems for monitoring and assisting (disabled) patients, and other application fields, such as managing inventory in business, and monitoring inaccessable and disaster areas.[1]

Figure 1 shows an example of a sensor node structure with its units and their connection to each other: The basis of the sensor node is its power unit, which keeps the sensor node and its units alive. The overall task of a sensor network is to monitor physical or environmental conditions through the sensing unit, which monitors the environment and converts the analog signals to digital ones, and finally passes it on to the processing unit. After performing quick local data processing, all the collected data will be passed through the network and routed back to the sink with a multi-hop architecture.[1]

Since sensor nodes are often deployed in inaccessable or disaster areas, it is common that they have to move their position from time to time due to unconventional events and

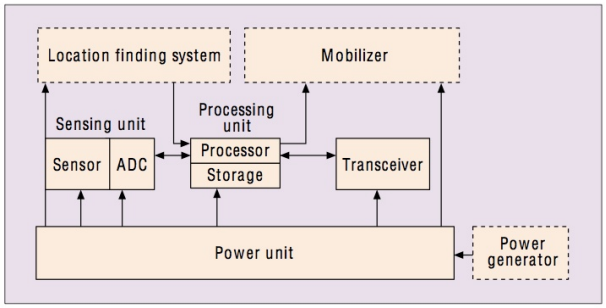


Figure 1: Sensor node structure [1]

situations. Therefore a location finding system and a mobilizer are often implemented as additional units.[1][2]

Wireless sensor networks are intended to perform for "long periods of time, over relatively large physical spaces, and in places that are difficult for people to reach."[2] Since sensor nodes in wireless sensor networks lack in terms of e.g. energy, operating systems and tools were developed to manage exhausting resources and constraints of the sensor nodes. Those tools are called resource management tool and help managing and distributing the resources to all the sensor nodes in a network.[1][3]

In the following section definitions are given on resource management and resource management tools and the challenges during the development of resource management tools in general, as well as the chosen resource management tools Peloton and Tiny Network Manager are described afterwards. In comparison of those resource management tools in section 3, their different and common features are characterized. Lastly a conclusion is given and a future outlook for sensor networks and resource management tools in technology is presented.

2. CHALLENGES DURING THE DEVELOPMENT OF RESOURCE MANAGEMENT TOOLS

2.1 Resource management functions
According to Tenrox, resource management is defined as an "efficient and effective deployment and allocation of an organization's resources when and where they are needed."[11] In terms of sensor networks it is about how to save the necessary resources and optimize their consumption for longer

doi: 10.2312/NET-2013-02-1_10

availability. Moreover resource management is not only about lowering the resource consumption, but also to optimize the communication between each node in the sensor node to distribute tasks and thereby the resource allocations effectively. The result of an efficient management process would be optimized and distributed allocation of resources, so that each single node is not overloaded with assignments, but is still kept busy. It is an efficient employment of sensor nodes to effectively perform and succeed their assigned tasks.[11]

2.2 Challenges and difficulties

Sensor networks, on the one hand, usually have stable sensor nodes that can withstand the harsh and uncertain environment and avoid physical damage, but lack in essential resources on the other hand. Moreover their disadvantage is that each sensor node can only collect and process data, but not communicate and share the collected information and also information on their node state with other nodes. Therefore resource management tools are used to compensate this and optimize all processes in the whole sensor network.[3]

The challenges in developing a resource management tool therefore is to implement efficient processes and algorithms to manage scarce resources of the sensor nodes, such as the minimal life-span, limited memory and stack space and limited radio bandwidth[1][2][5]:

- Minimal life-span
  Power consumptions of each sensing node have to be kept low, since energy is the scarcest resource of a sensor node. The resource management tools therefore aim for the optimization of power usage and with that also the maximization of a node's lifetime, so that sensor nodes are able to work for a long time and are prevented from a failure or breakdown in an early state.[2][9]

- Limited memory and stack space
  The memory and stack space of a sensor is limited and as a consequence the amount of data collection and processing as well as the amount of data transfers are also limited. If too much data is processed and transferred it will result in a stack overflow, causing the process or the node to crash at the worst. Therefore the management tool has to define and rank the data in terms of importance and consider which data to process first or which data to drop.[3]

- Limited radio bandwidth
  Besides having limited stack space, the bandwidth of wireless sensor networks is also limited and much lower than that of a wired sensor network. In this case the resource management tool has also to decide which data to transfer or which data to keep in the queue.[2]

Besides managing sensor node resources, the implemented processes have also to meet the quality standards through ensuring fault tolerance, the scalability of the whole system as well as real-time performances and the reliability of the collected and processed data, which also have a strong impact on the whole sensor network process[1][2][5]:

- Fault tolerance
  The term fault tolerance refers to the reliability of sensor nodes. In case of failure or a breakdown of single nodes due to lack of power, interruption or damage, the resource management tool has to ensure that the task of the broken sensor node and the overall task and functionality of the whole network are not be affected.[1]

- Scalability
  Sensor networks are composed of a huge amount of sensor nodes. Depending on the usage its scale can reach up to thousands or millions of sensor nodes. Since sensor nodes are not able to think and work autonomously, the resource management tool has to coordinate the communication and interaction of sensor nodes and other hardware components to perform tasks faster and more effective, and maintain a manageable structure of the sensor node.[1]

- Real-time performance
  Many sensor network applications have to stick to a certain time schedule to interact with other sensor nodes and hardware components. Moreover the assigned tasks have to be completed until a certain deadline. The resource management tool has to ensure real-time behavior through efficient management of the tasks and sensor nodes also in regard of the available resources of each sensor node. Constructing and implementing an algorithm to ensure real-time performance would be quite a challenge because of the dynamic environment.[3]

- Reliability
  The term reliability in this case refers to the collected information. Since the sensor networks are used to collect important data and information, the resource management tool has to make sure that the processing of collecting necessary information is always reliable and transferred to the correct places. Losing important information would lengthen the whole sensor network process duration or in the worst falsify the statistics like incorrect information.[1]

If we want to increase the functional performance of a wireless sensor network, we have to intelligently manage not only the limited resources, but also the interaction between each sensor node in the system.[5]

Until now a couple of resource management tools, such as TinyOS, Pixie, SOS and Eon, have been developed: They only focus on managing resources for individual nodes, although the coordination of resource management decisions across the whole network would be more efficient and accurate.[4]
This approach is not advisable, because the nodes in a network always have to interact with other nodes and hardware components. The sole management of a sensor network as a collection of independent sensor nodes would not solve the problems of limited resources and network constraints. On the contrary: It is necessary that nodes share information on their local state and communicate and collaborate with other nodes to assign tasks to achieve the most efficient use of resources and task processing.[2][5]

doi: 10.2312/NET-2013-02-1_10

As a result, the final tool should be able to support and enable "low latency, energy-efficient operation, built-in autonomy and survivability, and low probability of detection of operation"[3].

3. RESOURCE MANAGEMENT TOOLS

3.1 Peloton

One of those resource management tools is called Peloton and was developed in 2009 as a "new distributed sensor OS".[4] Peloton builds up on Pixie, a node-level operating system for sensor nodes to manage and control over resource availability: Pixie works with resource tickets, which are sent by and to the operating system. A ticket represents a time-bounded right to consume a certain amount of a resource and is a "flexible currency for resource management within the node".[4] In this way, Pixie can respond to the resource ticket allocations and distribute available resources.

Peloton has borrowed the Pixie's idea of using resource tickets and extended it to a new ticket abstraction, which has implemented resource management mechanism called vector tickets, distributed ticket agents and state sharing. But unlike Pixie and also other operating systems, such as TinyOS and EON, which manage the resources on individual nodes only, Peloton is not focusing on individual nodes but the cooperation of all sensor nodes for resource management.[4]

3.1.1 Ticket Abstraction

It has been already mentioned that Peloton has implemented its own ticket abstraction through extending Pixie's resource ticket mechanism. It consists of the vector tickets, distributed ticket agents and a state sharing system:

Vector Ticket is a programming abstraction, which represents the right of a sensor node to consume resources for performing operation. A vector ticket V consists of resource tickets T_i. Each vector ticket is displayed as a tuple (n, R, c, t_e) that represents the "time-bounded right to consume up to c units of resource R until expiry time t_e at node n"[4]. Moreover it can capture the complete resource allocation of an operation of several nodes and is also used for tracking and controlling this allocation in the network. Therefore vector tickets record the consumed resources of the sensor nodes and provide feedback to the application in terms of resource availability and usage. One strength of the vector tickets it that they can be decoupled if the allocation is not necessary anymore and give resources free for other needy nodes.[4]

The **distributed ticket agent** mechanism permits resource management decisions all across nodes, clusters and the network as a whole. They manage vector tickets, track resource availability of single nodes as well as of a set of nodes and distribute resource allocations and also perform resource management policies across all the nodes. Because of distributed ticket agents and their functions, resource allocations can be performed individually by nodes, collectively by a group of nodes, or globally by a base station. In the resource allocation process a node must either acquire a vector ticket from the node's local ticket agent or from a third-party agent, such as the base station or another node in the network, to get a resource allocation. But even if a ticket is acquired, it is possible that it may be revoked before its expiry time, because of changing conditions.[4]

The **state sharing** mechanism is in charge of sharing node states on local resource availability so that efficient coordination among all the sensor nodes is possible. For this efficient coordination among all the sensor nodes in the network, Peloton has build-in mechanisms for node sharing within the neighborhoods, clusters and across the whole network. Peloton's API makes it possible for nodes to share information on local resource availability into a shared tuple state. This information on the other hand can be called up from the shared tuple space to make it readable for each other node. Those data are refreshed frequently, but only between nodes, that are direct neighbors. Information from distance nodes is refreshed less often, but it is always possible to request a direct update from another node to obtain its latest state. Those state sharing operations consume energy and bandwidth and thus, also requite resource tickets.[4]

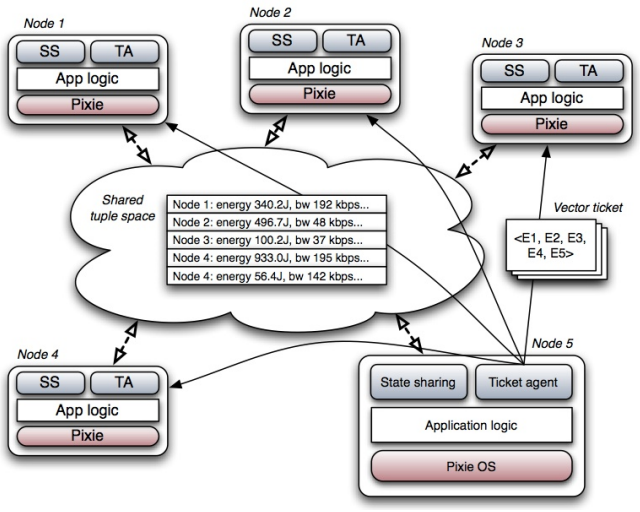


Figure 2: Peloton network with five nodes[4]

Figure 2 shows a Peloton network with five sensor nodes. Each node has the same structure: They are build up on the Pixie operation system and extended with an application logic and the ticket agent and state sharing mechanism, that enable the nodes to make resource requests and send all their vector tickets to other sensor nodes. Moreover all of the nodes are linked to the shared tuple space through state sharing and can provide and access information on the local node state of each node in the network.[4]

3.1.2 Resource Management

Besides the ticket abstraction, which supports the distribution of resource allocations in the sensor network and optimizes the overall sensor network process, Peloton has implemented adaptive cluster-based routing, duty cycling and reliable data collection as further resource management mechanism[4]:

Adaptive cluster-based routing is a frequently used method for energy-efficient routing in sensor networks: One routing and communication protocol for wireless sensor networks is

doi: 10.2312/NET-2013-02-1_10

called “LEACH” and stands for “Low Energy Adaptive Clustering Hierarchy”.[6]
LEACH works with cluster architecture and its main idea is to collect data from distributed sensors and transmit it to a base station. In LEACH some nodes can elect themselves as clusterheads. As clusterheads they are responsible for receiving data and data packets from other clusterheads and forwarding it to members of its cluster or to the base station. Since the data transmission to the base station requires and consumes too much energy the clusterheads “rotate”, which means that the clusterheads take turns in being the clusterhead. Through this rotation huge energy consumption can be avoided and a longer lifetime of the sensor node can be ensured. Moreover the clusterhead selection can be done on local information, which does not need a communication with the base station and other entities outside the network and therefore reduces the energy consumption as well.[6]
Peloton has an implemented variant of LEACH, in which a sensor node can elect itself as a clusterhead based on the energy consumption profile of neighboring nodes. Each clusterhead temporarily becomes the ticket agent for the cluster members. In this position it can assign vector tickets to manage the overall communication and resource consumption of the cluster members.[4]

Duty cycling the one of the most common forms of resource management.[4] The idea of duty cycling is based on the assumption that nodes are not used for communication and transmission issues all of the time. With this assumption, Peloton has build in mechanism that allows sensor nodes to change their active state into an idle one and vice versa to save up energy consumption. Therefore the sensor nodes are only used when it is necessary. It is often difficult to determine an appropriate duty-cycling schedule statically as the implementation and usage of duty cycling can affect data fidelity, network connectivity and also sensor coverage. But with a precise knowledge about the network topology and transmission structure, it is possible to coordinate and determine the sensor node schedules over the whole network.[4][7]

Another resource management function that is enabled in Peloton is the **reliable data collection** of high data-rate signals.[4]
Reliable transfer requires substantial bandwidth and sufficient energy resources. Sensor networks lack in both bandwidth and power in the attempt to acquire high data-rate and high-fidelity signals throughout the whole network. As a result only the most “interesting” or most “prominent” signals will be acquired, while the rather “uninteresting” signals are left out. Therefore the optimization of high data-rate signal collections would benefit the resource management in sensor networks.[4][8]
One approach to do so is to enable clusterheads in the network in a similar way as in LEACH: Clusterheads can manage the stored data of neighboring nodes and perform local optimization to rank the importance of the signals. The signals with the highest ranking or priority should be provided with energy and bandwidth resources. With the coordination of the clusterheads, the transfer of the high data-rate signals can be managed and scheduled to the other nodes or to the base station. This approach of reliable data collection therefore allows a reliable transfer of high data-rate signals without consuming much energy and causing a communication overhead. The maximization of reliability of sensors maximizes the reliability of the completion of the assigned tasks, and thus leads to an overall solid performance.[4]

With all those resource management mechanism, Peloton is able to efficiently distribute resources within the network. The combination of Peloton’s reliable data collection with the node energy schedule through duty cycling is not perfect yet, but make a good basis for the development of collaborative applications, which handle changing node states.[4]

3.2 Tiny Network Manager

Another approach for optimizing resource consumption and high performance is called Tiny Network Manager. Tiny Network Manager, short TNM, is a resource management tool that is developed in 2010, based on devisable management, which is a kind of “autonomous management, where different network managers detect network events and do the necessary tasks based on network resources, predefined policies, intuition and intelligence”.[5]

Two different kinds of Tiny Network Manager applications exist:
If Tiny Network Manager is used in large-scale sensor networks, i.e. a sensor network with a huge amount of sensor nodes, the software will reside and control the resources from the clusterhead nodes by collecting and analyzing information from the cluster members, whereas the Tiny Network Managers of the inner cluster members communicate each other to handle uncertain events.
If Tiny Network Manager, on the contrary, is used in a flat wireless sensor network architecture, the software will manage the resources from a base station through the Internet to handle uncertain events[5].

