AnonDrop – Räumlich begrenzte anonyme Informationsverbreitung

Alexander Zeier · Andreas Heinemann

Hochschule Darmstadt
Fachbereich Informatik
alexander.zeier@mailbox.org
andreas.heinemann@h-da.de

Zusammenfassung

Opportunistische Netze bieten ein alternatives Kommunikationssystem in Situationen, in denen ein repressiver Staat die klassische Internetkommunikation filtert oder ganz unterbindet. *AnonDrop* erlaubt hier eine räumlich begrenzte Kommunikation, die mittels dynamischer Netzadressen (MAC und IP) und weiterer Schutzmaßnahmen Angriffen auf die Identifizierung von Knoten überwiegend standhält. Auf Basis von Android Smartphones wurde ein Prototyp realisiert, der bei ersten Last- und Mobilitätstests zufriedenstellende Ergebnisse zeigt.

1 Einführung

In der heutigen Zeit ist die Verteilung und Beschaffung von Informationen über das Internet ein alltäglicher Vorgang. Viele Menschen nutzen das Internet, um ihre Meinung mit anderen zu teilen oder sich politisch oder anderweitig motiviert zu organisieren. In der Vergangenheit ist es jedoch bereits mehrfach vorgekommen, dass ein repressiver Staat den Zugang zum Internet für das eigene Land beschränkt hat, um solche Absprachen zu unterbinden. So hat z.B. erstmals die ägyptische Regierung im Januar 2011 das Internet sowie das Mobilfunknetz im eigenen Land abgeschaltet [Krem11]. Auch Libyen und Syrien [Soko11, Stö12] haben später zu diesem Mittel gegriffen, um Absprachen und den politischen Meinungsaustausch innerhalb der Bevölkerung zu erschweren.

In solchen Fällen müssen neue Wege zur Kommunikation gefunden werden. Ein vielversprechender Ansatz sind Opportunistische Netzwerke, eine Unterklasse der *Delay Tolerant* Netzwerke [Fall03]. Jedoch ist es wichtig, diese Kommunikationswege anonym zu gestalten, um nicht die individuelle Freiheit, Unversehrtheit oder sogar das eigene Leben in Gefahr zu bringen [LiHu09].

Vor diesem Hintergrund beschreibt die vorliegende Arbeit *AnonDrop*, ein räumlich begrenztes Opportunistisches Netzwerk zur Verbreitung von Informationen in Textform mit einem besonderen Fokus auf der Anonymität der Teilnehmer. *AnonDrop* besteht aus zwei Knotentypen: Mobile Knoten (AD Mobile Peer), die von Nutzern getragen werden und die zu verbreitende Informationen speichern und fest installiere Knoten (AD Fix Peer), die – im Idealfall versteckt – an öffentlichen Plätzen verteilt werden und Informationen mit mobilen Knoten austauschen. Abbildung 1 illustriert die Knotentypen.

P. Schartner (Hrsg.) · D•A•CH Security 2016 · syssec (2016) 241-252.

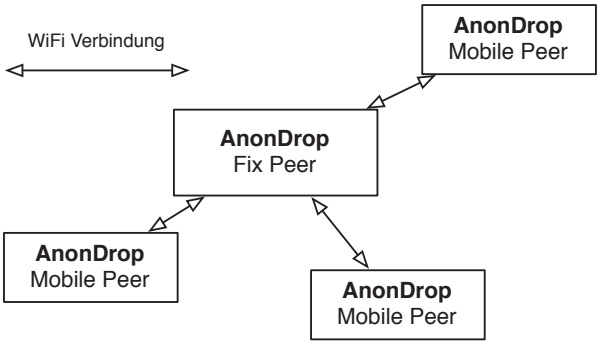

Abb. 1: *AnonDrop* Knotentypen

Feste Knoten stellen zur Informationsverbreitung räumlich begrenzt ein WiFi Netzwerk zur Verfügung. Mobile Knoten verbinden sich mit diesem Netzwerk, sobald sie in Reichweite des WiFi Signals sind. Da die festen Knoten keine Verbindung zum Internet haben, können diese nicht zentral (z.B. über eine DDOS-Attacke) angegriffen oder abgeschaltet werden. Ein Angriff wäre mit der physischen Präsenz des Angreifers und somit mit steigender Anzahl der AD Fix Peers mit hohem Ressourceneinsatz verbunden.

Im Zuge dieser Arbeit wurde ein Prototyp realisiert, dessen mobile Knoten auf Android Smartphones und dessen feste Knoten auf dem Einplatinencomputer Raspberry Pi [Ras16] basieren.

Anonymität der mobilen Knoten wird durch dynamische und temporäre Vergabe von MAC-[1] und IP-Adressen und durch weitere Maßnahmen ab Netzwerkschicht 4 erreicht (vgl. Abschnitt 4). Damit wird die Identifizierung der ursprünglichen Quelle einer Nachricht, d.h. des Urhebers, anhand von technisch bedingten Kommunikationsmerkmalen unterbunden.

Der weitere Teil dieser Arbeit ist wie folgt strukturiert: In Abschnitt 2 findet eine Einordnung von *AnonDrop* in die Literatur statt und verwandte Arbeiten werden kurz diskutiert. Abschnitt 3 stellt die Architektur und den Prototypen von *AnonDrop* vor. In Abschnitt 4 wird auf das Kommunikationsmodell von *AnonDrop* eingegangen. Abschnitt 5 diskutiert potentielle Angriffe und bewertet diese vor dem Hintergrund implementierter Schutzmaßnahmen. In Abschnitt 6 werden Ergebnisse erster Experimente zur Leistungsfähigkeit des Prototypen vorgestellt. Abschnitt 7 fasst die wesentlichen Aspekte dieser Arbeit abschließend zusammen und schließt mit einem Ausblick zu weiteren, noch zu bearbeitenden Fragestellungen.

