Seminar IT-Sicherheit

Freenet

An anonymous peer-to-peer Network

von

Michael Greßmann

(winf100373@fh-wedel.de)

im

Wintersemester 2015/2016

Dozent:

Prof. Dr. Gerd Beuster

(gb@fh-wedel.de)

Eingereicht am 06.12.2015

Inhaltsverzeichnis

1. Einleitung

In unserer heutigen Zeit sind netzwerkfähige Computersysteme kaum mehr wegzudenken. Die immer größer werdende Masse an Informationen lässt sich nur mittels Digitalisierung auffangen. Trotz dieses Mediums und der damit verbundenen Möglichkeit, Daten schnell zu verbreiten, gewähren herkömmliche Systeme keine Garantie, dass diese auch erhalten bleiben. Gerade sensible Informationen sind unter derartigen Umständen ständig der Gefahr ausgesetzt, von Gegnern aus Netzwerken entfernt zu werden. Dabei machen sich die Ersteller, Verbreiter und Fürsprecher von kontroversen Themen angreifbar, falls ein Rückschluss auf ihre Identität erfolgreich war. Die Frage der Sicherheit der Privatsphäre in Netzwerken stellt sich aber in unserer vernetzten Welt jedem Einzelnen. Der Wunsch nach Anonymität von Lesern und Konsumenten von Informationen auf der einen und Erstellern und Verbreitern auf der anderen Seite ist in diesem Zusammenhang gleichermaßen erstrebenswert. Infolge dessen entwickelten sich zahlreiche Lösungsversuche, um diesen Wunsch gerecht zu werden.

Eine interessante Umsetzung stellt in diesem Kontext die Peer-to-Peer Netzwerkanwendung „Freenet“ dar, die es Nutzern ermöglicht, anonymisiert Informationen auszutauschen. Mit dieser Seminararbeit wird Aufschluss über das Projekt „Freenet“, welches von Ian Clark ins Leben gerufen wurde, gegeben. Insbesondere wird die technische Funktionsweise im Hinblick auf die Aspekte der Datenübermittlung und ihrer Sicherheit erläutert sowie der geschichtliche Verlauf des Projekts dargestellt.

2. Theoretische Grundlagen

Die in dieser Arbeit aufbereitete Thematik setzt beim Leser einige Grundkenntnisse im Bereich der Informatik voraus, um sie bis ins Detail nachvollziehen zu können. Dennoch beschäftigt sich dieses Kapitel mit den wichtigsten darauf aufbauenden Inhalten, die insbesondere für das technische Verständnis eine große Rolle spielen.

Freenet ist eine Peer-to-Peer Netzwerkanwendung, die es ermöglicht, Daten im Netzwerk zu veröffentlichen, zu kopieren und zu erhalten. Dabei wird sowohl die Anonymität der Autoren als auch der Leser geschützt. Das Netzwerk besteht aus identischen Knoten, die zusammen den gesamten Speicher darstellen und Dateianfragen so untereinander routen, dass ein Zielort der Datei am wahrscheinlichsten gefunden wird. Dateien existieren nicht nur einmalig, sondern vervielfältigen sich dynamisch in der Nähe des Anfragenden und werden an Orten mit seltenen Anfragen gelöscht. Somit erfolgt das Referenzieren einer Datei ortsunabhängig und es ist schwierig, den Ursprung einer Datei festzustellen, die auf diese Weise durch das Netzwerk weitergeleitet wird. Auch besteht ein geringer Bezug zwischen den tatsächlichen Inhalten eines Knotens und dem, der ihn betreibt, da nicht nur kaum Einfluss auf die Verbreitung von Daten genommen werden kann, sondern auch das Wissen über den Besitz im Verborgenen bleibt. [1]

2.1. Peer-to-Peer Netzwerk

Peer (englisch für „gleichrangig“, „gleichgestellt“) beschreibt im Kontext der Informatik einen Kommunikationsendpunkt in einem Computernetzwerk. Diesem ist es möglich, z.B. mittels einer Software, Dienste anderer Peers zu nutzen und eigene Dienste anderen Peers zur Verfügung zu stellen, sodass ein Peer-to-Peer Netzwerk vorliegt. Dabei können auch Ressourcen (z.B. Speicherplatz, Rechenleistung etc.) und Dateien gleichberechtigt verwendet werden. Hierbei ergibt sich die signifikante Eigenschaft solcher Netzwerke, nämlich die Dezentralisierung, auf die im Kapitel 3 weitergehend eingegangen wird und von bedeutendem Charakter im Freenet-Projekt gekennzeichnet ist.