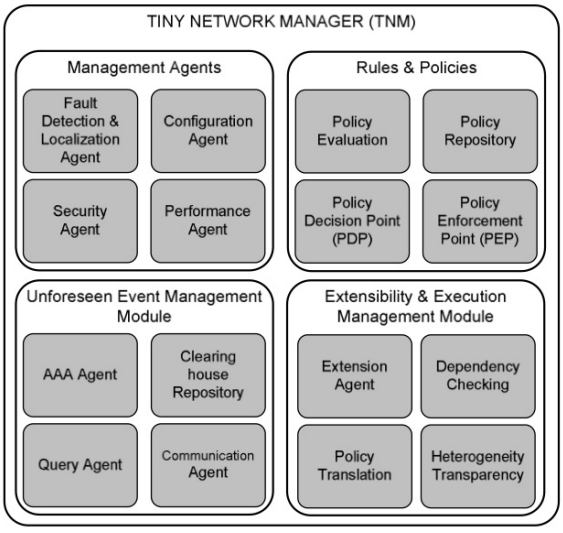


Figure 3: Tiny Network Manager modules[5]

In Figure 3 the Tiny Network Manager structure is displayed, which is composed of the four modules Management Agents, Rules and Policies, Unforeseen Event Management and Extendibility and Implementers, whereas each of them has its own additional components as well:

doi: 10.2312/NET-2013-02-1_10

Management Agents are in charge of detecting changes in the node collection and have specific management tasks such as configuration management, fault management and performance management.

The module **Rules and Policies** and its sub-modules provide policy based management, which makes autonomous management possible.

Unforeseen Event Management modules are trying to look for a solution if specific network state changes come up, which are impossible to handle with the existing rules and policies.

The **Extendibility Module** manages all management policies, agents and functions and can include, remove and modify them if it applies.

3.2.1 Entities

In the resource management process, Tiny Network Manager does not work on its on but with the collaboration of the following entities:

- Network Manager
- Clearing House
- Resource Manager

The **Network Manager**
is an entity that works at the WSN gateway and is composed of the TNM and the Resource Management Module.[5] The TNM performs its usual tasks, which are already described and explained in the previous chapter. The Resource Management Module carries out three main tasks - monitoring, managing and resolving – that are supported by their respective modules and sub-modules, which can be seen in Figure 4:

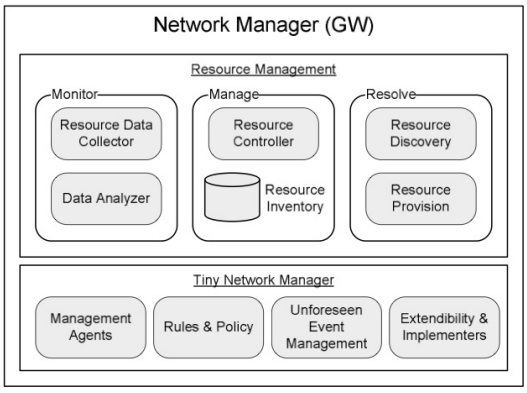


Figure 4: Network Manager[5]

- The monitoring module is composed of the Data Collector module, which collects resource information from the sensor node, and the Data Analyzer, which measures and evaluates the resource information.
- The managing module consists of its main maintenance entity Resource Controller and the database Resource Inventory, which stores all the collected resource information.
- The resolving module is in charge of finding new resources or alternative ones through Resource Discovery and implements it through the Resource Provision sub-module.

With the AAA Agent in the TNM module, the Network Manager can maintain secure sessions to the Clearing House and the Resource Manager.[5]

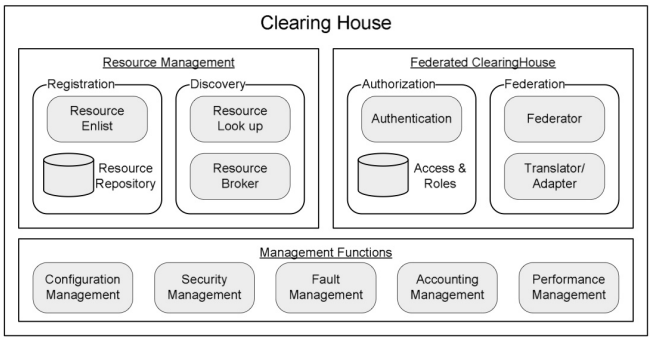


Figure 5: Network Manager[5]

The **Clearing House** (CH) works in the sensor network and consists of a Resource Management Module, the Federated Clearing House and Management Functions. It manages all the information about resources and can provide the location and connection mechanism of registered resources if it is requested.[5] Figure 5 shows the structure of the CH: Same as the Network Manager it has a module to store and maintain all resource information in a database. The Discovery module is the main entity of the CH and handles all the resource requests, whereas the module Management Functions is used for internal management and assistance. The Federated Clearing House is responsible for maintaining all Clearing Houses and providing authorization for communications between CH-CH, CH-RM, and CH-Gateway (Network Manager).[5]

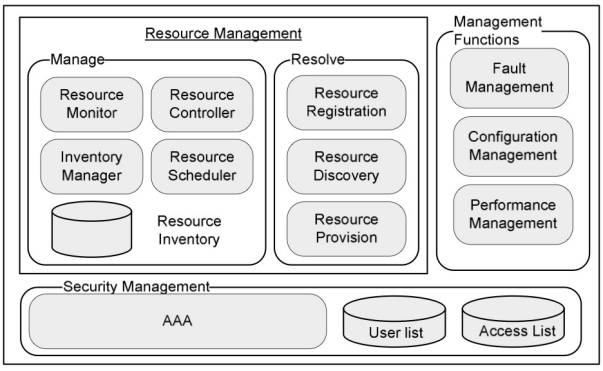


Figure 6: Resource Manager[5]

doi: 10.2312/NET-2013-02-1_10

The **Resource Manager** (Figure 6) is the last entity, which collaborates in the TNM process, and is in charge of resource management in the sensor network.[5]
It has two main modules, Manage and Resolve, and its submodules, which perform the management process that will be described in the further paper. Moreover it provides the same Management Functions module as the CH and a Security Enabler, which enables the interaction between CH-RM, and CH-Gateway.[5]

3.2.2 Resource Management

The Resource Manager performs the resource management process in collaboration with TNM, the Network Manager and the Clearing House: The Resource Management module as well as the Data Collector and the Analyzer monitor the resources. If a resource's performance is decreasing, the Resource Controller has to look for an alternative resource. If no solution can be found, then the Request-Reply-Mechanism will be initiated, which works in the following way[5]:

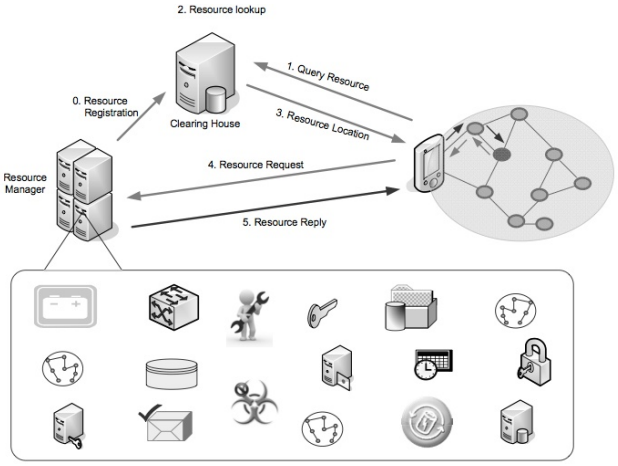


Figure 7: Request-Reply-Mechanism[5]

1. The Resource Discovery module establishes a connection to the Clearing House and sends a resource request

2. At the Clearing House, the Authorization module hosts this request establishes a new session

3. After the session is established successfully, the Authorization module sends the request to the Resource Broker, which looks up the required resource in the database

4. If the required resource is found in the resource database, the resource information is transferred to the gateway

5. After the resource information is collected at the gateway, another request is sent to the Resource Controller of the Resource Manager

6. Resource Discovery and Resource Scheduler take care of this request by looking in up and performing the scheduler if several requests are made at the same time

7. The Resource Discovery and Scheduler take care of this request and perform the scheduler, if several requests arrive in the same period of time

8. The Extensibility module is in charge of implementing the resource in the network after its arrival at the gateways

3.3 Comparison

Peloton and Tiny Network Manager both provide sensor networks a collection of functionalities to optimize the consumption of scarce resources and manage task assignments throughout the whole network. The comparison of Peloton and TNM in terms of energy, fault tolerance, reliability, scalability, real-time behavior, memory and autonomy, which can be seen in Table 1, show that both tools cover up most of the network constraints. The provided functions, however, vary in effectiveness and efficiency.

	Peloton	TNM
Energy	Cluster-Based Routing, Duty Cycling	Cluster-Architecture
Fault Tolerance	State Sharing, Decoupling allocations	Unforeseen Events, Management Agents
Reliability	Reliable Data Collection	Rules and Policies, Modules
Scalability	Ticket Abstraction, Duty Cycling	Modules (NW, CH, RM, TNM)
Real-Time Behavior	Ticket Abstraction	Resource Scheduler
Memory	Reliable Data, Cluster Routing	-
Autonomy	-	Devisable Management

Table 1: Comparison of Peloton and TNM

Energy:
Both management tools work with a cluster-architecture to gain low energy consumption and optimize the workflow between each sensor node. Peloton might be more effective since its cluster-architecture also enables cluster-based routing, in which cluster-heads can rotate and take turns to submit high-energy tasks. Moreover it has also implemented the Duty Cycling mechanism, which enables sensor node to switch their active state into an idle one.

Fault tolerance:
In terms of fault tolerance, Peloton provides the State Sharing mechanism, which shares all the information on a node's current information that allows other sensor nodes to react to a breakdown or failure of a sensor node. Moreover resource allocations can be decoupled in Peloton's ticket abstraction, which means that allocations to a broken sensor node can be revoked and transferred to another sensor node. Tiny Network Manager reacts and handles a node's breakdown in a more efficient way: The Management Agent is in charge of configuration, fault and performance management, on basis of the defined Rules and Policies, and supports the

doi: 10.2312/NET-2013-02-1_10

maintenance of the network, also in case of a node's breakdown. When there is no rule defined for a certain situation, the module Unforeseen Event Management is responsible for handling it.

Scalability:
Besides the cluster-architecture, which divides all the sensor nodes into cluster-groups and therefore supports the scalability of a large wireless sensor network, TNM and Peloton have further functions to make a sensor network scalable: The TNM abstraction contains many modules for resource management of sensor networks. Those modules have certain assigned tasks to monitor and manage sensor nodes, and providing resources and their information to all hardware components. This approach is supporting the scalability of a large-scale sensor network.[5]
Peloton has an implemented ticket abstraction, which implies that every single action in the sensor network requires a vector ticket and is managed by a ticket agent. This ticket abstraction enables tracking sensor nodes and resource allocations, and is therefore very useful to make a network more scalable. Moreover Peloton's Duty Cycling mechanism also supports a sensor network's scalability, since only tasking sensor nodes are active.

Reliabilty:
The Reliable Data Collection from Peloton makes sure that the whole collection process is done correctly and that important data is prioritized and highly ranked for transmission.
TNM is supported by Rules and Policies that are in charge of the accurate tasks assignments and workflow. The reliability and correctness of the functions and assignments are supported by TNM's modularized system: Certain data and functions are only available in certain modules, which provide a distributed system and simple architecture that prevents confusion and data lost.

Real-Time Behavior:
It is difficult to decide on the better tool in terms of real-time performance. Peloton's vector ticket limits the resource consumption to an expiry time t_e, which is managed by the distributed ticket agent in addition. TNM's Resource Scheduler, on the other hand, decides on the time schedule of all resource requests and allocations.

Memory:
Sensor nodes have limited memory space, that means that only a limited amount of information can be stored or processed in a node, else a stack overflow will be caused. Peloton possesses the resource management function Reliable Data Collection, which optimizes high data-rate signal collections. This process has the result that not only energy and bandwidth consumptions will be reduced, but also the amount of information that has to be stored in a node. Moreover the sensor nodes can also take turns in collecting the information through the rotation of the cluster-members enabled through the Cluster-Routing.

Autonomy:
TNM provides the architecture and design for autonomous resource management, since its development was based on devisable management. Therefore TNM is able to work autonomously and do the necessary tasks based on network resources, predefined policies, intuition and intelligence. Peloton on the contrary doesn't provide intelligent autonomy.

4. CONCLUSION AND FUTURE OUTLOOK

Overall it is difficult to decide on the better tool out of those two. It depends on the purpose of its usage:
Peloton should be used in sensor networks, which want to enable low energy consumption, reliable data collection, scalability, real-time performance and low memory consumption.
TNM on the other hand should be used in sensor networks, which require high fault tolerance, reliable data collection, real-time performance and autonomy.

The future development of resource management tools does not only depend on optimization algorithm and functions, but also on the development of the sensor networks in general. Throughout the past few years, the technology of sensor network has achieved enormous progress, as it has become more important day by day.