2 Verwandte Arbeiten

Die starke und weiter zunehmende Verbreitung von Smartphones weltweit [Stat16] fördert das Interesse an Opportunistischen Netzwerken, die Smartphones als Netzknoten verwenden: Trifunovic et al. [TKHL15] nutzen die Tethering Funktion von Smartphones zur Etablierung eines Opportunistischen Netzwerkes; Kärkkäinen et al. [KaP12] verwenden öffentliche WLAN HotSpots, um ein Opportunistisches Netzwerk auf Basis von Paketweiterleitungen auf Netzwerkschicht 2 (Link Layer) zu realisieren. Jedoch findet die Anonymität der Knoten[2] noch zu wenig Betrachtung. *AnonDrop* greift hier Überlegungen von Heinemann et al. [HeSt10] auf, die in [HeKM08] untersucht haben, wie gut sich Informationen mittels *single hop* Kommunikation

[1] MAC-Adressen werden schon heute aktiv zum Tracking von Nutzern eingesetzt. Siehe [WeGG14].
[2] Und damit der Nutzer.

in unterschiedlichen Szenarien verbreiten. Diese Art der Kommunikation erlaubt das periodische Ändern der MAC-Adresse, was allerdings nur theoretisch beschrieben wurde. Ähnlich beschreiben Lei et al. [LeHV07] den Austausch von MAC-Adressen zwischen aktiven Knoten eines WiFi Netzes, um Ortsinformationen aktiver Knoten zu verschleiern. Auch hier wurde das Verfahren nur theoretisch beschrieben.

Nach unserem Wissen ist dies die erste Arbeit, die demonstriert, dass ein periodisches, randomisiertes Ändern der MAC-Adresse auf Smartphones zur anonymen *single hop* Kommunikation in Opportunistischen Netzwerken in der Praxis funktioniert.

3 AnonDrop – Architektur und Prototyp

Wie in Abbildung 1 zu sehen ist, besteht die Anwendung aus einem AD Fix Peer und mehreren AD Mobile Peers. Auf dem AD Fix Peer müssen, neben der *AnonDrop* Software, auch ein Programm zur Bereitstellung eines Access Points (z.B. *hostapd*), ein DHCP Server (z.B. *dnsmasq*) und ein Webserver (z.B. *apache*) installiert sein.

Als Fix Peer verwenden wir den Einplatinencomputer Raspberry Pi 2 Model B. Als WLAN Komponente wird der WLAN-Stick Edimax EW-7811Un verwendet. Der Prototyp ist in Java implementiert.

Der Prototyp des AD Mobile Peers wurde auf Smartphones der Marke Nexus 5 und HTC Desire HD unter Android realisiert. Für die MAC-Adressänderung benötigt der AD Mobile Peer root-Rechte. Hierzu wurde auf beiden Geräten CYANOGENMOD [Cya16] installiert.

Abbildung 2 zeigt die Klassen des AD Mobile Peer Prototyps. `MainActivity` ist die zentrale Komponente der Anwendung und u.a. für die Interaktion mit dem Nutzer und für die Erzeugung der GUI zuständig. Sobald eine Verbindung zu einem AD Fix Peer besteht, wird ein `TCPClient`-Thread erstellt. Dieser baut eine TCP-Verbindung zum Fix Peer auf, über welche die Nachrichten vom Mobile zum Fix Peer gesendet werden. Zusätzlich erstellt er einen weiteren Thread, den `Receiver`. Dieser öffnet einen UDP Socket und wartet in einer Schleife auf eingehende UDP Nachrichten, die vom Fix an den Mobile Peer direkt oder als Broadcast gesendet werden. Sobald die Verbindung abbricht, werden die Threads zerstört und bei einer neuen Verbindung erneut erzeugt.

Die Kommunikation beider Threads zu der Hauptklasse findet über eine `Handler`-Klasse statt. Diese nimmt u.a. die vom Server empfangenen Nachrichten entgegen und verarbeitet sie.

Die beiden Klassen `ScanNetworkBroadcastReceiver` und `ScanNetworkService` dienen dazu, dass der Mobile auch bei ausgeschaltetem Bildschirm in kurzen Intervallen das Netzwerk nach einem AD Fix Peer durchsuchen kann.

Zur Änderung der MAC-Adresse sowie des Hostnamens trennt die Anwendung zuerst die Verbindung zum *Access Point*[3] und öffnet anschließend eine root-Shell, in der dann die beiden Befehle

```
busybox ifconfig wlan0 hw ether XX:XX:XX:XX:XX:XX
setprop net.hostname $(echo $(strings /dev/urandom
 | grep -o '[[:alnum:]]' | head -n 12 | tr -d '\n'))
```

[3] Dies ist nicht für alle Smartphones notwendig.