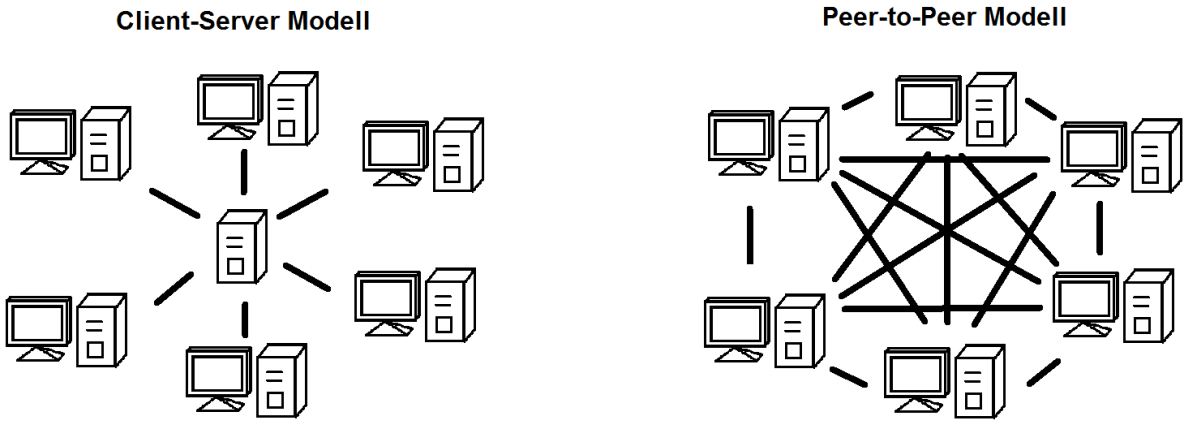

Abbildung 1 „Kommunikationsverbindungen im Client-Server- und P2P Modell“

In einem Client-Server-Modell, welches den Gegensatz zum Peer-to-Peer Netzwerk darstellt, besteht bereits ein Rollenkonzept, das nur Clients (englisch für „Kunde“) erlaubt, Dienste von Servern, die den Aspekt der Zentralisierung realisieren, in Anspruch zu nehmen. Insofern kann ein Peer auch so interpretiert werden, dass er gleichzeitig die Rolle des Clients und des Servers einnimmt. Der architekturbedingte und verbindungsspezifische Unterschied zwischen den beschriebenen Netzwerken wird in Abbildung 1 verdeutlicht. [2]

2.2. Overlay-Netzwerk

Da nicht jeder Peer dafür geeignet ist, um leistungsorientierte Aufgaben, wie die Organisation der Peers untereinander, wahrzunehmen, wird eine Gruppierung vorgenommen, sodass je nach Anforderung bestimmte Aufgaben zugewiesen werden. Dies wird durch das Aufsetzen von sogenannten „Overlay-Netzwerken“ (englisch für „Überlagerung“) auf die bestehende Infrastruktur (sog. Underlay) bzw. andere Overlay-Netzwerke ermöglicht, die damit die bereits vorhandene Struktur um eine zusätzliche (logische oder physikalische) Topologie (Verbindungsstruktur zwischen Computern) erweitert. Auf diese Weise kann auch eine Suchfunktion bereitgestellt werden. Je nach Implementierung können damit Peers und ihre zugeteilten Objekte identifiziert werden, was einem strukturierten Overlay gleichkommt. Im Fall von Freenet widerspricht dieser Ansatz jedoch dem Ziel der Anonymität der Nutzer. Aus diesem Grund kommen unstrukturierte Overlays zum Einsatz, in denen mit umfangreicher Verschlüsselung gearbeitet wird (mehr dazu im Kapitel 3.2). Weiterhin ergibt sich aus den unstrukturierten Overlays eine tiefere Unterteilung nach der Art des Aufbaus. So ist neben dem reinen Peer-to-Peer System, zu dem auch Freenet zählt, eine hybride

Umsetzung aus P2P- und Client-Server Modell möglich, bei der Server zur Kommunikationskoordination eingesetzt werden, Peers aber weiterhin untereinander interagieren.