	1980's - 1990's	2000 - 2003	2010
Manufacturer	Custom contractors, e.g. for TRSS	Commercial: Crossbow Technology Inc., Sensoria Corp., Ember Corp.	Dust, Inc. and others to be formed
Size	Large shoe box and up	Pack of cards to small shoe box	Dust particle
Weight	Kilograms	Grams	Negligible
Node architecture	Separate sensing, processing and communication	Integrated sensing, processing and communication	Integrated sensing, processing and communication
Topology	Point-to-point, star	Client server, peer to peer	Peer to peer
Power supply lifetime	Large batteries; hours, days and longer	AA batteries; days to weeks	Solar, months to years
Deployment	Vehicle-placed, airdrop single sensors	Hand-emplaced	Embedded, "sprinkled" left-behind

Table 2: Comparison of Peloton and TNM

The functionalities of the resource management have to adapt to the changes of the sensor network and its capabilities. Table 2 shows that sensor networks became smaller in its evolution and have already reached a ridiculous size of dust particles.[2]

In terms of optimizing resource constraints, Peloton and Tiny Network Manager have shown us first approaches on how to manage them. Even though these approaches can provide efficient resource management, none of them can completely address and handle with uncertainty, which is inevitable in dynamic networks.
All network constraints cannot be resolved, no matter how

doi: 10.2312/NET-2013-02-1_10

much resourceful a sensor network gets, since unforeseen events cannot be predicted and measured.For this reason, it is always important to keep in mind, which functionalities this resource management tool has to enable and which not.[5][10]

Peloton's and TNM's approaches carefully implemented on a case-by-case basis in sensor network and turned out to be constraint satisfying and utility based. Therefore they are definitely suited as a basis for future resource management systems and software.

It is very likely that future network management tools will keep focusing on developing and improving autonomous and distributed resource management for dynamic wireless sensor networks.[10] Therefore they should have a framework that can enable "a large set of applications with autonomous adaptation and minimum communication overhead", which is already implemented in another management tool called Collective Intelligence and aims for a higher system wide utility.[10]

With the combination of Peloton and Tiny Network Manager methods and structures and Collective Intelligence's theory, an even more efficient management tool can be created in the near future.

5. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci: *A Survey on Sensor Networks*, pages 102-105, IEEE Communications Magazine, 2002

[2] C. Chong, S. P. Kumar: *Sensor Networks: Evolution, Opportunities, and Challenges*, pages 1247-1252, Proceedings of the IEEE, Vol. 91, Nr. 8, 2003

[3] S. Walton, E. Eide: *Resource Management Aspects for Sensor Network Software*, PLOS '07 Proceedings of the 4th workshop on Programming languages and operating systems, Article No. 5, October 2007, National Science Foundation

[4] J. Waterman, G. W. Challen, M. Welsh: *Peloton: Coordinated Resource Management for Sensor Networks*, 12th Workshop on Hot Topics in Operating Systems (HotOS-XII) (2009), pages 1-5, May 2009, Harvard University

[5] M. S. Siddiqui, C. S. Hong: *Resource & Configuration Management for WSN in the Future Internet*, Applications and the Internet (SAINT), pages 193-196, July 2010, Dept. of Comput. Eng., Kyung Hee Univ., Yongin, South Korea

[6] M. J. Handy, M. Haase, D. Timmermann: *Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection*, Mobile and Wireless Communications Network, pages 368-372, September 2002, Institute of Applied Microelectronics & Computer Science, Rostock University, Germany

[7] D. Christmann: *Duty Cycling in drahtlosen Multi-Hop-Netzwerken*, Kommunikationssysteme WS 08/09, pages 1-3, 2009, Seminar für Kommunikationssysteme TU Kaiserslautern

[8] G. Werner-Allen, S. Dawson-Haggerty, M. Welsh: *Lance: Optimizing High-Resolution Signal Collection in Wireless Sensor Networks*, SenSys '08 Proceedings of the 6th ACM conference on Embedded network sensor systems, pages 169-182, November 2008, Raleigh, NC, USA

[9] H. E. Baarsma, M. G. C. Bosman, J. L. Hurink: *Resource Management in Heterogeneous Wireless Sensor Networks*, Capturing Ambient Intelligence for Mobile Communications though Wireless Sensor Networks, pages 1-9, July 2007, e-SENSE

[10] K. Shah, M. Kumar: *Resource Management in Wireless Sensor Networks using Collective Intelligence*, pages 423-428, Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP 2008, Dezember 2008

[11] WebFinance, Inc., Business Dictionary: *Resource Management*, `http://www.businessdictionary.com/definition/resource-management.html`; accessed on 10th January 2013

Simple PKI

Sebastian Wiesner
Supervisor: Ralph Holz
Seminar Innovative Internet Technologies and Mobile Communications
Chair for Network Architectures and Services
Fakultät für Informatik, Technische Universität München
Email: basti.wiesner@mytum.de

ABSTRACT

In this paper we discuss the SPKI standard as an alternative to the current X.509 and OpenPGP standards. The paper starts with a short history of PKI, and assesses the current state and the various flaws in the X.509 and OpenPGP standards. Then the main part of this paper explains the concepts of SPKI, and discusses how SPKI supports various notions of trust. Finally the paper concludes with an attempt to determine why SPKI lacks widespread implementation and deployment despite its advantages over conventional PKI.

Keywords

OpenPGP, PKI, SPKI, Trust, X.509

1. A SHORT HISTORY OF PKI

In their famous 1976 paper "New Directions in Cryptography" [10] Whitfield Diffie and Martin Hellman not only invented public key cryptography, they also envisioned the first PKI: "The enciphering key E can be made public by placing it in a public directory along with the user's name and address" [10, page 648].

Two years later the term "certificate" was introduced to describe cryptographically signed bindings from public keys to names that were to be used independently of the *public file* suggested by Diffie and Hellman [19].

By 1988 these ideas had developed into the X.509 standard, part of the X.500 family of standards. Following the idea of Diffie and Hellman these standards envision a "global, distributed directory of named entities" [14, page 6]. X.509 certificates consequently bind public keys to *global names*. These certificates are issued by *certificate authorities*.

In 1991 Phil Zimmerman published PGP (standardized as OpenPGP [5]), an alternative certificate infrastructure without certificate authorities. Instead OpenPGP allows entities and individuals to issue their own certificates. By mutual signing of certificates among different entitities, OpenPGP forms a "web of trust" [2, 8] in which people assert the validity of each others certificates. The idea behind this system is that given enough signatures all signatures of a certificate can jointly assert the validity of the certificates, even though some individual signatures might potentially be fraudulent.

In 1998, SPKI [13, 14] joined the family of public-key cryptography standards, following ideas of Ron Rivest [27] and Carl M. Ellison [12]. On the surface it can be considered as a combination of the hierarchical X.509 infrastructure and the flat PGP web. However, in fact SPKI abandons some basic assumptions of X.509 and PGP: The need of global names for identification, and the need of identification itself. SPKI uses *local names* that live within name spaces, and it introduces *authorization certificates* to allow for authorization without the need of a confirmed identity.

Today X.509 is used standards such as S/MIME [24] and SSL/TLS [9], making it the dominant certification infrastructure for secure HTTP communication and confidential mail communication. However, of late several serious incidents concerning certification authorities have raised concerns about the security of the current CA infrastructure. For instance, in 2011 over 200 fraudulent certificates were issued and supposedly used in MITM attacks after a breach into the Dutch DigiNotar CA [3, 15, 21, 22] resulting in the removal of DigiNotar from the root stores of major browsers and the subsequent bankruptcy of DigiNotar.

OpenPGP is widely used to ensure authentication and integrity of the infrastructure of free software projects[1], mostly because certificates can be issued and used without fees or additional costs, since no certification authorities are involved in the issuing of certificates and all necessary tools are provided free of charge[2]. However, OpenPGP has not been widely adopted among individuals, and thus there is no large web of trust as originally indented.

SPKI, however, has not been adopted at all. The purpose of this paper is to introduce SPKI, to assess in how far it is an improvement over X.509 and PGP, and to discuss why it was not adopted by software vendors and the internet community. The paper starts with a short introduction into the current PKI situations, its problems and its flaws. The following main section introduces SPKI, explaining the core concepts of name spaces and certificates. The next section determines how a notion of trust can be represented in SPKI, and compares this to the trust notion in OpenPGP and X.509. The paper concludes with a discussion of SPKI's success (or the lack thereof).

[1]For instance, the Debian project uses PGP to sign the packages in its software repository in order to guarantee their authenticity and integrity.
[2]For instance, the Free Software Foundation provides a fully featured PGP suite with its GNU Privacy Guard (GnuPG or PGP) project under a free software license with no charge.

doi: 10.2312/NET-2013-02-1_11

2. PKI—THE CURRENT STATE

Nowadays X.509 and OpenPGP are the dominating standards for public-key cryptography. On the surface, both standards appear to be vastly different, in fact even antithetical, however it turns out that these standards share some very basic assumptions on what a PKI is supposed to provide and how it should work.

2.1 The problem of global names

The most basic assumption of OpenPGP and X.509 is that names are *global*. This assumption is inherited from X.500 vision of a single global directory tree of named entities. In this directory tree, each entity is represented by a specific node, and this node is address by a globally unique *distinguished name*. Originally X.509 certificates were merely access tokens, granting a keyholder the permission to modify the corresponding node in the directory tree. Certificate authorities were not independent entities, but associated with nodes in the directory tree, issuing access certificates for subordinate nodes [14].

Only after it became clear that X.500 would never be deployed, X.509 certificates were extracted from the X.500 family and and—following the business needs—used as *identity certificates*. With the lack of a single directory tree, there was also no single certificate hierarchy any longer. Instead many independent and competing CAs continued to issue certificates for arbitrary global names. By and large the X.509 as used today thus ignores and even contradicts most of its original design concepts. Peter Gutmann gives and intriguing overview of the current X.509 flaws in [16].

OpenPGP [5] is not different in this aspect. Though it does not rely on CAs to issue certificates, the names—called *user ID*—in OpenPGP certificates are still global in the whole OpenPGP network.

However, global names contradict the human use of names. While there are global names—mostly world-wide brands and trademarks like Coca Cola, Apple or Amazon—the *exact meaning* of names that we use depends on our local domain.

For instance, to some people the name *Coca Cola* might refer to the Coca Cola company as a whole, to others it might only refer to the soft drink that goes by this name, and to others again it might stand as a name for all soft drinks of this kind (e.g. Pepsi Cola). Sometimes the meaning of name even depends on the context of its use.

The problem of the *meaning of a name* is increased by the fact that many names that we use are not actually unique. According to the U.S. Social Security Administration the name “Emily” was given to not less than 223,488 female children born in the U.S. between 2000 to 2009 [4]. Thus the name “Emily Smith” is not unique within U.S., probably not even within a larger city. In his popular book “Beyond fear” the renowned cryptographer and security expert Bruce Schneier tells the anecdote that another Bruce Schneier from Illinois is tired of getting Bruce Schneier's e-mail by mistake [28, page 184]. We obviously even fail to correctly identify a person who is present in the public by his books, talks and interviews.

To issue an identity certificate for Emily Smith, the ambiguous local name “Emily Smith” has to be turned into a unique global name artificially. This effectively turns the problem of distributing a public key into the problem of distributing the public name one goes by [16].

The impact of the problem can be observed in OpenPGP. Here keys are identified by user IDs which is typically the e-mail address. However, an e-mail address does not provide information about its owner, and hence cannot identified the person behind the address. Moreover due to its distributed nature OpenPGP cannot enforce unique user IDs: Two different keys may be bound to the same mail address. Thus a mail address cannot sufficiently identify a key. To address this problem, OpenPGP users typically distribute the IDs of their keys. A Key ID is a short and unique representation of the *public key itself*. Thus the problem of securely distributing one's public key is not at all solved by OpenPGP. It is merely turned into the problem of securely distributing the key ID.

2.2 Authentication and Authorization

With the trust in global names comes the believe that authenticating a key-holder by its certificate is sufficient to authorize her. The consequence is that authorization *without* authentication is impossible in OpenPGP and X.509. Neither of these standards provides means for anonymous authorization.

Thus any security system built on these standards is forced to identify and authenticate its clients in order to authorize actions, even if the identity of the client is not actually relevant for authorization. However, as explained in the previous section secure identification and authentication are no solved problems which obviously has an adverse impact on the design of any security system that could in theory work securely without any authentication at all.

3. SPKI—THE FUTURE?

Following these insights SPKI (Simple Public Key Infrastructure [13]) considers global names a fundamental conceptual flaw both in PGP and X.509 [14]. Instead, SPKI uses the *keys* themselves as global identifiers [14].These are naturally globally unique and hence serve well as a reliable global identifier.

Consequently, *keys* are the principal items in SPKI, contrary to X.509 and OpenPGP which use *names* as principals. This neatly avoids all the problems with names discussed above. Most notably it allows for anonymous authorization: Not names, but keys are the subject of authorization.

3.1 Local names

However, keys are hardly tractable by humans. Thus SPKI allows for *local* naming of keys within a *name space*. Each public key has an associated name space. In this name space, keys or names can be assigned with identifiers by *basic names*:

Definition 3.1. A *basic name* consists of a public key K and an identifier (being a word over a chosen alphabet) [7, 14].

doi: 10.2312/NET-2013-02-1_11

Example 3.1. A *basic name* might be K Fred. K is a key defining the *name space*, Fred an arbitrary identifier that may refer to another key or another basic or compound name.

A basic name is unique within a name space. However, a basic name may refer to multiple other keys or names to express *group memberships*.

A basic name is only meaningful within its name space. There is no immediate relation between names from different name spaces.

Example 3.2. K_1 Fred and K_2 Fred are *different* and *unrelated* local names that may map to different keys, even though both names happen to use the same identifier.

There are no rules restricting the identifiers of local names. Each name space may choose its own, arbitrary set of identifiers. A name space concerned with DNS (e.g. a list of known SSH hosts) may use DNS names as identifiers, another name space used for e-mail encryption may use e-mail addresses, and again another name space created from a local address book may use person names as identifiers.

Example 3.3. The following are valid, but again unrelated local names:

- K_1 Fred
- K_2 fred@example.com
- K_3 example.com

There is no top level name space, and not even any kind of hierarchy between name spaces. A priori each name space is unrelated to other name spaces.

3.2 Linkage of local namespaces

However, name spaces can be linked by combining local names to form *compound names* [7, 14]:

Definition 3.2. A *compound name* is a local name consisting of two or more identifiers.

Example 3.4. K Fred Sister is a compound name. K is a key defining the name space again. Within this name space, Fred identifies another arbitrary but fixed K_F. In the name space of K_F the identifier Sister maps to yet another arbitrary but fixed key K_S eventually. Hence the name spaces K and K_F are linked with this compound name so that Sister from K_F can be referred to from K.

By chaining basic names into compound names one can refer to names from other names spaces. However, compound names too are not meaningful outside of their name space.