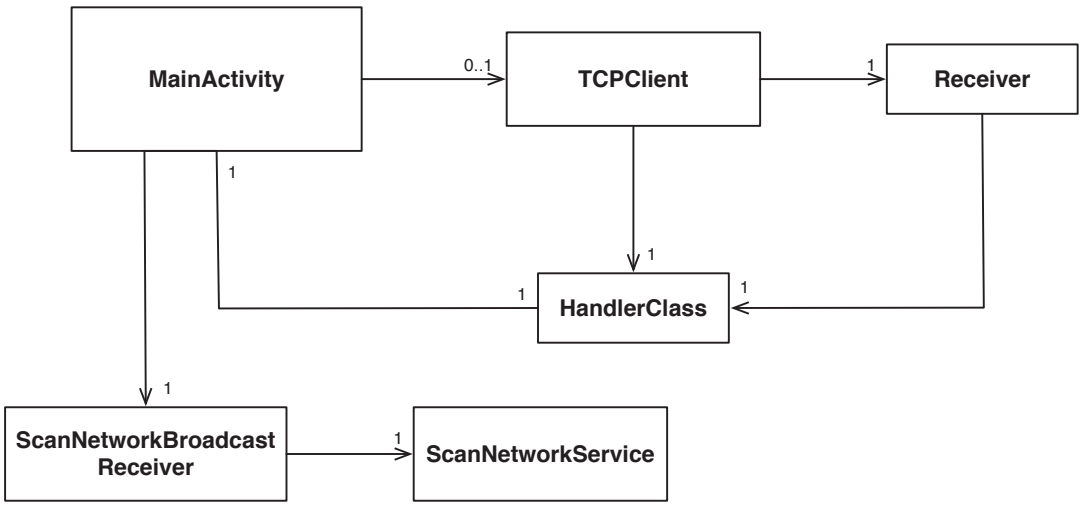

Abb. 2: AD Mobile Peer Klassendiagramm

ausgeführt werden. Der erste Befehl ändert die MAC-Adresse, wobei `XX:XX:XX:XX:XX:XX` für eine zufällig generierte MAC-Adresse steht. Der zweite Befehl ändert den Hostnamen des Smartphones in eine zufällig generierte Zeichenfolge aus zwölf alphanumerischen Zeichen. Zur Generierung der MAC-Adresse wird die `SecureRandom`-Klasse verwendet.

Abbildung 3 zeigt die Klassen des AD Fix Peer Prototyps. Bei jedem neuen Verbindungsaufbau durch einen AD Mobile Peer wird ein neuer `ServerThread` für diese Verbindung erzeugt. Dieser bleibt so lange bestehen, bis die Verbindung durch den Client getrennt wird, z.B. wenn dieser außer Reichweite kommt oder die MAC-Adresse geändert wird.

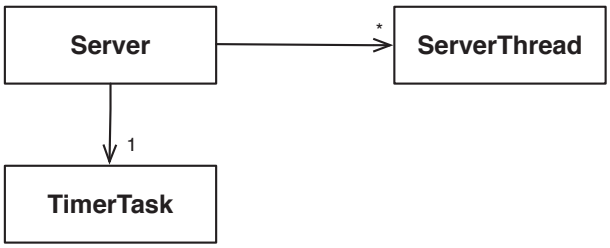

Abb. 3: AD Fix Peer Klassendiagramm

Mit einem AD Fix Peer können sich theoretisch beliebig viele AD Mobile Peers verbinden, um Nachrichten zu senden und zu empfangen. Die Anzahl der möglichen Verbindungen ist jedoch durch die Leistung des AD Fix Peers begrenzt.

Durch die `TimerTask`-Klasse werden die AD Mobile Peers in regelmäßigen Intervallen zur synchronen MAC-Adressänderung aufgefordert.

Die Verbindung mehrerer AD Fix Peer untereinander ist für diese Arbeit nicht vorgesehen, wäre jedoch eine Möglichkeit, die Verbreitungsreichweite der Nachrichten zu erhöhen. Theoretische Überlegungen hierzu finden sich in [HeSt10].

4 Anonyme Kommunikation zwischen AD Knoten

In diesem Abschnitt wird die Kommunikation zwischen einem AD Mobile Peer und einem AD Fix Peer detaillierter beschrieben. Es werden insbesondere Maßnahmen erläutert, die auf die Abwehr möglicher Angriffe (siehe Abschnitt 5) zur Identifizierung eines AD Mobile Peers und somit nachgelagert zur Identifizierung des Nutzers abzielen.

Um die Anwendung optimal nutzen zu können, werden viele AD Fix Peers großflächig in einem Gebiet verteilt, um eine möglichst große Flächenabdeckung zu erreichen und weniger anfällig gegen die gezielte Beseitigung einzelner AD Fix Peers zu sein. Solch ein AD Fix Peer ließe sich auch mobil betreiben. Dazu müsste der Server durch eine Batterie oder einen Akku mit Strom versorgt werden, damit der Nutzer diesen transportieren kann.

Die Kommunikation zwischen einem AD Mobile Peer und einem AD Fix Peer läuft nach folgendem Muster ab: Der AD Fix Peer verbreitet ein WiFi Signal mit der SSID AnonDrop-`xxx` (hierbei ist `xxx` eine eindeutig gewählte ID). Kommt ein AD Mobile Peer in Reichweite des Netzes, so tritt er diesem Netz bei. Er fordert über DHCP eine IP-Adresse an und ist jetzt in der Lage, Textnachrichten vom AD Fix Peer zu empfangen. Bevor er selbst Nachrichten sendet, wartet er allerdings zunächst auf ein spezielles UDP Broadcast Paket, welches ihn und alle anderen AD Mobile Peers dazu auffordert, die MAC-Adresse zufällig zu ändern[4].