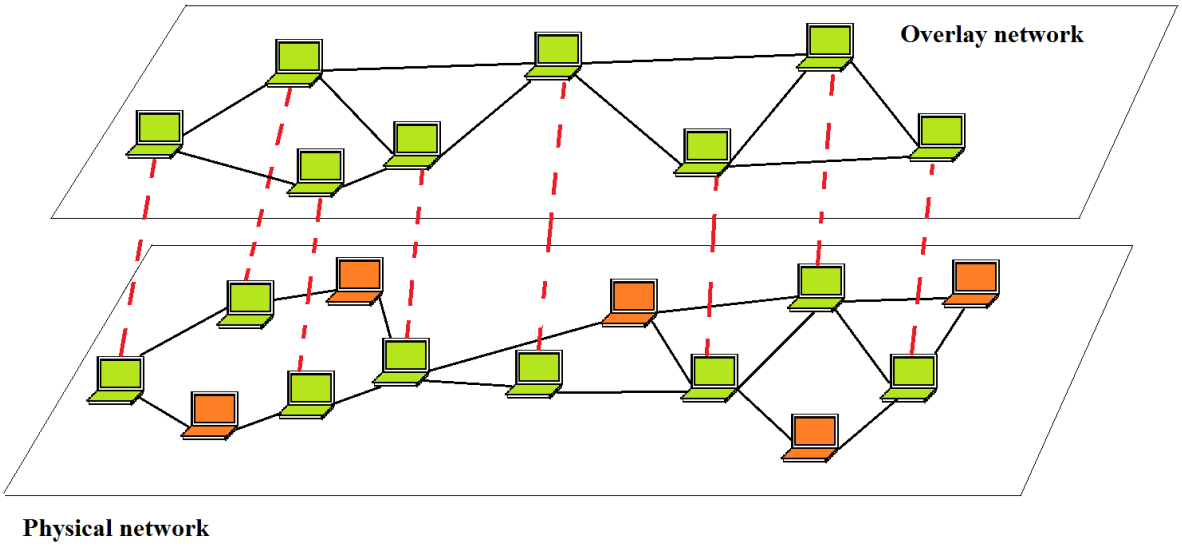

Abbildung 2 „Overlay-Netzwerk auf einem Underlay-Netzwerk“

Eine starke Vereinfachung einer möglichen Overlay-Underlay-Beziehung zeigt Abbildung 2. Wie bereits erwähnt, kann eine solche Struktur um weitere Schichten ergänzt werden. Sie findet auch in anderen Bereichen als der Informatik, wie z.B. bei Strom- oder Telekommunikationsnetzen, Anwendung. [3]

3. Technische Funktionsweise

3.1. Architektur

Der Kernaspekt bei der Implementierung von Freenet liegt in der Grundeigenschaft eines anpassungsfähigen Peer-to-Peer Netzwerks, die es Knoten erlaubt, sich untereinander Daten zuzusenden und zu speichern. Dabei verwaltet jeder Knoten seinen eigenen lokalen Speicher und stellt sie anderen im Netzwerk befindlichen Teilnehmern zur Verfügung. Außerdem wird sowohl eine dynamische Adressliste von anderen bekannten Knoten hinterlegt als auch ein Schlüsselverzeichnis, mit dem die Dateien unabhängig vom physischen Aufenthaltsort identifiziert werden können. Somit kann zum einen die Sicherheit anderer gewährleistet und zum anderen die Speicherkapazität des Netzwerks erhöht werden. Sicherheit aus dem Grund, dass beispielsweise die Verwendung von bedrohlichen und unbekannten Knoten auf diese Weise eingeschränkt werden kann.

Man kann dieses System als kooperatives und dezentrales Dateisystem betrachten, das Vervielfältigung und damit Unabhängigkeit bezüglich der Speicherorte vereint. Die Architektur basiert darauf, dass Anfragen für Schlüssel von Knoten zu Knoten weitergereicht werden und jeweils stellvertretend eine Anfrage an den nächsten Knoten geschickt wird, der wiederum selbstständig dessen Nachfolger bestimmt. Diese Vorgehensweise gleich dem IP Routing, nur dass Freenet im Laufe der Zeit schnellere Routen bestimmt, je mehr Wissen über die Knoten vorhanden ist und sich an das Netz anpasst, beispielsweise wenn sich neue Knoten dem Netzwerk bekannt machen. Die Privatsphäre bleibt unterdessen geschützt, da ein Knoten nur Kenntnis über seine direkten Nachbarknoten besitzt.

Eine weitere Parallele zum IP Routing lässt sich beim Hops-to-live (HTL) Limit finden, der analog zum Time-to-live Feld bei jedem Knoten dekrementiert wird, um Endlosschleifen zu vermeiden. Da innerhalb des gesetzten Limits gleiche Knoten nicht mehrfach durchlaufen werden sollen, können Anfragen anhand einer zufallsgenerierten Kennung identifiziert und abgelehnt werden. In solchen Fällen nimmt die Anfrage einen anderen Knoten. Dieser Prozess setzt sich fort bis eine Anfrage erfolgreich war oder aber das Hops-to-live Limit überschritten wurde. Danach wird eine entsprechende Information entlang der Kette gesendet. [1] [4]

3.2. Schlüssel und Suchverfahren

Freenets gesamte Inhalte werden durch Schlüssel (Hashwerte) identifiziert. Das Netzwerk verwendet drei Schlüsseltypen, die auf ihren Anwendungsbereich spezialisiert sind und sich durch jeweilige Konstruktion unterscheiden.

Mit dem signed-subspace key (SSK) werden persönliche Namensräume ermöglicht. Die dabei verwendete asymmetrische Verschlüsselung erlaubt es nur dem Besitzer bzw. Personen im Besitz des privaten Schlüssels, Veränderungen vorzunehmen und gleichzeitig allen anderen, Inhalte zu lesen. Als erstes wird ein zufälliges Schlüsselpaar (privat und öffentlich) erzeugt, das der Identifikation des Namensraums dient. Aus dem öffentlichen Schlüssel des Namensraums und einer Kurzbeschreibung entstehen dann jeweils unabhängig voneinander Hashwerte, die mittels der „XOR“ Operation zu einem Wert verkettet werden. Das Ergebnis wird erneut mit der Hashfunktion verarbeitet. Zum Schluss wird die Datei mit dem privaten Schlüssel unterzeichnet und macht sie damit von anderen verifizierbar. Man schafft damit trotz der Anonymität ein gewisses Maß an Vertrauen, weil zumindest sichergestellt ist, dass es sich um dieselbe Person handelt, die Inhalte im Namensraum bereitstellt. Um nun Zugriff auf eine Datei im Unternamensraum zu erhalten, benötigt man den dazugehörigen öffentlichen Schlüssel und dessen Kurzbeschreibung, sodass der SSK nachgebildet werden kann.