Example 3.5. K_1 Fred Sister and K_2 Fred Sister are different and unrelated compound names that may map to different keys.

When no distinction between these two kinds of names is required one may speak of a *local name* or just a *name* to refer to either a *basic name* or a *compound name*.

3.3 SPKI certificates

To define a name within the name space of a key SPKI provides *name certificates*. Furthermore SPKI provides *authorization certificates* to delegate authorization to a key or a name. In the SPKI language the term "certificate" is often abbreviated as "cert" [7]. This abbreviation will be used in the following.

3.3.1 Name space definitions with name certificates

Name certs define a basic name within the name space of the signing key. A name cert for the local name KA is a signed[3] tuple of four elements $\mathrm{S}_K(K, A, S, V)$ [7, 14].

Issuer K The key K which issued the certificate.

Identifier A The identifier of the *basic name* that is defined by this cert. Note that name certs *only* define basic names. Compound names are created by the composition of several names, but never defined directly.

Subject S The "meaning" of the basic name $K\,A$, typically the key K' to which the new basic name shall map.

Validity V The validity specification. Typically this is a pair (t_1, t_2) meaning that the cert is valid only within the interval $[t_1, t_2]$. However, SPKI also allows for online validity checks. Section 5 of [14] discusses the various validity conditions that may be imposed upon SPKI certificates.

Example 3.6. The name cert $\mathrm{S}_K(K, \text{Fred}, K_F, ())$ defines the name K Fred as Fred's key K_F (see example 3.1). The cert has unlimited validity.

As said basic names are not required to be unique. Hence one may issue multiple certs for the same basic name:

Example 3.7. The name certs $(K, \text{Friends}, K\,\text{Fred}, ())$ and $(K, \text{Friends}, K\,\text{Alice}, ())$ define the name K Friends both as K Fred and as K Alice respectively. Essentially this defines a group K Friends with the members K Fred and K Alice. Separate name certs are required to define the names K Fred and K Alice.

While name certs may only define local names the subject of a name certificate may also be a compound name:

Example 3.8. The cert $(K, \text{Friends}, K\,\text{Fred Sister}, ())$ defines K Friends as the compound name K Fred Sister (see example 3.4). To fully define this compound name both the owner of K and K Fred need to issue further name certs, for instance:

- $(K, \text{Fred}, K_F, ())$ to define K Fred as key K_F, that is as the name space of K_F.

[3] More precisely it is signed with the private key corresponding to the public key K. For the sake of brevity I say "signed by a key K" instead of "signed by the private key corresponding to the public key K".

doi: 10.2312/NET-2013-02-1_11

- $(K_F, \text{Sister}, K_F\ Alice, ())$ to define K_F Sister as the name K_F Alice. Transitively K Fred Sister is now also defined as K_F Alice.
- $(K_F, \text{Alice}, K_A, ())$ to eventually define K_F Alice as Alice' key K_A.

These bindings effectively make K_F Alice a member of the group K Friends. Note however that the owner of K has no authority over names outside of her own name space. Hence the meaning of the name K Fred Sister depends on certificates issued by the owner of K *and* K Fred.

3.3.2 Authorization certificates

Names as defined by name certs provide a convenient level of abstraction from keys. However security decisions are typically not made based on *who* someone is (*identity*) but rather on *what* she is allowed to do (*authorization*).

SPKI strictly separates identity established by name certs from authorization which is conveyed by *authorization certificates* or "auth certs". An auth cert is a signed tuple of five elements $\text{S}_K(K, S, d, T, V)$ [7, 14].

Issuer K Like with name certs, K is the key that issued this cert. The owner of this K is granting the authorization conveyed by this cert.

Subject S A local name or a key that receives the authorization[4].

Delegation bit d If set the subject may delegate this authorization or a subset thereof. Delegation is performed by issuing a *new* auth cert to another subject, signed with a key that corresponds to the subject S, either directly if S is a key or—if S is a name—indirectly with a key that is defined for S by a name cert. Section 3.3.3 explains the interpretation of this flag in more detail.

Authorization tag T The authorization that is granted by this cert. The interpretation and meaning of the authorization depends on the application. The verifier has to implement the appropriate logic to interpret the authorization. SPKI however provides a standard semantic for authorization tags.

Validity specification V The validity of the cert, just like for name certs (see section 3.3.1).

Example 3.9. The auth cert

$$(K, K_F, 0, read \text{ ftp://example.com/}, ())$$

grants K_f the undelegatable and unlimited authorization to read from the URL `ftp://example.com/`. The owner of K_F can include this authorization certificate in her FTP request to the server and then sign the request with K_F. This *proves* the FTP server that the request is legitimate and authorized.

[4]SPKI also has *threshold subjects* to reflect the requirement of having K out of N key holders agree on an authorization to be granted. The discussion of these subjects is beyond the scope of this paper. See section 6.3.3 of [14] and section 10 of [7] for information about threshold subjects.

3.3.3 Chains of certificates

Auth certs are not required to grant authorization directly to a specific key. They may also grant authorization to *names*:

Example 3.10. The auth cert

$$(K, K\ Fred, 0, read \text{ ftp://example.com/}, ())$$

grants the authorization from example 3.9, but not to a key K_F, but to the name $K\ Fred$.

Granting authorization to names has some advantages over granting to keys directly:

- It abstracts from the key, making the authorization resilient in face of key changes. If $Fred$ needs to change her key the auth cert does not the to be re-issued.
- It allows for authorization to be delegated to *groups* as in example 3.7.
- It provides a user-friendly way to refer to the entity that is the subject of a cert, i.e. speaking of $Fred$ is just easier than speaking of a public key K_F.

This imposes a difficulty to $K\ Fred$: Since the auth cert does not contain a key to sign the FTP request, he needs a *another* name cert that provides a key K_F for the name $K\ Fred$. She then needs to include the auth cert *and* the name cert into her FTP request to prove her authorization. In short he needs to provide the server a *certificate chain* from the signing key to an auth cert accepted by the server.

The discovery of such chains is an interesting computational problem discussed in [7]. This paper introduces the concept of *rewrite rules* of the form $L \to R$ which rewrite names in certificates.

Definition 3.3. Name and auth certs can be represented as rewrite rules:

- A name cert $C = (K, A, S, V)$ is represented as rule $K\ A \to S$.
- An auth cert $C = (K, S, d, T, V)$ is represented as rule $K\boxed{1} \to S\boxed{z}$ with $z = 1$ if d is set and $z = 0$ otherwise. The boxed suffixes (*tickets*) control the delegation of authorization during re-writing, and ensure that undelegatable auth certs are not rewritten any further.

Based on rewrite rules the *composition of certs* is defined:

Definition 3.4. Given two certs C_1 and C_2 with the corresponding rewrite rules $L_1 \to R_1$ and $L_2 \to R_2$ respectively and $R_1 = L_2 X$ (L_2 is a prefix of R_1) then $C_3 = C_1 \circ C_2 = L_1 \to R_2 X$. X may be empty.

Composing certs allows to combine multiple certs into a single, "virtual" certs that captures the "meaning" of all these certs:

Example 3.11. Two name certs

$$C_1 = (K, \text{Friends}, K \text{ Fred Sister}, V)$$
$$C_2 = (K, \text{Fred}, K_F, V)$$

doi: 10.2312/NET-2013-02-1_11

together define K Friends as K_F Sister. With the rewrite rules corresponding to C_1 and C_2

$$C_1 = K \text{ Friends} \rightarrow K \text{ Fred Sister}$$
$$C_2 = K \text{ Fred} \rightarrow K_F$$

we can now *compute* the virtual certificate that captures this definition:

$$C_3 = C_1 \circ C_2 = K \text{ Friends} \rightarrow K_F \text{ Sister} = \\ = (K, \text{Friends}, K_F \text{ Sister}, V)$$

Now the interpretation of the delegation bit becomes clear. If the delegation bit is set, the right hand side of the rule has a "live" ticket $\boxed{1}$. This right hand side can be rewritten further with other auth certs, reflecting the delegation of authority to another subject.

Example 3.12. The auth certs

$$C_1 = (K, K_F, 1, A, V) = K \boxed{1} \rightarrow K_F, \boxed{1}$$
$$C_2 = (K_F, K_A, 0, A, V) = K_F \boxed{1} \rightarrow K_A \boxed{0}$$

can be composed to

$$C_3 = C_1 \circ C_2 = K \boxed{1} \rightarrow K_A \boxed{0} = \\ = (K, K_A, 0, A, V).$$

The certificate C_3 proves the authorization of the owner of K_A to the owner of K. Thus using the chain C_1, C_2 the authorization A is effectively delegated to the owner of K_A.

If the delegation bit is unset however, the right hand side of a rule has a "dead" ticket $\boxed{0}$. Since the left hand side of an auth cert rule always has a live ticket, the left hand side of an auth cert rule cannot be a prefix of a term with a dead ticket. Thus a right hand side with a dead ticket cannot be rewritten by another auth cert, but only by name certs. Hence the authorization cannot be delegated to other subjects.

Example 3.13. The authorization conveyed by the auth cert

$$C_1 = (K, K \text{ Friends}, 0, A, V) = K \boxed{1} \rightarrow K \text{ Friends} \boxed{0}$$

cannot be delegated any further, because there is no auth cert can be a prefix of the right hand side of this rule. Thus there is no valid chain that can include another auth cert after this cert.

However this cert can still be rewritten with name certs. Consider the name cert

$$C_2 = (K, \text{Friends}, K_F, V) = K \text{ Friends} \rightarrow K_F.$$

This cert can be composed with C_1 to yield

$$C_3 = C_1 \circ C_2 = K \boxed{1} \rightarrow K_F \boxed{0} = \\ = (K, K_F, 0, A, V).$$

Hence using the chain C_1, C_2 the owner of K_F can prove her authorization to the owner of K. Effectively K_F is granted the authorization A for being a member of the $Friends$ group.

The composition of certs is transitive, hence there is a transitive closure C^+ for a set of certificates C. The *name reduction closure* $C^\#$ is a subset of C^+ created by only including compositions that *strictly reduce* the right hand side of the composition.

In order to find a chain that proves authentication A for the key K one now takes the set C of all valid auth certs for A and all valid name certs and computes the name reduction closure $C^\#$. From this closure which contains all intermediate compositions from the auth certs to the key K (if there are any) one can create a graph with a vertex for each key and an edge for each auth cert in $C^\#$.

A certificate chain that proves authorization A for the key K can now be created by a simple breadth-first search over this graph. If no chain is found, no such chain exists in the set C meaning that the owner of K actually lacks the authorization A [7].

3.4 S-expressions

SPKI uses S-expressions—as known from the LISP family of languages[5]—to encode names, authentications and even certs[6] These expressions are a human-readable and well-understood syntax to encode data structures which makes SPKI certificates pleasingly simple to read and comprehend if compared to the complexity of the ASN.1-encoded X.509 certificates.

3.4.1 Names as S-expressions

S-expressions starting with the tag `name` encode local names:

Example 3.14. Within a cert issued by the key K the S-expression `(name Fred)` encodes the basic name K Fred (see example 3.1). Outside of a cert fully qualified names [14] such as `(name (hash sha1 H(K)) Fred)` must be used to explicitly specify the name space. $H(K)$ must be substituted with the SHA1 hash of the key K[7].

Example 3.15. Within a cert issued by the key K the S-expression `(name Fred Sister)` encodes the compound name K Fred Sister (see example 3.4). Outside of a cert a fully qualified name must be used, just like in the previous example.

3.4.2 Authentication as S-expressions

The encoding of names as S-expressions is merely convenient but in the encoding of authentication S-expressions become truly powerful. Authorization is encoded as a (nested) list of strings [14] starting with `tag`:

Example 3.16. The authorization tag

```
(tag (ftp ftp.example.com /foo/bar))
```

might allow FTP access to the directory `/foo/bar` on the host `ftp.example.com`.

[5]See Rivest's S-expressions page [26] and guide [25] for more information about S-expressions.
[6]The encoding of certs as S-expressions is beyond the scope of this paper. [11] has examples of certs as S-expressions. This S-expression encoding of certs is not mandatory. Section 6.5 of RFC 2693 [14] defines translation rules from X.509 and PGP certificates to SPKI certs.
[7]Of course, other hash algorithms may be used just as well.

doi: 10.2312/NET-2013-02-1_11

SPKI leaves the definition of the meaning of authorization to the application that verifies the authorization, allowing for arbitrarily complex authorizations to be conveyed. However SPKI specifies a simple and entirely optional semantic to interpret and intersect authorizations to provide developers with a standardized, yet flexible interpretation of authorization [14]:

In this semantics each item in an authorization further restricts the granted authorization:

Example 3.17. The two authorization tags

```
(tag (ftp ftp.example.com))
(tag (ftp ftp.example.com /foo/bar))
```

intersect to

```
(tag (ftp ftp.example.com /foo/bar))
```

Furthermore some wild card constructs are supported, most notably `(*)`, which matches everything, and `(* set)`, `(* prefix)` and `(* range)` which match against a set of values, a string prefix and a range of values respectively [14].

Example 3.18. The two authorization tagss

```
(tag (ftp ftp.example.com
          (* set read write)))
(tag (ftp ftp.example.com
          (* set read delete)))
```

intersect to

```
(tag (ftp ftp.example.com (* set read)))
```

Example 3.19. The two authorization tagss

```
(tag (ftp ftp.example.com
          (* prefix /foo/)))
(tag (ftp ftp.example.com
          (* prefix /foo/bar/)))
```

intersect to

```
(tag (ftp ftp.example.com
          (* prefix /foo/bar/)))
```

By relying on this standard semantics developers can easily implement authorization checks in applications. One just needs to construct the complete S-expression that authorizes access to a desired resource and intersect this expression with the expression contained in the authorization certificate that desires access. Access will be granted if the expressions intersect.

4. TRUST

SPKI provides a distributed certificate infrastructure. In such an infrastructure participating entities are likely to have only an indirect relation between each other. For instance, in PGP one may receive a key not directly from the key owner, but instead from a 3rd party like a public key server. This key may be signed by other keys whose owners by not be known at all. In SPKI, one may have to verify an authorization delegated to an unknown entity.

Naturally the question arises in how far such signatures or such delegated authorizations can be *trusted* in various aspects. In PGP one might want to know if a signer really verified the key owner's identity before signing the key. In SPKI one might want to know whether the delegator of an authorization really ensured that the subject of the delegation will use the authorization appropriately.