Nun lädt sich der AD Mobile Peer via HTTP einen vom AD Fix Peer generierten öffentlichen RSA-Schlüssel. Damit sendet er verschlüsselt einen symmetrischen, selbst gewählten AES-Schlüssel an den Fix Peer, mit welchem fortan die Nachrichten des AD Mobile Peers verschlüsselt werden. Nach jeder MAC-Adressänderung muss sich der AD Mobile Peer erneut mit dem Netzwerk verbinden und einen neuen AES-Schlüssel an den AD Fix Peer senden.

Zusätzlich zu den Nachrichten des Nutzers sendet ein AD Mobile Peer automatisch in unregelmäßigen Abständen sogenannte Phantomnachrichten an den AD Fix Peer. Dabei handelt es sich um Nachrichten mit zufälliger Länge und zufälligem Inhalt.

Für den Empfang von Textnachrichten implementiert *AnonDrop* zwei Mechanismen, die in Kombination genutzt werden. Tritt ein AD Mobile Peer dem Netz bei, so fordert er alle Nachrichten der Vergangenheit an, die bereits mittels UDP Broadcast verbreitet wurden. Diese Anfrage kann er limitieren, z.B. auf die letzten 100 Nachrichten. Der AD Fix Peer übermittelt diese dann via UDP Unicast an den anfordernden Knoten.

Daneben versendet der AD Fix Peer, nachdem er eine gewisse Anzahl an Nachrichten von unterschiedlichen mobilen Knoten eingesammelt hat, diese wieder als UDP Broadcast an alle aktiven AD Mobile Peers in Reichweite. Der Schwellwert liegt aktuell bei 5 Nachrichten.

Neben temporären MAC- und IP-Adressen müssen weitere Maßnahmen getroffen werden, um die Identifikation eines AD Mobile Peers zu unterbinden. So müssen DHCP Requests den Empfehlungen von [HuMK16] folgen, um auf diesem Wege keine Informationen preiszugeben, die zur Wiedererkennung führen könnten. Ebenfalls wird der *hostname* des AD Mobile Peers mit Ändern der MAC-Adresse zufällig gewählt. Schließlich informiert der AD Fix Peer über die aktuelle Anzahl aktiver AD Mobile Peers. Im Sinne der k-Anonymität [Swee02] nimmt ein AD Mobile Peer erst dann aktiv an der Kommunikation teil, wenn die aktuelle Anzahl der Peers den Wert 10 übersteigt. Diesen Schwellwert kann ein AD Mobile Peer ändern.

[4] Der seltene Fall, dass sich zwei Knoten zufällig die gleiche MAC-Adresse wählen, wird in der aktuellen Implementierung außer Acht gelassen.

5 Mögliche Angriffe und Schutzmaßnahmen

Im Folgenden werden die im vorherigen Abschnitt beschriebenen Schutzmaßnahmen anhand der zu verhindernden Angriffe diskutiert.

Angriff 1: Wenn ein AD Mobile Peer seine MAC-Adresse ändert, lässt sich dies bei Beobachtung des Nachrichtenverkehrs durch das Verschwinden der alten und Hinzukommen der neuen Adresse nachverfolgen. Dadurch könnte ein Angreifer MAC-Adressen in Beziehung setzen.

Aus diesem Grund stößt der AD Fix Peer mittels Versenden eines UDP Broadcast an alle Peers die gleichzeitige Adressänderung auf allen AD Mobile Peers an.

Angriff 2: Wenn sich ein Angreifer so positioniert, dass sich nur ein einzelner AD Mobile Peer in seiner Reichweite befindet, so könnte er eine von diesem Gerät gesendete Information einer MAC-Adresse zuordnen.

Aus diesem Grund werden die Nachrichten, die der AD Mobile Peer an den AD Fix Peer sendet, symmetrisch verschlüsselt.

Angriff 3: Positioniert sich ein Angreifer in der Nähe des AD Fix Peers, so kann er anhand der eingehenden, verschlüsselten Nachrichten abschätzen, wie viele Nutzer aktiv *AnonDrop* verwenden, um neue Nachrichten zu verbreiten.

Um zu verschleiern, wie viele Nutzer die Anwendung aktiv und wie viele sie nur passiv nutzen, werden von jedem AD Mobile Peer automatisch Phantomnachrichten versendet.

Angriff 4: Befindet sich ein Angreifer in Reichweite eines einzelnen AD Mobile Peers und des AD Fix Peers, so kann er auch die unverschlüsselten Nachrichten lesen, die vom AD Fix Peer nach dem Empfang zurück an alle AD Mobile Peers gesendet werden und kann so eine verschlüsselte Nachricht der dazugehörigen unverschlüsselten Nachricht zuordnen.

Um dies zu verhindern, werden empfangene Nachrichten vom AD Fix Peer nicht sofort verschickt, sondern erst nach Erhalt einer bestimmten Anzahl an Nachrichten durch verschiedene AD Mobile Peers. Erst wenn dieser Wert erreicht wurde, schickt der AD Fix Peer alle neuen Nachrichten in zufälliger Reihenfolge an die AD Mobile Peers via UDP Broadcast heraus. Es wäre prinzipiell möglich, hier einen *known plaintext*-Angriff durchzuführen, um Nachrichteninhalte mit MAC-Adressen zu verknüpfen. Dies würde aufgrund des hohen Aufwands in Abhängigkeit des Verschlüsselungsalgorithmus und der regelmäßigen, in kurzen Intervallen stattfindenden MAC- und IP-Adressänderungen durch die AD Mobile Peers nicht zur Identifizierung dieser führen.