Der Namensraum kann vom Nutzer so aufgebaut werden, dass eine hierarchische Struktur vorherrscht, indem nur auf die entsprechenden Daten referenziert wird, anstatt sie dort direkt zu hinterlegen. Auf diese Weise können Daten einem Update unterzogen werden, ohne dass die Referenz beeinträchtigt wird, da der Besitzer den ursprünglichen SSK auf die neue Version aktualisieren kann.

Der zweite verwendete Schlüssel, der content-hash key (CHK), bezieht sich auf die Verarbeitung von Daten selbst. Hierbei wird lediglich der Inhalt durch SHA-256 (Secure Hash Algorithm 2) auf einen Hashwert abgeleitet. So wird eine gewisse Einzigartigkeit von Dateien erreicht, weil jeder Knoten den gleichen Wert berechnen würde. Identische Kopien werden daher automatisch erkannt und vereinigt.

Der Zugriff auf eine Datei erfordert den entsprechenden CHK und den Entschlüsselungscode (muss vom Besitzer veröffentlicht werden), da Daten lokal verschlüsselt, aber nie mit diesem Code zusammen gespeichert werden, aus Gründen, die in Kapitel 3.4 erläutert sind. Hier lässt sich das Zusammenspiel der beiden Schlüssel erkennen, denn so können Daten auf verschiedene Weisen veröffentlicht werden.

Der KSK (keyword-signed key), der ähnlich zum SSK gebildet wird, aber weniger sicher ist, weil die Kurzbeschreibung allein gehasht wird und damit die Schlüsselgenerierung hervorruft, ermöglicht das Suchen von Daten in Freenet. Folglich können Nutzer unabhängig voneinander die gleiche Kurzbeschreibung für unterschiedliche Inhalte verwenden. Mit Zusammenstellungen von häufig aufgerufenen oder beliebten Inhalten könnte dem entgegengewirkt werden. Die effiziente Verbreitung, sodass Hashwerte für Namensräume, Inhalte oder Kurzbeschreibungen einfach und schnell auffindbar sind, stellt dennoch ein Problem dar. [1] [4]

3.3. Datenaustausch und Routing

3.3.1. Datei erhalten

Um eine Datei zu erhalten, muss der Teilnehmer zuerst den zugehörigen binären Dateischlüssel (CHK, SSK, KSK) erhalten oder berechnen. Danach wird eine Anfrage an den eigenen Knoten mit diesem Schlüssel gestellt und das HTL Limit gesetzt. Wenn nun ein Knoten eine Anfrage erhält, wird der zugehörige Speicher durchsucht und bei Erfolg zurückgegeben mit einem Hinweis, dass dies die Quelle war. Falls nicht, leitet der Knoten die Anfrage weiter, in dem der lexikografisch ähnlichste Schlüssel in der Routingtabelle aufgesucht wird. Wenn die Anfrage erfolgreich ist, wird die Datei entlang der Kette zurückgegeben, im lokalen Cache hinterlegt und ein entsprechender Eintrag in der Routingtabelle angelegt. Falls ein anzusteuernder Knoten nicht verfügbar ist oder eine Schleife erzeugen würde, wird immer der lexikografisch nächst dichtere Nachbar kontaktiert, bis Erfolg oder Fehler (HTL überschritten oder kein Knoten mehr übrig) eintritt.