4.1 The problem of transitive trust

These questions can be generalized to the question in how far statements of entities are trusted. If the entity making a statement is directly known one can directly assess trust into this entity. One can gather enough information and execute sufficient checks to ensure that this entity will behave as expected.

If the entity is not directly known, this is not longer possible. Instead one has to rely on *somebody else's* statement about the trust into this entity. Of course this statement is affected by the trust one has into the entity making this statement. Trust becomes a *transitive* relation which arises the requirement to adequately communicate trust and to make automated decisions about trust.

4.2 Trust in X.509

As explained in section 2.1 there are different competing CAs that issue certificates. Hence every X.509 application has a *root store* containing certificates of implicitly trusted CAs.

However, there is no standardized formal process for the inclusion of CAs in root stores. Every application and every organization has its own processes and guidelines regarding the inclusion of CAs in root stores or their removal thereof. These processes and guidelines greatly vary in quality, and consequently many applications include a lot of CAs in their root stores.

All of these root CAs have equal signing authority and may issue certificates for arbitrary names. They may even create subordinate CAs by issuing *intermediate certificates* which are normally permitted to issue arbitrary certificates themselves.

Thus one effectively trusts a lot of CAs with equal signing authority, many of which are not even known to the user. The security of the whole infrastructure is thus lowered to the weakest CAs reachable via the root store. The trust model of X.509 is simply *unlimited* and *ultimate transitive* trust.

4.3 Trust in GPG

OpenPGP provides a far less transitive, but still simple trust model. A OpenPGP user may assign an *Owner trust* level to a key in her key ring. This trust level should reflect in how far the owner of that key is trusted to *introduce* keys, i.e. in how far the owner's signature on another key is trusted. PGP knows three trust levels *not trusted*, *marginally trusted* and *completely trusted* [2, 8, 18].

doi: 10.2312/NET-2013-02-1_11

To OpenPGP these levels assess the quality of a signature of this key. PGP calculates the authenticity of a key as a weighed sum of the number of *marginally trusted* and *completely trusted* signatures on that key, based on the user's preferences on how many marginally or completely trusted signatures make a key trusted.

While these trust levels define the trust given to introduced keys, the trust levels themselves are private and not communicated to the outside. They are *not* transitive.

4.4 Trust in SPKI—By example

Contrary to X.509 and OpenPGP SPKI does not specify any kind of trust management model. Trust management in SPKI is left to the application which may implement arbitrary trust semantics and algebras.

One such algebra is presented in [18]. It assesses trust based on a framework called *Subjective Logic* which is essentially a calculus for *opinions* [18]:

Definition 4.1. An opinion is a triple $\omega = \{b, d, u\}$ where $b + d + u = 1$ and $\{b, d, u\} \in [0, 1]^3$. The components are the *belief* b, the *disbelief* d and the *uncertainty* u. $\omega_p^A = \{b_p^A, d_p^A, u_p^A\}$ is the opinion of an *agent* A about a statement p.

The *belief* and *disbelief* components describe the probability of whether a statement is true or not. The *uncertainty* component compensates for the absence of knowledge about a statement.

Such opinions provide the base for a logic calculus of *conjunction*, *recommendation* and *consensus* [18][8].:

Definition 4.2. The *conjunction* $\omega_{p\wedge q}^A = \omega_p^A \wedge \omega_q^A$ of the opinions ω_p^A and ω_q^A about two discrete binary statements p and q represents A's opinion about p *and* q being true.

Definition 4.3. The *recommendation* $\omega_p^{AB} = \omega_B^A \otimes \omega_p^B$ is A's opinion about the statement p *as a result of a recommendation from* B. ω_B^A is A's opinion about B's recommendations. ω_p^B is B's opinion about the statement p *as recommended to* A[9].

In order two assess trust in an SPKI certificate two special opinions are of importance [18]:

Definition 4.4. The opinion $\omega_{KA(K_I)}^R$ is the opinion of the recipient R of a certificate about the *authenticity of the key* K_I of the certificate's issuer I, that is, whether the key really belongs to the issuer.

Definition 4.5. The opinion $\omega_{RT(I)}^R$ is the opinion about the recipient R of a certificate about the *recommendation trustworthiness* of the certificate's issuer I, that is, how much R trusts I to issue certificates correctly[10].

[8]The exact mathematical definition of these operators are not reproduced here, as these are not relevant for the discussion of this algebra within the scope of this paper. Refer to section 3 of [18] for details.

[9]This is not necessarily identical to B'S *real* opinion about p.

[10]This is essentially equivalent to the Owner trust of PGP, however at a much finer level.

Together these two opinions form R's opinion about a certificate issued by I, expressed as the *conjunctive recommendation term* $\omega_I^R = \omega_{KA(K_I)}^R \wedge \omega_{RT(I)}^R$ [18]. The trust in a certificate subject S is consequently expressed as a recommendation $\omega_S^R = \omega_I^R \otimes \omega_{KA(K_S)}^I$ where K_S is the public key of S. By chaining such expressions one can now compute the relative trust of a certificate chain in which each certificate's subject has an opinion about the subsequent certificate.

This can naturally be applied to SPKI. As discussed in section 3.3.3 the verifier of an authorization needs a complete chain of certificates as proof of authorization. To express the trust in such a chain, each certificate C_i in this chain has to include an opinion about the authenticity of its subject.

The only obstacle are names used as certificate subjects. Obviously one cannot assess the *key authenticity* of names and hence not calculate their trust opinion. However, in a valid certificate chain a name subject must eventually be bound to a key by a name cert. Hence the opinion of trust into a name is equivalent to the opinion about trust in all name certs needed to obtain a key for the name.

The verifier of an authorization can calculate the trust of the chain for use as parameter in its decision about a request to a protected resource, and for instance only accept request that provide chains whose trust belief exceed a certain threshold. Combined with SPKI's authorization delegation this provides a flexible way of delegating and verifying authorization that can model a wide range of organizational structures.

5. FAILURE AND SUCCESS OF SPKI

This paper has introduced SPKI and discussed the definition and semantics of its certificates, and revealed the flexibility, elegance and expressiveness of naming and authorization. It has furthermore analyzed how trust can be measured and assessed in certificate chain.

It has however not given an practical use of SPKI, simply because there is none. There has been research about using SPKI with various technologies and protocols, for instance DNS [17], HTTP [6] or sensor network [23], but to this days SPKI has not seen wide-spread adoption in real applications. Especially it has not replaced X.509 in application though that was an original intent of SPKI.

One may speculate about the reasons for this lack of deployment. Probably SPKI just came too late. By the time it was standardized and sufficiently researched, X.509 was already widely employed and had become the cryptographic backbone of the internet. With a cryptographic infrastructure at hand, there was little motivation to implement yet another one. Moreover SPKI did not offer a business model for companies, thus there was little interest to invest into the deployment of this standard.

A decade after the standardization of SPKI it seems unlikely that SPKI will ever happen to replace X.509 in important application. However, it may serve well in special domains and niche applications that have need of a simple and easy to implement public key standard. This paper may help developers of such applications to look beyond X.509 and

doi: 10.2312/NET-2013-02-1_11

possibly discard its complexity in favour of a really *simple public key infrastructure*.

References

[1] M. Abadi. "On SDSI's Linked Local Name Spaces". In: *CSFW*. Ed. by IEEE Computer Society. IEEE. 1997, pp. 98–108.

[2] A. Abdul-Rahman. "The PGP trust model". In: *EDI-Forum: the Journal of Electronic Commerce* 10.3 (1997), pp. 27–31.

[3] H. Adkins. *An update on attempted man-in-the-middle attacks*. Google Inc. Aug. 29, 2011. Url: http://googleonlinesecurity.blogspot.de/2011/08/update-on-attempted-man-in-middle.html.

[4] U.S. Social Security Administration, ed. *Top names of the 2000s*. May 14, 2012. Url: http://www.socialsecurity.gov/OACT/babynames/decades/names2000s.html.

[5] J. Callas et al. *OpenPGP Message Format*. RFC 4880 (Standards Track). IETF, Nov. 2007. Url: http://www.ietf.org/rfc/rfc4880.txt.

[6] D. Clarke. "SPKI/SDSI HTTP Server / Certificate Chain Discovery in SPKI/SDSI". Master's Thesis. Massachusetts Institute of Technology, 2001.

[7] D. Clarke et al. "Certificate Chain Discovery in SPKI/SDSI". In: *Journal of Computer Security* 9.4 (2001), pp. 285–322.

[8] M. Copeland, J. Grahn, and D. Wheeler. *The GNU Privacy Handbook*. The Free Software Foundation. 1999. Url: http://www.gnupg.org/gph/en/manual.html.

[9] T. Dierks and E. Rescorla. *The Transport Layer SEcurity (TLS) Protocol Version 1.2*. RFC 4492 (Standards Track). IETF, Aug. 2008. Url: http://www.ietf.org/rfc/rfc5246.txt.

[10] W. Diffie and M. Hellman. "New Directions in Cryptography". In: *IEEE Transactions on Information Theory* 6.22 (Nov. 1976), pp. 644–654.

[11] J. Elien. "Certificate Discovery Using SPKI/SDSI 2.0 Certificates". Master's Thesis. Massachusetts Institute of Technology, 1998.

[12] C. Ellison. "Establishing Identity Without Certification Authorities". In: *Proceedings of the 6th USENIX Security Symposium*. Ed. by USENIX. 1996, pp. 67–76.

[13] C. Ellison. *SPKI Requirements*. RFC 2692 (Experimental). IETF, Sept. 1999. Url: http://www.ietf.org/rfc/rfc2692.txt.

[14] C. Ellison et al. *SPKI Certificate Theory*. RFC 2693 (Experimental). IETF, Sept. 1999. Url: http://www.ietf.org/rfc/rfc2693.txt.

[15] Mozilla Foundation, ed. *Protection against fraudulent DigiNotar certificates*. MFSA 2011-34. Aug. 30, 2011. Url: https://www.mozilla.org/security/announce/2011/mfsa2011-34.html.

[16] P. Gutmann. *Everything you Never Wanted to Know about PKI but were Forced to Find Out*. 2002.

[17] T. Hasu and Y. Kortesniemi. "Implementing an SPKI Certificate Repository within the DNS". In: *Poster Paper Collection of the Theory and Practice in Public Key Cryptography (PKC 2000* (2000), pp. 18–20.

[18] A. Jøsang. "An Algebra for Assessing Trust in Certification Chains". In: *NDSS*. Ed. by The Internet Society. 1999.

[19] L. Kohnfelder. "Towards a Practical Public-key Cryptosystem". Bachelor. Massachusetts Institute of Technology, May 1978.

[20] I. Lehti and P. Nikander. "Certifying Trust". In: *Public Key Cryptography*. Ed. by Hideki Imai and Yuliang Zheng. Springer, 1998, pp. 83–98.

[21] J. Nightingale. *DigiNotar Removal Follow Up*. Mozilla Foundation. Sept. 2, 2011. Url: http://blog.mozilla.org/security/2011/09/02/diginotar-removal-follow-up/.

[22] Johnathan Nightingale. *Fraudulent *.google.com Certificate*. Mozilla Foundation. Sept. 6, 2011. Url: http://blog.mozilla.org/security/2011/08/29/fraudulent-google-com-certificate/.

[23] C. Pearce, V. Yin-Man Ma, and P. Bertok. "A secure communication protocol for ad-hoc wireless sensor network". In: *ICISSNIP*. Ed. by IEEE Computer Society. IEEE. 2004, pp. 79–84.

[24] B. Ramsdell and S. Turner. *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification*. RFC 5751 (Standards Track). IETF, Jan. 2010. Url: http://www.ietf.org/rfc/rfc5751.txt.

[25] R. Rivest. *S-Expressions*. Massachusetts Institute of Technology. May 4, 1997. Url: http://people.csail.mit.edu/rivest/Sexp.txt.

[26] R. Rivest. *SEXP---(S-expressions)*. Massachusetts Institute of Technology. May 4, 1997. Url: http://people.csail.mit.edu/rivest/sexp.html.

[27] R. Rivest and B. Lampson. *SDSI - A Simple Distributed Security Infrastructure*. Massachusetts Institute of Technology. Sept. 15, 1996. Url: http://people.csail.mit.edu/rivest/sdsi10.html.

[28] B. Schneier. *Beyond Fear. Thinking Sensibly About Security in an Uncertain World*. Springer Science+Business Media, 2006. ISBN: 978-0-387-02620-6.

What is individual-related data?

Christian Eckert
Betreuer: Johann Schlamp
Seminar Innovative Internettechnologien und Mobilkommunikation WS12/13
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: christian.eckert@in.tum.de

KURZFASSUNG

Der Schutz der Privatsphäre gewinnt durch die steigende Vernetzung und Nutzung des Internets an Bedeutung. In dieser Arbeit werden deshalb die rechtlichen Grundlagen in Deutschland und in der EU bezüglich personenbezogener Daten aufgearbeitet. Dabei stellt sich in der Praxis heraus, dass die Auslegung dieser Definitionen oftmals schwammig ist und zu kontroversen Gerichtsentscheidungen führt. In der Vergangenheit wurde in Forschungsprojekten gezeigt, dass es neben Cookies noch weitere Möglichkeiten gibt, einen Web-Browser über eine Art Fingerabdruck wiederzuerkennen. Dadurch ist es möglich, Bewegungsprofile einzelner Nutzer zu erstellen, was eine Bedrohung der Privatsphäre darstellt. Der Autor beschränkt sich hierbei auf den Informationsgehalt, der über den HTTP-Header des Web-Browsers gewonnen werden kann. Um dieser Bedrohung entgegenzuwirken, werden in dieser Arbeit verschiedene Maßnahmen und Modelle vorgestellt, die dem Schutz der Privatsphäre dienen. Dabei zeigt sich, dass es der Internetnutzer selbst in der Hand hat, dieses wichtige Gut zu bewahren.