Angriff 5: Ein Angreifer beobachtet die Personen in Reichweite eines AD Fix Peers und erkennt so, welcher Nutzer eine Nachricht über sein Smartphone eingibt und versendet.

Nachrichten können ohne Nutzerinteraktion versendet werden, d.h., dass die Nachricht vorab verfasst und auf dem AD Mobile Peer gespeichert wird. Sobald sich ein Nutzer in Reichweite eines AD Fix Peers befindet, wird die Nachricht automatisch an diesen gesendet.

Angriff 6: Der Angreifer stellt selbst einen manipulierten AD Fix Peer auf, um so einige oder alle der bisher genannten Sicherheitsmechanismen zu umgehen.

Dieser Angriff kann nur abgemildert werden. Hierzu könnte auf dem AD Fix Peer ein von

a)

```
▸ Frame 220: 326 bytes on wire (2608 bits), 326 bytes captured (2608 bits)
▾ Ethernet II, Src: MS-NLB-PhysServer-12_b8:63:d5:d4 (02:0c:b8:63:d5:d4), Dst: EdimaxTe_f7:0a:45 (80:1f:02:f7:0a:45)
  ▸ Destination: EdimaxTe_f7:0a:45 (80:1f:02:f7:0a:45)
  ▸ Source: MS-NLB-PhysServer-12_b8:63:d5:d4 (02:0c:b8:63:d5:d4)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 192.168.2.62, Dst: 192.168.2.1
▸ Transmission Control Protocol, Src Port: 41102 (41102), Dst Port: 6789 (6789), Seq: 8, Ack: 1, Len: 260
▸ Data (260 bytes)
```

b)

```
▸ Frame 279: 326 bytes on wire (2608 bits), 326 bytes captured (2608 bits)
▾ Ethernet II, Src: 02:a9:41:f2:7b:64 (02:a9:41:f2:7b:64), Dst: EdimaxTe_f7:0a:45 (80:1f:02:f7:0a:45)
  ▸ Destination: EdimaxTe_f7:0a:45 (80:1f:02:f7:0a:45)
  ▸ Source: 02:a9:41:f2:7b:64 (02:a9:41:f2:7b:64)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 192.168.2.19, Dst: 192.168.2.1
▸ Transmission Control Protocol, Src Port: 57500 (57500), Dst Port: 6789 (6789), Seq: 1, Ack: 1, Len: 260
▸ Data (260 bytes)
```

Abb. 4: Datenpakete a) vor der Änderung von MAC- und IP-Adresse und b) nach der Änderung.

einer vertrauenswürdigen, dritten Instanz[5] signiertes Zertifikat hinterlegt werden, welches seine Vertrauenswürdigkeit erhöht. Dies setzt jedoch die Existenz einer bzw. die Kooperation mit einer vertrauenswürdigen Instanz voraus, die an der Konfiguration und Verbreitung von AD Fix Peers und AD Mobile Peers beteiligt ist.

Sollte keine vertrauenswürdige, dritte Instanz vorhanden sein, so kann der AD Mobile Peer zumindest erkennen, ob anhand von UDP Broadcast Paketen eine gleichzeitige MAC-Adressänderung angestoßen wird. Bleibt dies aus, so kann der Nutzer gewarnt werden bzw. sich der Knoten passiv verhalten.

Angriff 7: Ein Angreifer bringt einen manipulierten AD Fix Peer und einen oder mehrere AD Mobile Peers ins Netz ein.

Eine Möglichkeit zur Abwehr eines solchen Angriffs ist uns derzeit nicht bekannt. Bei einer hohen Nutzerzahl und bei an vielen Plätzen ausgelegten AD Fix Peers ist dieser Angriff jedoch auch für den Angreifer mit sehr hohem Aufwand verbunden.

Ein Spezialfall dieses Angriffs ist die *Sybil Attack* [Douc02]. Piro et al. beschreiben Möglichkeiten zur Erkennung einer Sybil Attack [PiSL06]. Indiz für einen solchen Angriff ist eine Ansammlung von MAC-Adressen, die sich gemeinsam bewegen und aufgrund einer gemeinsamen Antenne niemals gleichzeitig senden. Die erfolgreiche Abwehr eines solchen Angriffs ist aufwändig und wird in der aktuellen Implementierung nicht erkannt.

6 Evaluation

Zunächst wurde überprüft, ob sich die implementierte MAC-Adressänderung in den gesendeten Frames eines AD Mobile Peers nachvollziehen lässt. Hierzu wurde die Kommunikation der beteiligten Knoten aufgezeichnet.

[5] Hier kämen Organisationen wie die Electronic Frontier Foundation (www.eff.org) in Frage.

Abbildung 4 zeigt zwei der Pakete, die vor und nach einer MAC-Adressänderung von einem AD Mobile Peer an einen AD Fix Peer gesendet wurden. Zu sehen ist, dass sich sowohl die IP- als auch die MAC-Adresse geändert haben.

Neben diesen Adressen lässt sich der Hostname als weiteres Merkmal zur Identifikation des Clients beobachten. In Abbildung 5 ist zu sehen, dass auch dieser durch die Anwendung geändert wird.

a)

```
18:28:59.154319 IP s1cJ3VGZwBde.56257 > localhost.6789: Flags [P.], seq 1:261, ack 1, win 1369,
 options [nop,nop,TS val 33598217 ecr 481766], length 260
```

b)

```
18:29:20.059319 IP gwIXJrJcNBrb.39944 > localhost.6789: Flags [P.], seq 1:261, ack 1, win 1369,
 options [nop,nop,TS val 33600307 ecr 483855], length 260
```

Abb. 5: Datenpakete a) vor der der Änderung des Hostnamens und b) nach der Änderung.