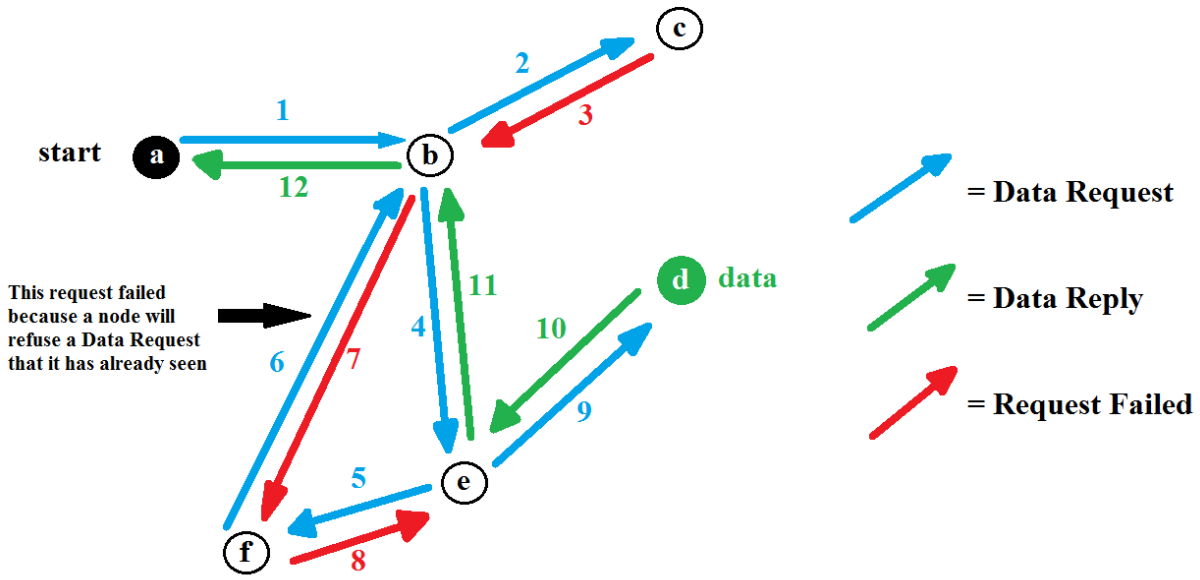

Abbildung 3 „Freenet Routing“

In Abbildung 3 wird ein typischer Ablauf einer Anfrage dargestellt. Sie wird von Knoten „a“ an „b“ geleitet. Dessen Nachfolger „c“ hat keine anderen Knoten zur Verfügung und leitet die Anfrage zurück an „b“. Dieser kontaktiert „e“ als nächst „besseren“ Nachbar. Über „f“ und „b“ wird eine Schleife erzeugt, sodass zurückgeroutet wird, bis ein alternativer Peer gefunden wird, der noch nicht angesteuert wurde, in diesem Fall „d“, der die gesuchte Datei auf dem Weg an den Anfragenden zurückgibt, der keine Fehler hervorgerufen hat: über „e“ nach „b“ zu „a“. Auf dieser Route wird die Datei auch in den Cache der jeweiligen Peers gespeichert. Dieser Algorithmus ermöglicht dem Netzwerk einen dynamisch fortlaufenden Lernprozess, der zwei Effekte generiert: Knoten erweitern ihre Routingtabelle stetig und dessen Einträge sind sich sehr ähnlich, da Anfragen anhand von Schlüsselwertvergleichen weitergereicht werden und so eine Spezialisierung entsteht. [1] [4]

3.3.2. Datei veröffentlichen

Die Funktionsweise zur Veröffentlichung einer Datei im Freenet Netzwerk gestaltet sich analog zu der in 3.3.1 beschrieben Art und Weise und nutzt die Wiederverwendbarkeit des Algorithmus´. Nachdem der Schlüssel für eine zu veröffentlichende Datei erzeugt und ein „insert“ angefordert wurde, agiert jeder Knoten wie bei einer Anfrage und prüft auf Schlüsselkollision, bevor der Schlüssel wie auf zuvor beschriebene Weise weitergereicht wird. Auftretende Fehler behandelt der Algorithmus wie in 3.3.1 erläutert. Das Überschreiten des HTL entspricht in diesem Fall keinem Fehler, denn nun kann die Datei entlang der Route gespeichert und entsprechende Routingeinträge angelegt werden. Die gleichen Effekte wie bei einer Anfrage treten ein und zudem kann ein „insert“ dazu genutzt werden, um den eigenen Knoten im Netz bekannt zu machen. [1] [4]

3.4. Datenmanagement

Freenet erlaubt es Benutzern, die für das Netzwerk zur Verfügung gestellte Speicherkapazität zu wählen (von 128 Megabyte bis zu mehreren Gigabyte). Daten werden nach dem LRU-Prinzip (Least Recently Used) verwaltet, bei dem sie absteigend nach dem Zeitpunkt der jüngsten Anfrage (oder Veröffentlichung) sortiert werden. Sobald eine neue Datei eintrifft, unabhängig davon, ob durch einen „request“ oder einen „insert“, und die Speicherkapazität erreicht wurde, verlassen so viele am Ende der Liste stehenden Daten den Speicher, bis wieder genügend Platz für die neue Datei vorhanden ist. Die Einträge der Routingtabelle hingegen können, obwohl hier auch LRU zu Anwendung kommt, länger bestehen bleiben, da sie viel weniger Speicher verbrauchen, sodass für

einen Knoten die Möglichkeit besteht, eine erneute Kopie der Datei erhalten zu können. Die zuvor erwähnte cache-ähnliche Verbreitungsfunktion des Netzwerks ist somit durch die Häufigkeit der Anfragen eingeschränkt und es besteht keine Kopie mehr, nachdem alle Knoten die entsprechende Datei entfernt haben.