Schlüsselworte

personenbezogene Daten, Datenschutz, Anonymität im Internet, HTTP-Header, k-anonymitiy, l-diversity, t-closeness

1. EINLEITUNG

Nicht nur durch die steigende Nutzung des Internets, sondern auch die Entwicklung hin zum Web 2.0 stellen eine Bedrohung der Privatsphäre im Internet dar. Dabei agiert der Benutzer im Internet nicht mehr nur als Konsument von Texten, Bildern und Videos, sondern wird dazu ermutigt, selbst eigene Inhalte zu veröffentlichen. Während diese Gefahr noch selbst reguliert und beispielsweise durch die Reduzierung der freiwilligen Preisgabe von persönlichen Inhalten gemindert werde kann, stellt die steigende Nutzung des Internets weiterhin ein Problem dar. Wie in einem Forschungsprojekt der Electronic Frontier Foundation demonstriert wurde, können Web-Browser nicht nur durch Cookies, sondern auch durch eine Reihe anderer Informationen identifiziert werden [9]. Die Auswertung dieses Projekts zeigt, dass die Kombination des zugrundeliegenden Betriebssystems, systemseitige Einstellungen wie Bildschirmauflösung und Spracheinstellungen, sowie die Wahl des Browsers und die darin installierten Plugins oftmals eine einzigartige Signatur ergeben, wodurch der Benutzer verfolgbar wird [8].

Mit diesem Hintergedanken werden in dieser Arbeit weniger die technischen Aspekte zur Erstellung eines Browser-Fingerabdrucks beleuchtet, sondern vielmehr die zugrundeliegenden gesetzlichen Regelungen zur Speicherung und zum Schutz von personenbezogenen Daten erläutert. In Kapitel 3 werden, basierend auf den gesetzlichen Bestimmungen, die Daten eines HTTP-Headers sowie die Log-Möglichkeiten eines Webservers näher betrachtet und bewertet. Verschiedene Modelle zum Schutz der Privatsphäre werden in Abschnitt 4 vorgestellt. Dabei werden mögliche Schutzmaßnahmen nicht nur aus der Perspektive des Benutzers vorgestellt, sondern auch auf die Rollen des Software-Entwicklers und des Webseiten-Betreibers eingegangen. Auf verwandte Arbeiten wird in Kapitel 5 verwiesen und Kapitel 6 enthält abschließend neben einer Zusammenfassung einen Ausblick zum Thema.

2. RECHTLICHE GRUNDLAGEN

Die nationalen und internationalen Beschlüsse und Gesetze rund um das Thema Datenschutz und Privatsphäre stellen die Grundlage dieser Arbeit dar. Darauf basierend können, die im Internet anfallenden Daten, beispielsweise die Inhalte eines HTTP-Header aus einem Webserver-Log, bezüglich ihrer rechtlichen Bedeutung, klassifiziert werden. Daraus wiederum resultieren die Vorschriften und die zu treffenden Maßnahmen für die Speicherung und den Schutz dieser Daten.

Einer der ersten Ansätze zum Schutz der Privatsphäre wurde im Jahre 1948 in der Generalversammlung der Vereinten Nationen beschlossen. Dabei kann das Recht auf Privatsphäre im weitesten Sinne als ein Bestandteil der international geltenden Allgemeinen Erklärung der Menschenrechte angesehen werden. Der Artikel 12 wurde dabei wie folgt definiert:

> "Niemand darf willkürlichen Eingriffen in sein Privatleben, seine Familie, seine Wohnung und seinen Schriftverkehr oder Beeinträchtigungen seiner Ehre und seines Rufes ausgesetzt werden. Jeder hat Anspruch auf rechtlichen Schutz gegen solche Eingriffe oder Beeinträchtigungen." [23]

Einen expliziten Schutz der Privatsphäre im Internet wurde dabei zwar nicht definiert - zumal das Internet damals noch gar nicht existierte - dennoch kann dies als erster globaler Grundstein zum Schutze des Privatlebens angesehen werden.

Aufbauend auf dieser Basis wurden sowohl in der Europäischen Union, als auch auf nationaler Ebene mit dem Bundes-

doi: 10.2312/NET-2013-02-1_12

datenschutzgesetz weitere Regelungen zum Schutz der Privatsphäre und folglich auch zum Schutz von personenbezogenen Daten im Internet definiert. Innerhalb der Europäischen Union ist dies in der Richtlinie 95/46/EG geregelt, die zum Schutz natürlicher Personen bei der Verarbeitung von personenbezogenen Daten und zur freien Datenverkehr dient [10].

Derzeit wird über eine Erneuerung der Datenschutzrichtlinie 95/46/EG verhandelt. Die ersten Reformvorschläge wurden bereits im Frühjahr 2012 präsentiert. Die Reform zielt im Besonderen darauf ab, ein einheitliches und hohes Datenschutzniveau in der EU sicherzustellen. Das Hauptaugenmerk liegt dabei vor allem auf den neuen technologischen Weiterentwicklungen wie beispielsweise mobiles Internet, Suchmaschinen und soziale Netzwerke. Das Ziel dabei ist die Sicherstellung der Transparenz in der Datenverarbeitung, die Gewährleistung der Betroffenenrechte sowie die Verpflichtung der Unternehmen, ihre Produkte von vornherein mit datenschutzfreundlichen Technologien auszustatten [11].

Die Umsetzung der Datenschutzrichtlinie 95/46/EG wird in Deutschland durch das Bundesdatenschutzgesetzt (BDSG) geregelt [1].

2.1 Was sind personenbezogene Daten?

Die zentrale Frage dieser Arbeit behandelt die Definition von personenbezogenen Daten, welche hier anhand der gesetzlichen Regelung in Deutschland als auch anhand der Richtlinie der Europäischen Union beantwortet werden soll. Innerhalb der EU Richtlinie steht der Ausdruck *personenbezogene Daten* für "[...] alle Informationen über eine bestimmte oder bestimmbare natürliche Person" (vgl. Artikel 2a Richtlinie 95/46/EG, [10]). Weiterhin wird eine Person als bestimmbar bezeichnet, wenn diese "direkt oder indirekt identifiziert werden kann, insbesondere durch Zuordnung zu einer Kennummer oder zu einem oder mehreren spezifischen Elementen, die Ausdruck ihrer physischen, physiologischen, psychischen, wirtschaftlichen, kulturellen oder sozialen Identität sind" (vgl. Artikel 2a Richtlinie 95/46/EG, [10]).

An diese Richtlinie angelehnt wird in Deutschland folgende Begriffsbestimmung verwendet:

> "Personenbezogene Daten sind Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person (Betroffener)." (vgl. § 3 Absatz 1 BDSG [1])

Einfache Beispiele personenbezogener Daten sind der vollständige Name oder die Anschrift einer Person. Mittels dieser Daten kann offensichtlich direkt eine bestimmte Person ausgemacht werden. Bei Informationen wie Familienstand, Alter oder Kontonummer handelt es sich zunächst um Einzelangaben über persönliche Verhältnisse. Auch der Besitz eines bestimmten Gegenstandes, beispielsweise eines Autos, ist als Einzelangabe sachlicher Verhältnisse einzuordnen. Diese Einzelangaben werden erst dann zu personenbezogenen Daten, wenn daraus eine bestimmte oder bestimmbare natürliche Person ableitbar ist. Dies ist immer dann der Fall, wenn diese Einzelangaben in einer Art Liste zusammen mit dem Namen gespeichert werden.

Deutlich schwieriger allerdings wird die Einstufung der Daten, wenn nicht direkt von einem einzelnen Datum auf eine Person geschlossen werden kann, aber aus einer Kombination verschiedener Einzelangaben. Bei einer Liste, die lediglich das Geschlecht und das Geburtsdatum verschiedener Personen enthält, kann sicherlich noch nicht von personenbezogenen Daten gesprochen werden. Was ist aber, wenn diese Liste neben dem Geschlecht und dem Geburtsdatum zusätzlich noch eine Postleitzahl enthält? Zunächst könnte angenommen werden, dass es sich selbst dann noch um eine anonyme Liste handelt. Die Informatikerin Latanya Sweeney hat jedoch gezeigt, dass allein durch diese Informationen 53% der Bevölkerung in den USA einzigartig und sogar 83% der US-Amerikaner mit großer Wahrscheinlichkeit identifiziert werden können [20].

Weiterhin spezifiziert das BDSG *besondere Arten* von personenbezogenen Daten. Dabei handelt es sich um die Angaben über religiöse oder philosophische Überzeugungen, die rassische und ethnische Herkunft, politische Meinungen, Gewerkschaftszugehörigkeit, Gesundheit oder Sexualleben (vgl. § 3 Absatz 9 BDSG [1]).

Neben den allgemeinen Bestimmungen im BDSG, die in § 1 bis § 11 definiert sind, wird zusätzlich zwischen der Datenverarbeitung seitens öffentlicher Stellen (§§ 12-26) und nicht-öffentlicher Stellen (§§ 27-38a) unterschieden. Unter nicht-öffentlichen Stellen sind (private) Unternehmen jeglicher Rechtsform aber auch öffentlich-rechtliche Wettbewerbsunternehmen zu verstehen. Abhängig vom Anwendungsbereich werden dabei die allgemeinen Bestimmungen aus § 1 bis § 11 bezüglich der Speicherung, Verarbeitung und Nutzung personenbezogener Daten im Einzelnen genauer definiert.

2.2 Verarbeitung und Schutz von personenbezogenen Daten

Bei der Zulässigkeit der Datenerhebung, -verarbeitung und -nutzung stimmen die Regelungen der EU sowie die Bestimmungen im BDSG größtenteils überein. Um personenbezogene Daten zu schützen, wird dabei eine Erhebung, Verarbeitung und Nutzung dieser Daten zunächst generell verboten. Ausnahmen werden nur dann erlaubt, wenn eine Einwilligung der betroffenen Person vorliegt oder eine andere Rechtsvorschrift dieses Gesetztes dies erlaubt (vgl. § 4 BDSG [1] und Artikel 7 Richtlinie 95/46/EG [10]). Zusätzlich muss nach § 28 des BDSG der Zweck der Datenverarbeitung oder -nutzung konkret festgelegt werden, wenn diese dem eigenen Geschäftszweck dient [1].

Weiterhin besagt § 3a des BDSG, dass die Systeme zur Erhebung, Verarbeitung und Nutzung personenbezogener Daten mit dem Ziel der Datenvermeidung und Datensparsamkeit auszurichten sind. Zudem müssen personenbezogene Daten anonymisiert oder pseudonymisiert werden, sofern dies nach dem Verwendungszweck möglich ist und nicht einen unverhältnismäßigen hohen Aufwand erfordert [1]. Eine solche Regelung ist in der Datenschutzrichtlinie 95/46/EG nicht enthalten.

doi: 10.2312/NET-2013-02-1_12

2.3 Regelung zur Speicherung von personenbezogenen Daten

Für elektronische Informations- und Kommunikationsdienste besagt das Telemediengesetz (TMG) nach § 13 Abs. 4 Nr. 2, dass anfallende personenbezogene Daten nach Ablauf der Nutzung unmittelbar gelöscht werden müssen. Die Verwendung von Nutzungsdaten über das Ende der Nutzungsdauer hinaus ist nach § 15 Abs. 4 TMG nur dann zulässig, wenn sie für die Zwecke der Abrechnung mit dem Nutzer erforderlich sind [2]. Darüber hinaus besagt § 9 des BDSG, dass bei der Verarbeitung und Speicherung von personenbezogene Daten ein Mindestmaß an Sicherheitsvorkehrungen getroffen werden muss. In der Anlage zu § 9 werden dabei konkrete Maßnahmen definiert. Dazu soll Unbefugten der Zutritt sowie der Zugriff auf die Datenverarbeitungssysteme verwehrt werden. Weiterhin muss bei der Übertragung von personenbezogenen Daten ein Verschlüsselungsverfahren verwendet werden, welchem dem Stand der Technik entspricht [1].

3. ANALYSE VON HTTP-HEADERN UND WEBSERVER-LOGS

Der HTTP-Header ist ein Bestandteil des Hypertext Transfer Protocol (HTTP), der wichtige Informationen bei der Kommunikation zwischen Browser und Webserver übermittelt. Hierbei soll sichergestellt werden, dass Nachrichten auf beiden Seiten korrekt interpretiert werden. Darüber hinaus wird die Möglichkeit geschaffen, mit Hilfe des HTTP-Headers die Webseite an die Gegebenheiten des Clients anzupassen. Dies betrifft beispielsweise die Anpassung der Sprache an die Voreinstellungen des Benutzers, oder auch die Optimierung der Webseite für mobile Geräte. Dabei wird unterschieden zwischen den Anfrage-Headerfeldern, welche vom Browser an den Webserver gesendet werden, und den Antwort-Headerfeldern, die der Webserver bei der Antwort an den Browser verwendet.

Dass der HTTP-Header nicht nur zur korrekten Interpretation der Daten sowie zur Optimierung der Webseite an den Client genutzt werden kann, hat das Forschungsprojekt der Electronic Frontier Foundation (EFF) gezeigt [9]. Dabei ist es gelungen, aus den erhaltenen Daten des HTTP-Headers sowie durch weitere Abfragen mittels Java, JavaScript und Flash einen Fingerabdruck des Browsers herzustellen. In dieser Arbeit jedoch liegt der Fokus ausschließlich auf dem HTTP-Header. Dabei ist in diesem Fall eine weitere Einschränkung auf die Anfrage-Headerfeldern möglich, da nur diese eine potenzielle Gefahr für den Benutzer darstellen. Die Header-Felder werden mittels Request for Comments (RFCs) von der Internet Engineering Task Force (IETF) spezifiziert. Zusätzlich dazu können Hersteller eigene Erweiterungen implementieren, die dann allerdings nicht von jedem Webserver bzw. Browser verstanden werden. Die aktuell gültige Spezifikation des Hypertext Transfer Protocol ist in RFC 2616 dokumentiert, welcher später durch weitere RFCs erweitert wurde [12].

3.1 Inhalt des HTTP-Headers

Mit Hilfe der gesetzlichen Bestimmungen aus dem vorherigen Kapitel ist eine Basis geschaffen, um die Inhalte eines HTTP-Headers bezüglich ihrer rechtlichen Relevanz einzuordnen. Zunächst soll Tabelle 1 einen Überblick der in einem HTTP-Header enthaltenen Felder bieten. Dabei handelt es sich lediglich um einen Auszug der Anfrage-Felder, die im Rahmen dieser Arbeit sinnvoll erscheinen.