6.1 Performancetests

Zum Test der Leistungsfähigkeit des AD Fix Peer-Prototyps wurde eine Java-Anwendung geschrieben, welche beliebig viele AD Mobile Peers in Form von Threads simuliert, die in einem zuvor festgelegten Zeitintervall Nachrichten an den AD Fix Peer senden. Der Startzeitpunkt wird dabei zufällig festgelegt, damit sich die einzelnen Sendezeitpunkte gleichmäßig über dieses Intervall verteilen. Für die folgenden Messungen wurde das Intervall auf 30 Sekunden festgelegt.

Die Anzahl der erzeugten AD Mobile Peers wurde für die einzelnen Messungen schrittweise erhöht, um zu testen, wie sich die Anwendung unter wachsender Belastung verhält. Um zu einer realistischen Einschätzung zu kommen, wie viele AD Mobile Peers im Extremfall mit dem AD Fix Peer verbunden sein können, wird von einer Reichweite von 20 Metern für den verwendeten WLAN-Stick in alle Richtungen ausgegangen[6]. Dadurch ergibt sich eine Fläche von rund 1257m^2. Geht man nun von einer Menschendichte von 1 Person/m^2 aus, könnten maximal 1257 AD Mobile Peers mit dem AD Fix Peer verbunden sein. Dieser Wert wurde als obere Schranke für die Simulation gewählt.

Zusätzlich wurde eine für den Test modifizierte Version des AD Mobile Peer auf einem Smartphone gestartet, welche selbst regelmäßig Nachrichten an den AD Fix Peer sendet. Damit galt es zu überprüfen, ob die durch das Smartphone versendeten Nachrichten zum AD Fix Peer und anschließend wieder zurück zum Smartphone gelangen oder ob diese verloren gehen. Jede Messung umfasste das Senden und anschließende Empfangen 20 solcher Nachrichten.

Darüber hinaus wurde durch einen Neustart der modifizierten Version des AD Mobile Peer überprüft, ob auch bei der Anforderung aller gespeicherten Nachrichten vom AD Fix Peer diese komplett an den AD Mobile Peer übertragen wurden.

Tabelle 1 zeigt die Ergebnisse der Messung. Es ist zu sehen, dass auch bei hoher Auslastung alle Nachrichten den AD Fix Peer erreichen und bei der Rücksendung auch vom AD Mobile

[6] Dieser Wert wurde anhand dieses Tests gewählt: http://www.heise.de/ct/artikel/WLAN-Stick-Edimax-EW-7811Un-1901481.html. Zuletzt besucht am 22.01.2016.

Tab. 1: Messungen des Performancetests

#AD Mobile Peers	Gesendete Nachrichten pro Peer per Minute	Erhaltene Nachrichten (in Prozent)	Erhaltene Nachrichten (nach Anforderung aller Nachrichten, in Prozent)
1	2	100%	100%
10	20	100%	100%
20	40	100%	100%
50	100	100%	100%
100	200	95%	50%
250	500	100%	50%
500	1000	100%	0%
700	1400	100%	0%
800	1600	100%	0%
900	1800	100%	0%
1000	2000	100%	0%
1100	2200	-	-
1200	2400	-	-
1257	2514	-	-

Peer problemlos empfangen werden können. Bei der Anforderung aller Nachrichten treten jedoch bereits bei 100 zusätzlich erzeugten AD Mobile Peers hohe Verluste auf und anschließend erreicht keine einzige der durch das Smartphone gesendeten Nachrichten mehr den AD Mobile Peer. Dafür ist vermutlich die Überlastung des Eingangspuffers des Smartphones verantwortlich. Dieser ist durch die Nachrichten der zusätzlichen AD Mobile Peers bereits vollständig belegt und es können keine weiteren Nachrichten empfangen werden. Dieses Problem ließe sich lösen, indem man nicht alle auf dem AD Fix Peer gespeicherten Nachrichten an einen neuen AD Mobile Peer übertragen würde, sondern lediglich eine begrenzte Anzahl der aktuellsten.

Beim Verbindungsaufbau durch die simulierten AD Mobile Peers kam es gelegentlich zum Einfrieren der Realisierung des AD Fix Peer auf einem Raspberry Pi. Im Test konnte dies ab 700 Threads beobachtet werden. Abhilfe konnte an dieser Stelle nur ein Neustart des Gerätes schaffen. Ab 1100 Threads geschah dies bei jedem Versuch; entweder gleich beim Verbindungsaufbau oder aber im Testverlauf selbst. Daher konnten ab 1100 Threads keine Daten mehr erhoben werden. Die Gründe hierfür konnten wir noch nicht endgültig identifizieren.

Die Ergebnisse zeigen, dass ein Angreifer einen Denial of Service Angriff ausführen und den AD Fix Peer durch die Erzeugung sehr vieler simulierter AD Mobile Peers unbrauchbar machen könnte. Dem lässt sich entgegenwirken, indem der AD Fix Peer nur Verbindungen von AD Mobile Peers zulässt, die nicht bereits mit ihm verbunden sind. Jedoch kann dies auch im normalen Betrieb bei sehr vielen verbundenen Geräten zu Schwierigkeiten führen, da sich nach der Änderung der MAC-Adresse alle Geräte gleichzeitig erneut mit dem AD Fix Peer verbinden wollen. Um in einer solchen Situation einen Ausfall zu verhindern, könnte man neue Verbindungen nach einer maximal zulässigen Anzahl abweisen. Weiter könnten die AD Mobile Peers nach Ändern der MAC-Adresse zunächst eine zufällige Zeit warten, bevor sie erneut eine Verbindung zum AD Fix Peer aufbauen.