Aus Rechtsgründen wird jede Datei im lokalen Speicher von Knoten verschlüsselt. Somit, so die Idee der Entwickler, können Knotenbetreiber den Inhalt des eigenen Speichers nicht kennen und es damit begründen, dass nur der Dateischlüssel bekannt ist, nicht aber der Verschlüsselungscode. Dieser wird nämlich bei SSK's und KSK's erst durch das Umkehren des Hashwertes zugänglich. Für CHK's gilt dies nicht aus bereits genannten Gründen in Kapitel 3.1. [1] [4]

3.5. Knoten hinzufügen

Ein neuer Knoten kann dem Netzwerk beitreten, indem eine Ankündigungsnachricht mit der eigenen Adresse und einem Hashwert eines zufälligen „Seeds" an einen bereits existierenden Teilnehmer geschickt wird. Nach Erhalt der Nachricht verkettet der Knoten diesen Wert mit seinem zufälligen „Seed" mittels der „XOR" Funktion und hasht das Ergebnis erneut. Dieser Vorgang wird an weiteren zufällig aus der Routingtabelle ausgewählten Nachbarn wiederholt, bis das HTL abgelaufen ist. Danach bildet sich der Schlüssel des neuen Knotens aus der „XOR" Verknüpfung der „Seeds" der anderen Peers. Dabei kann jeder Knoten die Hashergebnisse aller anderen überprüfen. Somit ist sichergestellt, dass der Schlüssel des neuen Knotens nicht manipuliert wurde und absolut zufällig ist. Zum Schluss hinterlegen die beteiligten Peers den entsprechenden Eintrag des neuen Teilnehmers. [1] [4]

4. Sicherheit

Das oberste Ziel, das sich Freenet gesetzt hat, ist der Schutz der Anonymität von Benutzern, sowohl Autoren als auch Konsumenten von Daten. Den Entwicklern von Freenet ist es dabei auch wichtig, dass die Identität von Knotenbetreibern geschützt ist, schließlich verbreiten sich Dateien durch Suchanfragen gegebenenfalls über den eigenen Knoten (die dargelegten Gründe finden sich in 3.4). Außerdem basiert die Kommunikation zwischen Knoten darauf, dass nur die direkten Nachbarn voneinander Kenntnis haben und sie gleichzeitig trotzdem nicht unterscheiden können, wer ursprünglich eine Datei nachgefragt oder veröffentlicht hat, da jeder in der Kette befindliche Teilnehmer dies selbst vortäuscht.

4.1. Angriffsszenarien

Trotz der Schutzmaßnahmen sind einige Szenarien denkbar, mit denen Angreifern die Identität (IP Adresse) von Knoten und dessen Inhalte offenlegen können, die im Folgenden skizziert werden. Im typischen Fall eines Whistleblowers, der geheime Informationen veröffentlichen möchte, muss ein Angreifer in der Lage sein, die hochzuladenden Daten im Voraus zu kennen (siehe Aspekte der Schlüssel in Freenet Kapitel 3.2). Zudem muss ein Weg gefunden worden sein, um sich schnell im Netzwerk bewegen zu können, z.B. über eine Vielzahl von manipulierten Knoten. Sind diese Voraussetzungen gegeben, muss der Angriff während des „inserts“ von Statten gehen, da nach diesem Vorgang, wie bereits erwähnt, keine eindeutige Zuordnung von Knoten und Datei aufgrund der dynamischen Verbreitung von Inhalten mehr möglich ist. Dabei besteht auch die Möglichkeit, dass der Autor das Netzwerk bereits wieder verlassen hat, bevor der Angriff erfolgreich abgeschlossen wurde.

Harvesting (engl. für „Abernten“) beschreibt das Sammeln von Benutzerinformationen in einem Netzwerk, häufig im Bereich von Mailinglisten. Analog dazu wäre das „Ernten“ auch auf Knoten in Freenet umsetzbar. Mittels performanten und manipulierten Peers, die einen so großen Bereich in Freenet abdecken, sodass komplette Ketten von „requests“ und „inserts“ abgefangen werden, sind die Identitäten relativ einfach herauszufinden.

Bootstrapping (engl. für „Ureingabe“) Angriffe stellen die Gefahr dar, dass sich neue Knoten beim ersten Verbinden mit Freenet direkt von manipulierten „Seeds“ abgefangen werden oder aber die von Freenet selbst zur Verfügung gestellten „Seeds“ könnten direkt von einer Firewall blockiert werden. Wie bereits eingangs erwähnt, ist eine „hit and run“ Strategie durchaus sicher, aber dennoch durch den soeben beschriebenen Angriffsweg mit einem gewissen Risiko verbunden.

Adaptive Search (engl. für „anpassungsfähige Suche“) umfasst einen Angriff, der den Autor für bestimmte Dateien ausfindig machen lässt. Dabei ist es notwendig, die Schlüssel für die Datei vorherzusagen und an neuen Knoten einen Abgleich für zu veröffentlichende Daten zu machen. Daraus können Schlussfolgerungen über den Vorgänger gemacht werden, was aber eher mit „Raten“ zu tun hat. Wenn die Folgerungen zu Knoten dicht bei dem Veröffentlicher leiten, kann es zum Erfolg führen, da in der Nähe immer mehr Schlüssel bekannt werden, die dem gesuchten ähnlich sind. Es gibt noch weitere mögliche Angriffe wie Correlation attacks, Traffic analysis und Swapping attacks, auf die an dieser Stelle verwiesen sei. [1] [4] [5] [6]