Tabelle 1: Auszug der HTTP-Header Anfrage-Felder

Feld	Beschreibung
Accept	Vom Browser akzeptierte Dateitypen
Accept-Charset	Vom Browser unterstützte Zeichensätze
Accept-Encoding	Gibt die unterstützten komprimierten Formate an
Accept-Language	Unterstützte Spracheinstellungen
Authorization	Enthält Authentifizierungsdaten
Expect	Besschreibt das Verhalten, das der Client vom Server erwartet
From	Beinhaltet die E-Mail Adresse des Nutzers; allerdings muss der Nutzer ausdrücklich zustimmen, wenn diese gesendet wird
Host	Domain-Name des Webservers
Referer	Enthält die URI der Webseite von der aus auf die aktuelle Webseite über ein Link verwiesen wurde
User-Agent	Beinhaltet Informationen über den Client, wie zum Beispiel Betriebssystem und Webbrowser

Ein weiteres für diese Arbeit relevantes Anfrage-Headerfeld wurde in RFC 4229 hinzugefügt. Dabei handelt es sich um das sogenannte Cookie, das von dem Webbrowser gespeichert wird und dazu dient, einen Browser beim erneuten Besuchen einer Webseite wiederzuerkennen [18].

3.2 Rechtliche Einstufung der Daten

Die in Tabelle 1 aufgelisteten Anfrage-Headerfelder sind einzeln für sich betrachtet aus gesetzlichen Gesichtspunkten eher unkritisch, da es sich weder um sensible Informationen noch um Schlüsselattribute handelt, mit Hilfe derer eine Person direkt identifiziert werden kann. Einzige Ausnahme stellt das *From* Header-Feld dar, welches die E-Mail Adresse des Benutzers enthält. Allerdings wurde in RFC 2616 beschrieben, dass diese Information nicht ohne die Einwilligung des Nutzers gesendet werden sollte, da dies mit den Datenschutzinteressen des Nutzers in Konflikt stehen könnte. Des Weiteren sollte der Benutzer die Möglichkeit haben, das Feld zu deaktivieren bzw. den Inhalt des Feldes vor dem Absenden der Anfrage zu modifizieren [12]. Auch wenn die E-Mail-Adresse definitiv als personenbezogenes Datum einzustufen ist, ist das *From* Header-Feld dennoch konform mit den gesetzlichen Regelungen, da eine ausdrückliche Einwilligung des Nutzers vorhergehen muss (vgl. § 4 BDSG [1] und Artikel 7 Richtlinie 95/46/EG [10]). Eine mögliche Problematik diesbezüglich besteht auch deswegen nicht, da die Verwendung dieses Header-Feldes heutzutage kaum Verwendung findet. Offen allerdings bleibt die Frage, ob die Kombination sämtlicher Headerfelder so aussagekräftig ist, um eine Person eindeutig zu identifizieren. Auf diese Frage wird im folgenden Kapitel näher eingegangen.

Neben diesen HTTP-Header-Feldern kann ein Webserver weitere Informationen aus einem TCP/IP Datenpakt extrahieren. Hierbei ist vor allem die IP-Adresse zu nennen. Inwie-

doi: 10.2312/NET-2013-02-1_12

fern die IP-Adresse als personenbezogenes Datum angesehen wird, ist aktuell noch umstritten. Datenschützer argumentieren, dass bei einer statisch vergebenen IP-Adresse ein direkter und andauernder Bezug zum Anschlussinhaber möglich ist. Doch auch selbst bei einer dynamischen zugewiesenen IP-Adresse in Kombination mit dem Zeitstempel des Zugriffs wird argumentiert, dass in vielen Fällen mit Hilfe Dritter ein Bezug zu einer bestimmbaren Person hergeleitet werden kann. Diese Auffassung wurde in verschiedenen Gerichtsentscheidungen [14][4], in einem öffentlichen Schreiben des Bundesjustizministerium [7] sowie vom Bundesamt für Sicherheit in der Informationstechnik (BSI) [6] geteilt. Allerdings liegen auch Gerichtsbeschlüsse vor, bei denen die IP-Adresse nicht als personenbezogenes Datum aufgefasst wurde [3]. Ein eindeutiger und gesetzlich haltbarer Beschluss fehlt damit weiterhin. Durch den Vorschlag des Europäischen Parlaments zur Reform der 95/46/EG ist allerdings die Tendenz erkennbar, wonach zukünftig IP-Adressen als personenbezogenes Datum eingestuft werden sollen [11].

3.3 Bewertung des Informationsgehalts

Um die verschiedenen Variablen bezüglich ihres Informationsgehaltes vergleichen zu können, wird die Entropie als Maß der Informationsdichte verwendet. Dabei entspricht die Entropie, die in Bits gemessen wird, dem mittleren Informationsgehalt einer Messvariable. Der Informationsgehalt wiederum hängt von der Auftrittswahrscheinlichkeit der Messvariable ab. Generell gilt, je geringer die Auftrittswahrscheinlichkeit einer Variable ist, desto höher ist der Informationsgehalt und andersherum. Wichtig in diesem Zusammenhang ist zu erwähnen, dass die Entropie verschiedener Variablen nur dann miteinander addiert werden kann, sofern keine statistische Abhängigkeit zwischen den Messvariablen besteht [8].

Die folgenden Informationen beziehen sich auf die Auswertung des Forschungsprojekts Panopticlick der EFF. Die Analyse der Daten basiert auf einem Datensatz von insgesamt über einer Million Klicks auf ihrer Test-Webseite, wovon durch eine Reduktion von redundanten Daten immer noch 470.161 Datensätze übrig geblieben sind.

Tabelle 2: Durchschnittliche Entropie der HTTP-Felder [8, S. 5-17]

Header-Feld	Entropie	Quelle
User-Agent	10,0	HTTP-Header
Accept-Headers	6,09	HTTP-Header
Cookies erlaubt?	0,353	HTTP-Header
Schriftarten	13,9	Flash / Java Applet
Browser Plugins	15,4	JavaScript / AJAX
Bildschirm-Auflösung	4,83	JavaScript / AJAX
Zeitzone	3,04	JavaScript / AJAX
Supercookie Test	2,12	JavaScript / AJAX

Für den *User-Agent* bedeutet dies beispielsweise, dass wenn aus der Gesamtmenge der Datensätze ein Browser zufällig ausgewählt wird, nur jeder 2^{10}-te (=1024) Browser genau die gleiche *User-Agent* Charakteristik aufweist [8]. Der *User-Agent* ist somit ein relativ aussagekräftiges Kriterium bei der Identifikation von Browsern. Neben den gewonnen Informationen über den HTTP-Header sind in dieser Tabelle noch weitere Kriterien aufgelistet, die über Java, Javascript (bzw. AJAX) und Flash gewonnen werden, damit die Werte besser eingeordnet werden können.

Leider enthält die statistische Auswertung von Eckersley keine Aufschlüsselung der einzelnen Accept-Header-Feldern. Stattdessen sind die vier in Tabelle 1 beschriebenen Accept-Header zusammengefasst worden, weshalb eine Aussage zu den einzelnen Feldern nicht möglich ist. Dennoch lässt sich ablesen, dass die Accept-Felder zwar einen nicht zu vernachlässigbaren Anteil zur Identifikation der Browser ausmachen, aber in der Gesamtsumme eher untergehen. Kaum Bedeutung hingegen ist der Abfrage, ob Cookies erlaubt sind oder nicht, zuzumessen. Hierbei ist anzumerken, dass die einzelnen Header-Felder statistische Abhängigkeiten besitzen. Die Abfrage, ob Cookies erlaubt sind oder nicht, ist beispielsweise von den Grundeinstellungen des Web-Browser und somit vom User-Agent abhängig, weshalb die Entropie-Bits nicht miteinander addiert werden dürfen.

Des Weiteren ist aus der Tabelle abzulesen, dass generell die Aussagekraft, die über den HTTP-Header gewonnen werden kann, eher gering ist im Vergleich zu den Möglichkeiten, die JavaScript bzw. AJAX bietet. Allerdings muss an dieser Stelle relativiert werden, dass der HTTP-Header tatsächlich bei jeder Anfrage übertragen wird und nur schwer zu blocken ist. JavaScript hingegen kann in den gängigen Browsern entweder direkt, oder durch diverse Plugins deaktiviert werden. Es ist allerdings auch schon für die Browser Firefox [17] und Chrome [13] ein Plugin vorhanden, welches die Modifikation des HTTP-Anfrage-Headers ermöglicht. Laut Beschreibung sind diese Plugins allerdings primär an Web-Entwickler gerichtet, um die Funktionalität einer Webseite zu testen, wenngleich auch das Thema Schutz der Privatsphäre erwähnt wird.

Welcher Informationsgehalt der IP-Adresse in diesem Zusammenhang beizumessen ist, geht aus der Arbeit von Eckersley nicht hervor. Dennoch ist festzuhalten, dass es sich bei der IP-Adresse um eine Variable handelt, die sich im Regelfall täglich verändert. Bei den bereits besprochenen Messvariablen sind die Werte hingegen über einen längeren Zeitraum relativ stabil. Diese Eigenschaft ist auch bei der Erstellung eines Browser-Fingerabdrucks erforderlich. Durch die Verwendung von Network Address Translation (NAT) unter IPv4 wird die Zuordnung der IP-Adresse zu einem Browser weiter erschwert. Eine fundierte Aussage über den Informationsgehalt der IP-Adresse ist daher nur schwer möglich. Eine statische und fest zugewiesene IP-Adresse würde hingegen, in Verbindung mit den anderen Messvariablen, die Wiedererkennung der Geräte aus diesem Internetanschluss deutlich vereinfachen.

4. MAßNAHMEN UND MODELLE ZUM SCHUTZ DER PRIVATSPHÄRE

Um ein Mindestmaß an Schutz der Privatsphäre eines Internetnutzers zu gewährleisten, existieren verschiedene Ansätze. Dabei hängen die Beweggründe als auch die Möglichkeiten stark von der jeweiligen Rolle ab. Im Folgenden werden dazu die Perspektiven des Benutzers, des Software-Entwicklers als auch die eines Webseiten-Betreibers näher betrachtet.

doi: 10.2312/NET-2013-02-1_12

4.1 Aus Sicht eines Internetnutzers

Wem der Schutz der eigenen Privatsphäre wichtig ist, kann selbst dazu beitragen, dass dieser Schutz gewährleistet ist. Dazu bieten alle gängigen Web-Browser verschiedenste Einstellmöglichkeiten an. So kann beispielsweise konfiguriert werden, ob Cookies überhaupt erlaubt und wenn ja, nach welchem Zeitpunkt wieder gelöscht werden sollen. Ebenso kann JavaScript komplett deaktiviert werden. Deutlich komfortabler und flexibler können diese Einstellungen auch über verschiedene Plugins vorgenommen werden, welche bei den meisten Browsern nachinstalliert werden können. Um zusätzlich noch die eigene IP-Adresse zu verschleiern, kann die Verbindung über einen ausländischen Proxy geleitet werden. Noch weitreichender sind die Möglichkeiten, die ein Anonymisierungs-Tool wie beispielsweise Tor bietet [5]. Bei der großen Vielzahl an Plugins gilt es allerdings zu beachten, dass einige Schutzmaßnahmen auch einen kontraproduktiven Einfluss auf die Identifizierbarkeit des Browser haben können. Dies ist vor allem dann der Fall, wenn ein Plugin installiert wurde, das nur von einer geringen Anzahl an Nutzern verwendet wird, wodurch sich der Fingerabdruck des Browsers auf einen kleinen Nutzerkreis eingrenzen lässt [8].

Der Kompromiss, den man mit solchen Schutzmaßnahmen eingeht, ist meist ein Verlust an Komfort. Im Regelfall wird durch die genannten Maßnahmen das Surf-Erlebnis reduziert, da einzelne Webseiten nicht korrekt dargestellt werden können, oder die Latenz der Verbindung erhöht wird.

An dieser Stelle ebenso zu nennen, ist ein verantwortungsvoller Umgang mit den eigenen Daten, die man selbst veröffentlicht. Denn auch die besten Anonymisierungs-Tools und Plugins sind nutzlos, wenn im Internet freiwillig intime Daten preisgegeben werden, wie beispielsweise in Sozialen Medien.

4.2 Maßnahmen für Software-Entwickler

Aus der Perspektive eines Software-Herstellers, egal ob es sich dabei um die Software eines Browsers oder die einer Browser-Komponente handelt, mag es zunächst wenig Beweggründe geben, sich um den Schutz der Privatsphäre zu kümmern; schließlich resultiert daraus kein direkter Nutzen. In diesem Zusammenhang ist vor allem die Übertragung von detaillierten Versionsnummern in den HTTP-Header-Feldern gemeint, die dann laut Eckersley bei der Erstellung eines Device-Fingerprints zu Lasten der Privatsphäre genutzt werden können [8]. Die Software-Entwickler selbst hingegen können von der Angabe dieser Mikroversionen profitieren, beispielsweise bei der Analyse von Fehlverhalten einzelner Versionen oder auch bei der Erstellung einer Statistik bezüglich der Nutzungsverteilung der einzelnen Versionen der Software.

Dennoch gibt es auch Gründe für einen Software-Hersteller, bei der Entwicklung der Produkte auf Datenschutzvorkehrungen zu achten. Wenn beispielsweise durch verschiedenste Maßnahmen die Privatsphäre der Benutzer besonders geschützt wird, könnte dies das Vertrauen der Benutzer in die Produkte stärken, wodurch die Reputation des Unternehmens steigt, was wiederum zu steigenden Nutzungszahlen führen kann. Ebenso ist es denkbar, dass Software-Entwickler gesetzlich aufgefordert werden, die Privatsphäre der Nutzer von vornherein zu schützen. Dieses Szenario könnte sogar schon bald durch eine mögliche Reform der Richtlinie 95/46/EG Realität werden [11].

4.3 Modelle zur Datenspeicherung seitens des Webseiten-Betreibers

Wenn einem Webseiten-Betreiber die Privatsphäre der Nutzer am Herzen liegt, dann gibt es eine simple Methode dies zu realisieren: Indem gar keine Daten gespeichert werden. Dies beinhaltet den Verzicht der Nutzung von Cookies, Logging und im Idealfall auch eine Vermeidung von JavaScript. Derzeit lässt sich solch einen Zusammenschluss mehrerer Webseiten finden, die dem Benutzer versprechen, keine personenbezogene Daten zu speichern. Ebenso dürfen auch keine externen Dienste wie Statistiken oder Werbung eingebunden werden, die diese Regeln missachten. Jeder Webseiten-Betreiber, der an dieser Aktion teilnehmen will und die Bedingungen erfüllt, darf schlussendlich ein Gütesiegel auf der eigenen Webseite einbinden [24].

Dennoch lässt es sich in manchen Situationen nicht vermeiden, auch personenbezogene Daten zu speichern. Dies ist beispielsweise dann der Fall, wenn die Daten - im Einklang mit den gesetzlichen Bestimmungen - zu Zwecken der Abrechnung über die Nutzungsdauer hinaus gespeichert werden müssen. Weiterhin fordert das Gesetz eine Anonymisierung oder Pseudonymisierung der Daten, sofern dies nicht mit einem unverhältnismäßig hohen Aufwand verbunden ist.