6.2 Praxistest

Bei diesem Test wurde untersucht, wie sich die Anwendung verhält, wenn der Nutzer das Smartphone mit ausgeschaltetem Bildschirm in der Hosentasche mit sich trägt und kurzzeitig in Reichweite eines AD Fix Peers kommt. Dies ist in Abbildung 6 zu sehen. Der Test kommt dem praktischen Einsatz der Anwendung sehr nahe, berücksichtigt jedoch nur einen einzelnen Nutzer. Zusammen mit dem Test in Abschnitt 6.1 sollte dies trotzdem zu einer realitätsnahen Einschätzung der Leistungsfähigkeit der Anwendung führen.

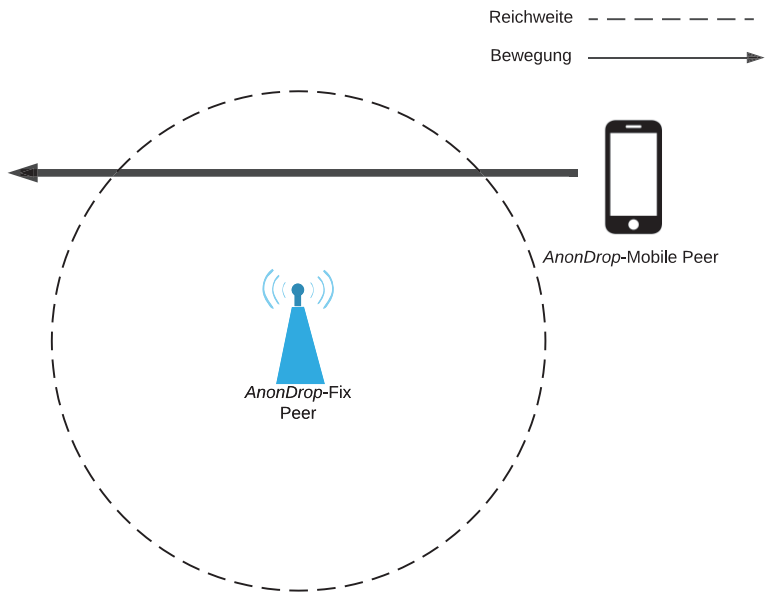

Abb. 6: Bewegung durch das Empfangsgebiet mit verschiedenen Geschwindigkeiten

Die Versuchsperson bewegt sich dabei mit verschiedenen Geschwindigkeiten durch das gekennzeichnete Gebiet. Bei dem Test befand sich der AD Fix Peer im zweiten Stock eines Wohnhauses hinter einem Fenster. Die Reichweite des Signals beträgt circa 50 Meter in beide Richtungen der anliegenden Straße, anschließend bricht die Verbindung ab. Die Messung wurde jeweils drei Mal bei verschiedenen Geschwindigkeiten durchgeführt. Auf dem AD Fix Peer waren dazu 200 Nachrichten gespeichert, die nach der erfolgreichen Herstellung einer Verbindung durch den AD Mobile Peer an diesen übertragen werden sollten. Außerdem wurde auf dem AD Mobile Peer eine Nachricht zum Senden im Hintergrund gespeichert, die nach dem erfolgreichen Schlüsselaustausch an den Fix Peer übertragen werden sollte. Dieser wurde so eingestellt, dass die Nachricht direkt an den Mobile Peer zurück gesendet wurde. Es wurde also nicht auf eine größere Anzahl gesendeter Nachrichten oder Phantomnachrichten gewartet. Die Ergebnisse sind in Tabelle 2 zu sehen. Die Zahl in der jeweiligen Spalte zeigt an, wie viele der zuvor auf dem AD Fix Peer gespeicherten Nachrichten an den AD Mobile Peer übertragen werden konnten und ein + kennzeichnet, ob die Hintergrundnachricht des AD Mobile Peers an den AD Fix Peer übertragen werden konnte.

Die Messungen zeigen deutlich, dass bei höheren Geschwindigkeiten der Nachrichtenaustausch seltener stattfindet oder sogar völlig ausbleibt. Dies hängt mit der mangelnden Zeit des AD Mobile Peers zusammen, das vom AD Fix Peer angebotene WiFi Signal durch einen Scan der verfügbaren WLAN-Netzwerke zu finden und eine Verbindung aufzubauen.

Tab. 2: Messungen des Praxistests

Geschwindigkeit	Messung 1	Messung 2	Messung 3
ca. 4 km/h (zu Fuß)	200+	200+	200+
ca. 6 km/h (zu Fuß)	200+	200+	200+
15 km/h (im Auto)	200+	200+	0
20 km/h (im Auto)	52+	0	0
30 km/h (im Auto)	0	0	0

Es ist festzuhalten, dass bei beiden Messreihen in Schrittgeschwindigkeit die Nachrichten vollständig zwischen AD Mobile Peer und AD Fix Peer übertragen werden konnten. Bei einer Geschwindigkeit von 15 km/h ist dies zumindest bei zwei von drei Versuchen gelungen.

Aktuell nicht betrachtet sind Fragen zur Usability des UI des AD Mobile Peers, bspw. wie man einem Nutzer sinnvoll mehrere hundert Nachrichten präsentiert.