4.2. Darknet

All die zuvor erwähnten möglichen Angriffe lassen Freenet nicht besonders sicher erscheinen. Jedoch beziehen sie sich im Großteil auf das sogenannte „Opennet“, dem Netzwerk, bei dem Verbindungen zu unbekannten Knoten aufgebaut werden und das bis zu diesem Punkt erläutert wurde. Um den meisten der Sicherheitslücken entgegenzuwirken entstand das „Darknet“ (engl. für „dunkles Netz“) mit der Version 0.7 von Freenet im Jahr 2008. Dabei handelt es sich um ein abgegrenztes Netzwerk, in das sich nur bestimmte Nutzer, die ein gewisses Vertrauen mitbringen, verbinden können. In der Praxis sind dies Personen, die sich auch privat kennen. Es gibt also keinen zentralen Knoten, dem jeder Benutzer seine Identität preisgeben muss, sondern nur denen, die sie auch selbst im Gegenzug offenbaren. Es bleibt den Teilnehmer selbst überlassen, ob sie sich mit dem „firend-to-friend network“ verbinden und für Unbekannte im Verborgenen bleiben oder das „Opennet“ bevorzugen, um, trotz mangelnder Sicherheit, die Vorzüge eines größeren Wissensspeichers zu nutzen. Aus der Tatsache der Abgrenzung von Knoten heraus muss es somit auch gleichzeitig mehrere „Darknets“ geben, in denen kommuniziert wird.

Dieser Modus erforderte auch eine Veränderung des Routingalgorithmus innerhalb des „Darknet“, da ja nun nicht mehr potentiell jeder beliebige Teilnehmer im Netzwerk nach Daten gefragt und so die Spezialisierung der unbekannten Knoten genutzt werden kann. Die Entwickler implementierten daher eine Lösung, die jedem Teilnehmer (fortwährend immer innerhalb des „Darknet“) eine Verortung mittels eines Zahlenwertes zwischen 0 und 1 vergibt. Anhand dessen werden Hashwerte angefragter Schlüssel, bei fehlender Übereinstimmung im lokalen Speicher, nur minimal verändert, sodass sie im gleichen Bereich (Zahlenwert-Ort-Zuweisung) an einen dem Schlüssel ähnlichen Knoten in der Nähe weitergeroutet werden. Dies wird mit den in 3.3.1 und 3.3.2 beschriebenen Aspekten fortgesetzt und verhält sich analog zu dem im „Opennet“ vorherrschenden Ablauf für „requests“ und „inserts“. Anders als dort gibt kein Knoten eine Information über die Quelle,

sondern speichert nur eine Kopie im Cache. Möchte man den Speicherort ausfindig machen, resultiert dies nur in eine noch weitere Verbreitung im Cache anderer Knoten.

Das Netzwerk (aller „Darknets“) geht dabei davon aus, dass die Teilnetze dem Kleine-Welt-Phänomen unterworfen sind. Diese aus der Sozialpsychologie stammende Theorie besagt, dass jeder Mensch mit jedem anderen über eine relativ kurze Kette von Bekanntschaften verbunden ist. Insbesondere bedeutet dies, dass die Überzahl der sozialen Akteure viele kurze (auf die Distanz bezogen) Beziehungen pflegt und nur wenige, die über lange Strecken bestehen und z.B. von interkontinentaler Natur geprägt sind. Somit lässt sich ebenfalls eine hohe Skalierbarkeit wie im „Opennet“ erreichen, die potentiell ein globales „Darknet“ ermöglicht. [4] [5] [6]

5. Geschichte

Im Juli 1999 erstellte Ian Clarke das Konzept zu Freenet in seiner Bachelorarbeit an der University of Edinburgh. Der Grundgedanke zur Entwicklung des Freenets ist Daten verteilt zu speichern, eine Zensur zu unterbinden und somit einen anonymen, freien Austausch von Informationen zu ermöglichen. Durch Dezentralisierung, Redundanz, Verschlüsselung und dynamisches Routing soll dieser Grundgedanke realisiert werden. Geprägt wurde dieser Ansatz durch folgendes Zitat:

"I worry about my child and the Internet all the time, even though she's too young to have logged on yet. Here's what I worry about. I worry that 10 or 15 years from now, she will come to me and say 'Daddy, where were you when they took freedom of the press away from the Internet?'"

–Mike Godwin, Electronic Frontier Foundation, "Fear of Freedom" (1995)

Bereits kurz nach Veröffentlichung der Bachelorarbeit fanden sich einige Freiwillige an dem Programm zu arbeiten. Version 0.1 machten die Entwickler im März 2000 der Öffentlichkeit zugänglich und daraufhin wurde über Freenet in den Nachrichten häufig berichtet. Allerdings beschäftigte sich die Presse dabei hauptsächlich auf die Auswirkungen auf Urheberrechte, anstatt über den Grundgedanken der freien Kommunikation. Nicht nur die Presse beschäftigte sich mit Freenet, sondern auch die akademische Welt und so wurde laut „CiteSeer" die Abhandlung von Clarke das meistzitierte wissenschaftliche Dokument der Informatik im Jahr 2000.