Zur Veranschaulichung werden an dem Beispiel einer medizinischen Tabelle verschiedene Modelle zur Datenspeicherung vorgestellt.

Tabelle 3: Medizinische Krankheitstabelle

Name	Geburts-datum	Ge-schlecht	PLZ	Krankheit
Hans Maier	05.03.82	M	81247	Diabetes
Max Müller	13.08.82	M	81252	Diabetes
Ida Schmitt	01.08.81	W	81246	Tumor
Anna Meier	12.01.82	W	81247	HIV

Das sensitive Attribut in dieser Tabelle ist die Krankheit. So wie die Tabelle hier vorliegt, steht die Erkrankung in einem direkten Bezug zu einer Person. Aber auch eine Anonymisierung, bei der lediglich die Namen aus dieser Tabelle gestrichen werden, ist nicht ausreichend. Denn auch die Attribute Geburtsdatum, Geschlecht und PLZ sind in diesem Beispiel ausreichend, um eine Personen eindeutig identifizieren zu können. Eine solche Kombination von Attributen wird auch als *quasi-identifier* bezeichnet [21]. Der Name hingegen ist ein Schlüsselattribut, da über diesen eine Person direkt identifiziert werden kann.

4.3.1 k-anonymity

Um eine solche indirekte Identifikation mittels *quasi-identifier* zu verhindern, kann das Modell *k-anonymity* eingesetzt werden. Dieses Modell besagt, dass für jeden Datensatz und für jede beliebige Kombination aus *quasi-identifiern* mindestens k-1 andere ununterscheidbare Datensätze existieren müssen [15].

Mit dem Wissen über Geburtsdatum, Geschlecht und PLZ einer einzelnen Person ist es mit den Informationen aus Ta-

doi: 10.2312/NET-2013-02-1_12

Tabelle 4: Anonymisierte Tabelle mit k=2

Geburts-datum	Ge-schlecht	PLZ	Krankheit
..82	M	812**	Diabetes
..82	M	812**	Diabetes
..8*	W	8124*	Tumor
..8*	W	8124*	HIV

belle 4 generell nicht mehr möglich, auf die Krankheit einer Person zu schließen, da mindestens zwei Personen in Frage kommen. Je höher der Wert von k ist, desto besser ist die Anonymisierung eines Individuums. Mathematisch ausgedrückt bedeutet dies, dass durch die Kenntnis eines *quasi-identifier* eine ausgewählte Person nur mit einer Wahrscheinlichkeit von $\frac{1}{k}$ bestimmt werden kann. Allerdings existieren verschieden Angriffsmöglichkeiten gegen dieses Modell. Beispielsweise kann mit dem Wissen, dass die Person Hans Maier in dieser Tabelle gelistet ist und Geburtsdatum, PLZ und Geschlecht bekannt sind, aus Tabelle 4 abgelesen werden, dass Hans Maier an Diabetes leidet, da dies für beide in Frage kommenden Personen gilt [15].

4.3.2 ℓ-diversity

Um genau die beschriebene Schwachstelle von *k-anonymity* anzugehen, wurde das Modell *ℓ-diversity* eingeführt, das einen besseren Schutz bietet. Dabei wird zunächst zwischen sensitiven und nicht-sensitiven Attributen unterschieden. Weiterhin muss neben den Bedingungen von *k-anonymity* zusätzlich noch gelten, dass für jede mögliche Kombination mindestens ℓ-verschiedene sensitive Attribute vorhanden sind. Wenn Hans Maier beispielsweise statt Diabetes eine andere Krankheit hätte, dann würde Tabelle 4 die Eigenschaften von *ℓ-diversity* (mit ℓ=2) erfüllen. Allerdings stößt auch dieses Modell an seine Grenzen, wenn die sensitiven Attribute in den Datensätzen unausgeglichen sind oder auch wenn ein sensitives Attribut lediglich mit Ja/Nein beantwortet werden kann. Unter diesen Umständen kann der Datensatz im besten Fall lediglich die Bedingungen für ℓ=2 erfüllen [16].

4.3.3 t-closeness

Das Konzept *t-closeness* ist ein weiteres Modell zur Anonymisierung von Daten, das auf *ℓ-diversity* aufbaut. Zunächst bezeichnen wir eine Menge an Tupeln, die sich durch einen *quasi-identifier* nicht unterscheiden lassen, als *q*-Block*. Dabei wird versucht, den Gewinn an Wissen durch die anonymisierten Daten möglichst gering zu halten. Dies wird in diesem Modell dadurch sichergestellt, indem die Wahrscheinlichkeitsverteilung der sensitiven Attribute jedes *q*-Blocks* nur eine maximale Distanz t zu der Verteilung der sensitiven Attribute in der gesamten Tabelle enthält. Die Distanz zweier Wahrscheinlichkeitsverteilungen kann wiederum mathematisch berechnet werden, beispielsweise mit Hilfe der Kullback-Leibler-Divergenz. Ziel dieses Modells ist es, den Wert von t möglichst gering zu halten. Die gesamte Tabelle besitzt genau dann die Eigenschaft *t-closeness*, wenn für jeden *q*-Block* die Eigenschaft *t-closeness* erfüllt ist [15]. Konkret bedeutet dies also, dass der Wert von t genau dann klein ist, wenn die prozentuale Verteilung der sensitiven Attribute für jeden *q*-Block* möglichst identisch ist.

Wenn die Verteilung der sensitiven Attribute innerhalb der *q*-Blöcke* hingegen sehr verschieden ist, könnte wie folgt ein Angriff durchgeführt werden. Angenommen eine Tabelle enthält ein sensitives Attribut, das die Information beinhaltet, ob eine Person an HIV erkrankt ist oder nicht. Dabei seien 99% der Personen gesund und 1% habe die Krankheit HIV. In einem der *q*-Blöcke* allerdings ist die Verteilung der gesunden und erkrankten Personen ausgeglichen, also im Verhältnis von 50:50. Wenn die *quasi-identifier* einer gesuchten Person auf diesen *q*-Block* zutreffen, dann kann daraus geschlossen werden, dass die gesuchte Person mit einer Wahrscheinlichkeit von 50% HIV hat. Damit erhält ein potenzieller Angreifer einen nicht vernachlässigbaren Informationsgewinn über eine einzelne Personen gegenüber der Allgemeinheit. Würde die Tabelle hingegen die Eigenschaft *t-closeness* mit einem kleinen Wert für t erfüllen, wäre ein solcher Angriff nicht möglich beziehungsweise der Wissensgewinn wäre deutlich geringer, da ein solch abweichender *q*-Block* gar nicht vorkommen dürfte.

5. VERWANDTE ARBEITEN

Während in dieser Arbeit der Fokus auf den rechtlichen Bestimmungen zum Thema Datenschutz liegt, wurden in der Seminararbeit von Thomas Pieronczyk die technischen Hintergründe zur Erstellung der Browser-Fingerabdrücke näher erläutert [19]. Diese Arbeit basierte ebenfalls auf den statistischen Auswertungen von Eckersley [8], in der eine ausführliche Beschreibung der Datensammlung sowie eine Deutung der Ergebnisse enthalten ist.

Aus aktuellem Anlass ebenfalls sehr interessant ist in diesem Zusammenhang das Projekt von Henning Tillmann, der im Rahmen seiner Diplomarbeit selbst Untersuchungen zur Identifizierbarkeit von Web-Browsern durchführt. Die Auswertung des Projekts lag bei der Entstehung dieser Arbeit allerdings noch nicht vor, soll aber im Frühjahr 2013 veröffentlicht werden [22].

Eine ausführliche Erläuterung, der in dieser Arbeit vorgestellten Modelle zur Anonymisierung der Daten, wird in dem Artikel von Latanya Sweeney [21] sowie in Referenz [15] sehr gut beschrieben und mit zahlreichen Beispielen veranschaulicht. Dabei werden auch die mathematischen Hintergründe ausgiebig erklärt.

6. ZUSAMMENFASSUNG UND AUSBLICK

Trotz der expliziten Definition von personenbezogenen Daten in den Gesetzesbüchern bestehen nach wie vor kontroverse Meinungen bezüglich der Auslegung dieser Definition. Die größten Differenzen beziehen sich auf die Fragestellung, inwiefern aus den übertragenen Daten eine bestimmte oder bestimmbare Person ableitbar ist.

Die Analyse der in dem HTTP-Header enthaltenen Feldern hat gezeigt, dass diese vielmehr nützlich als schädlich und in der Praxis alleine kaum ausreichend sind, um ein effektives Device-Fingerprinting zu ermöglichen. Deutlich kritischer und umfangreicher ist der Informationsgehalt der mittels Java, Flash und JavaScript gewonnen werden kann, wie Eckersley gezeigt hat [8]. Ebenfalls als bedenklich ist die IP-Adresse einzustufen. Denn mit Hilfe dieser ist prinzipiell die Möglichkeit gegeben, eine Verbindung zwischen den Daten und einer bestimmbaren Person herzustellen.

doi: 10.2312/NET-2013-02-1_12

Die verschiedenen Maßnahmen zum Schutz der Privatsphäre haben verdeutlicht, dass auf Seiten des Benutzers am meisten Potenzial vorhanden ist, um die eigene Privatsphäre zu schützen – auch wenn dies meist mit einem Verlust an Komfort verbunden ist. Allerdings gilt zu beachten, dass einige Maßnahmen auch einen kontraproduktiven Effekt haben können. Ebenfalls interessant sind die vorgestellten Modelle, die einem Web-Hoster zur Verfügung stehen, um bei der Speicherung von personenbezogenen Daten eine bestmögliche Anonymisierung zu ermöglichen.

Mit Spannung zu erwarten ist die Entwicklung der gesetzlichen Beschlüsse, denn nur wenn dieser Grundstein gelegt ist, kann darauf aufbauend ein erhöhtes Maß an Privatsphäre im Internet sichergestellt werden. Dazu müssen aber zunächst die Menschen bezüglich der Bedeutung der Privatsphäre sensibilisiert werden. Denn nur wenn der Wunsch und die Forderung seitens der Bürger zu mehr Datenschutz besteht, ist eine Reaktion der Legislative denkbar.

7. LITERATUR

[1] Bundesdatenschutzgesetz in der Fassung der Bekanntmachung vom 14. Januar 2003 (BGBl. I S. 66), das zuletzt durch Artikel 1 des Gesetzes vom 14. August 2009 (BGBl. I S. 2814) geändert worden ist. http://www.gesetze-im-internet.de/bundesrecht/bdsg_1990/gesamt.pdf, Dec. 1990.

[2] Telemediengesetz vom 26. Februar 2007 (BGBl. I S. 179), das zuletzt durch Artikel 1 des Gesetzes vom 31. Mai 2010 (BGBl. I S. 692) geeändert worden ist. http://www.gesetze-im-internet.de/bundesrecht/tmg/gesamt.pdf, Feb. 2007.

[3] Amtsgericht München. Urt. vom 30.09.2008 - Az. 133 C5677/08. http://tlmd.in/u/524.

[4] Amtsgericht Wuppertal. Urt. vom 03.04.2007 - 29 Ds 70 Js 6906/06. http://www.jurpc.de/jurpc/show?id=20080110.

[5] J. Appelbaum. The Tor Project. https://www.torproject.org/about/overview.html.en. Abgerufen am 8 Februar 2013.

[6] Bundesamt für Sicherheit in der Informationstechnik. Datenschutzgerechtes E-Government. In *E-Government-Handbuch*, pages 12–19. Bundesanzeiger, Köln, 2005.

[7] Bundesministerium der Justiz. R B 3 - zu 4104/8 - 1 -R5 39/2008. http://www.daten-speicherung.de/data/bmj_2009-02-02.pdf.

[8] P. Eckersley. How unique is your web browser? In *Proceedings of the 10th international conference on Privacy enhancing technologies*, PETS'10, pages 1–18, Berlin, Heidelberg, 2010. Springer-Verlag.

[9] Electronic Frontier Foundation. https://panopticlick.eff.org/. Abgerufen am 05 Januar 2013.

[10] EU-Parlament. Richtlinie 95/46/EG. *Amtsblatt*, (L 281):31–50, Oct. 1995. Richtlinie 95/46/EG des Europäischen Parlaments und des Rates vom 24. Oktober 1995 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr.

[11] Europäische Kommision. Vorschlag für Verordnung des Europäischen Parlaments und des Rates zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr (Datenschutz-Grundverordnung), Jan. 2012.

[12] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, IETF, June 1999.

[13] Google Chrome. Change HTTP Request Header. https://chrome.google.com/webstore/detail/change-http-request-heade/ppmibgfeefcglejjlpeihfdimbkfbbnm. Abgerufen am 10 Februar 2013.

[14] Landgericht Berlin. Urt. vom 06.09.2007 - 23 S 3/07, MMR 2007, 799-800. http://www.daten-speicherung.de/data/Urteil_IP-Speicherung_2007-09-06.pdf.

[15] N. Li, T. Li, and S. Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, Apr. 2007.

[16] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *TKDD*, 1(1), 2007.

[17] Mozilla Corporation. Modify Headers. https://addons.mozilla.org/en-us/firefox/addon/modify-headers/. Abgerufen am 15 Dezember 2012.

[18] N. Nottingham and J. Mogul. HTTP Header Field Registrations. RFC 4229, IETF, Dec. 2005.

[19] T. Pieronczyk. Device Fingerprinting mit dem Web-Browser. In G. Carle and C. Schmitt, editors, *Proceedings of the Seminars Future Internet (FI), Innovative Internet Technologies and Mobile Communication (IITM) and Aerospace Networks (AN)*, volume NET-2012-08-1, pages 23–29, Aug. 2012.

[20] L. Sweeney. Uniqueness of Simple Demographics in the U.S. Population, 2000.

[21] L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, Oct. 2002.

[22] H. Tillmann. Browser Fingerprinting. http://bfp.henning-tillmann.de/. Abgerufen am 5 Januar 2013.

[23] United Nations. The Universal Declaration of Human Rights. 1948.

[24] Wir speichern nicht! http://www.wirspeichernnicht.de/. Abgerufen am 19. Dezember 2012.

doi: 10.2312/NET-2013-02-1_12

ISBN 3-937201-33-5
DOI 10.2313/NET-2013-02-1

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