7 Zusammenfassung und Ausblick

Es wurde *AnonDrop* vorgestellt, ein dezentrales System zur anonymen, räumlich begrenzten Verbreitung von Informationen. Kommunikationsabläufe und Schutzmaßnahmen gegen mögliche Angriffe zur Identifizierung von Nachrichtenquellen wurden diskutiert und bewertet. Mithilfe eines Prototyps konnte gezeigt werden, dass es technisch möglich ist, für *single-hop* Kommunikation komplett auf statische Netzwerk IDs zu verzichten, die einen Knoten und somit einen Nutzer identifizieren könnten. Die dezentrale Architektur macht – eine hohe Verbreitung der AD Fix Peers vorausgesetzt – *AnonDrop* sehr robust gegen staatliche An- bzw. Eingriffe zum Zwecke der Zensur. An dieser Stelle sei aber auch darauf hingewiesen, dass *AnonDrop* sich für die Kommunikation illegaler Aktivitäten eignet und eine Aufdeckung durch Strafverfolgungsbehörden erschweren würde.

Als nächstes sollen Sybil-Angriffe sowie Vertrauensmodelle, die das gezielte Verbreiten von Falschinformationen in *AnonDrop* erschweren, untersucht werden.

Literatur

[Cya16] CyanogenMod. http://www.cyanogenmod.org, Stand: 27.05.2016 (2009 – 2016).

[Douc02] J. R. Douceur: The Sybil Attack. *In: Peer-to-peer Systems*, Springer (2002), 251–260.

[Fall03] K. Fall: A Delay-tolerant Network Architecture for Challenged Internets. *In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, ACM (2003), 27–34.

[HeKM08] A. Heinemann, J. Kangasharju, M. Mühlhäuser: Opportunistic Data Dissemination Using Real-World User Mobility Traces. *In: 1st IEEE Intl. Workshop on Opportunistic Networking (WON-08)* (2008), 1715–1720.

[HeSt10] A. Heinemann, T. Straub: Opportunistic Networks as an Enabling Technology for Mobile Word-of-Mouth Advertising. *In: Handbook of Research on Mobile Marketing Management*, IGI Global (2010), 1618–1637.

[HuMK16] C. Huitema, T. Mrugalski, S. Krishnan: Anonymity profile for DHCP clients. https://datatracker.ietf.org/doc/draft-ietf-dhc-anonymity-profile, Stand: 23.03.2016 (2016).

[KaP12] T. Kärkkäinen, M. Pitkänen, J. Ott. Enabling Ad-hoc-style Communication in Public WLAN Hot-spots. *In: Proceedings of the Seventh ACM International Workshop on Challenged Networks*, CHANTS '12, ACM (2012), 31–38.

[Krem11] M. Kremp: Totalabschaltung: Wie Ägypten aus dem Internet verschwand. http://www.spiegel.de/netzwelt/netzpolitik/totalabschaltung-wie-aegypten-aus-dem-internet-verschwand-a-742232.html, Stand: 23.03.2016 (2011).

[LeHV07] M. Lei, X. Hong, S. V. Vrbsky: Protecting location privacy with dynamic MAC address exchanging in wireless networks. *In: Global Telecommunications Conference, 2007. GLOBECOM'07*, IEEE (2007), 377.

[LiHu09] A. Lindgren, P. Hui: The quest for a Killer App for Opportunistic and Delay Tolerant Networks. *In: Proceedings of the 4th ACM workshop on Challenged networks*, ACM (2009), 59–66.

[PiSL06] C. Piro, C. Shields, B. N. Levine: Detecting the Sybil Attack in Mobile Ad Hoc Networks. *In: Securecomm and Workshops, 2006*, IEEE (2006), 1–11.

[Ras16] Raspberry Pi – Teach, Learn, and Make with Raspberry Pi. https://www.raspberrypi.org, Stand: 23.03.2016 (2012 – 2016).

[Soko11] D. A. Sokolov: Internet-Abschaltung: Libyen hat von Ägypten gelernt. http://www.heise.de/netze/meldung/Internet-Abschaltung-Libyen-hat-von-Aegypten-gelernt-1206016.html, Stand: 23.03.2016 (2011).

[Stö12] C. Stöcker: Kämpfe in Damaskus: Syrien vom Internet abgekoppelt. http://www.spiegel.de/netzwelt/netzpolitik/syrien-vom-internet-abgekoppelt-a-870046.html, Stand: 23.03.2016 (2012).

[Stat16] Statista: Prognose zur Anzahl der Smartphone-Nutzer weltweit von 2012 bis 2019. http://de.statista.com/statistik/daten/studie/309656/umfrage/prognose-zur-anzahl-der-smartphone-nutzer-weltweit, Stand: 01.06.2016 (2016).

[Swee02] L. Sweeney: K-Anonymity: a Model for Protecting Privacy. In: *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10, 5 (2002).

[TKHL15] S. Trifunovic, M. Kurant, K. A. Hummel, F. Legendre: WLAN-Opp: Ad-hoc-less opportunistic networking on smartphones. In: *Ad Hoc Networks*, 25, Part B (2015), 346 – 358.

[WeGG14] G. Weston, G. Greenwald, R. Gallagher: CSEC used airport Wi-Fi to track Canadian travellers. http://www.cbc.ca/news/politics/csec-used-airport-wi-fi-to-track-canadian-travellers-edward-snowden-documents-1.2517881, Stand: 23.03.2016 (2014).