Seitdem wird Freenet unter Zuhilfenahme des Internets weiterentwickelt und es entstand die gemeinnützige „The Freenet Project Inc.". Diese beschäftigt einen Vollzeitprogrammierer, welcher durch Spenden und verkauften Produkten bezahlt wird. Zusätzlich helfen Freiwillige mit, welche durch den Grundgedanken oder der technischen Herausforderung motiviert werden.

Im Jahr 2005 wurde für die neue Version Freenet 0.7 eine neue Richtung eingeschlagen und die P2P Anwendung neu zu entwickeln. Die Entwickler überlegten erstmals Freenet als „Darknet" zu gestalten, um so Teilnehmer nur per Einladung Zugriff zum abgegrenzten Freenet zu gestatten (siehe dazu Kapitel 4.2). Die erste Alpha-Version von Freenet 0.7 wurde im April 2006 veröffentlicht. Diese Grunderneuerung Freenets präsentierten Ian Clarke und der schwedische Mathematiker Oskar Sandberg auf der 13. DEF CON Veranstaltung (2005) sowie auf der im gleichen Jahr stattgefundenen 22. Chaos Communication Congress.

In den darauffolgenden Jahren wurde es um das Freenet Projekt „ruhiger" in Bezug auf Neuigkeiten seitens der Entwickler, denn die fortwährenden Sicherheitsprobleme bereiteten dem Entwicklerteam zunehmend Probleme. Besonders in Anbetracht der vergleichsweise wenigen Programmierer, die ihre

Tätigkeit neben dem Hauptberuf oder als Hobby betreiben, leidet das Projekt hauptsächlich durch die geringen verfügbaren Ressourcen und entwickelt sich bis zum heutigen Stand nur in kleinen Schritten (2005 Version 0.7, 2015 Version 0.7.5). Dennoch erhalten diese Menschen das Projekt am Leben. So sind stetige Einträge im Newsbereich der offiziellen Homepage des Freenet Projekts zu finden, die über Updates zur Software, Artikel über Freenet oder ähnliches gefunden werden. Zudem hat das Netzwerk durch die möglich gemachte absolute Meinungsfreiheit auch mit illegalen Inhalten an Reputation eingebüßt, die unweigerlich in Kauf genommen werden müssen, will man doch die Zensurfreiheit allen, auch Kriminellen, gestatten.

Durch die Enthüllungsberichte Edward Snowdens gelang es dem Netzwerk wieder verstärkt Aufmerksamkeit zu erreichen, denn im Rahmen des 7. SuMa Awards wurde unter der Thematik „Schutz gegen Totalüberwachung“ das Freenet Projekt unter insgesamt 50 Vorschlägen als Gewinner ausgelobt. Dabei zeichnete sich die Software durch ihre bereits beschriebenen Kernaspekte aus und eignet sich nach Meinung der Jury, um zukünftige Meinungsfreiheit im Netz zu gewährleisten. [1] [4] [5] [6] [7] [8]

6. Fazit und Ausblick

Die P2P Netzwerkanwendung Freenet setzte bereits Anfang des 21. Jahrhunderts erste Maßstäbe für ein dezentrales und anonymes Netzwerk, das sich insbesondere im Bereich der Verschlüsselung und Redundanz trotz der P2P Architektur auszeichnet. Der Ansatz, dem Netzwerk den „Darknet“ Modus hinzuzufügen, der es nur einem vertrauten Teilnehmerkreis erlaubt, untereinander zu kommunizieren, zeigt bereits seit der Einführung ihre Vorteile. Trotz der wenigen Entwickler schafft es das Team, die Anwendung stetig zu verbessern und besonders die jüngst zu verzeichnenden Erfolge lassen auf weitere Versionen und Verbreitung schließen.

7. Abbildungsverzeichnis

8. Literaturverzeichnis

[1] Ian Clarke, Oskar Sandberg, Brandon Wiley, Theodore W. Hong (2000) „*Freenet: A Distributed Anonymous Information Storage and Retrieval System*“ http://homepage.cs.uiowa.edu/~ghosh/freenet.pdf

[2] Nick Gehrke (2004), Deutscher Universitätsverlag, *„Peer-to-Peer Applikationen für elektronische Märkte“* S. 197-199

[3] James F. Kurose,Keith W. Ross (2008), Pearson Studium, *„Computernetzwerke: Der Top-Down-Ansatz“* S. 184-186

[4] Ian Clarke, Oskar Sandberg, Matthew Toseland, Vilhelm Verendel (2010) „*Private Communication Through a Network of Trusted Connections: The Dark Freenet*“ https://freenetproject.org/assets/papers/freenet-0.7.5-paper.pdf

[5] Stefanie Roos, Benjamin Schiller, Stefan Hacker, Thorsten Strufe (2014) „*Measuring Freenet in the Wild: Censorship-resilience under Observation*“ https://freenetproject.org/assets/papers/roos-pets2014.pdf

[6] The Freenet Project, https://freenetproject.org/help.html, Abgerufen am 27.10.2015

[7] The Freenet Project, https://freenetproject.org/news.html#news, Abgerufen am 29.10.2015

[8] SUMA Awards http://suma-awards.de/preistraeger.html, Abgerufen am 19.11.2015