The Tor Project

Schwachstellen und Risiken

Vorwissenschaftliche Arbeit

Vorgelegt bei Mag. Harald Pietersteiner

Von Pascal Vallaster, 8b

Innsbruck, 23.2.2024

Abstract

The aim of this scientific study is to explain how the Tor anonymization network ensures its security. To this end, the following research questions are posed: How does Tor work in general? How does Tor operate with the traffic that is passed through? Which encryption algorithms does Tor use and when? How can Tor be exploited through vulnerabilities to de-anonymize users? How secure is Tor overall? To provide answers to these questions, the methodological approach of the literature review was chosen.

Tor secures its communication to its maximum using modern strong encryption and a network of servers, so called Nodes, to protect the user. Nevertheless, it still has vulnerabilities which actively are exploited by intelligence agencies but can mostly be avoided by following certain security-rules. Blocking JavaScript has turned out to be the most effective measurement under these rules.

In the evolving digital world, Tor emerges as a beacon of online anonymity. Understanding its restrictions and security-rules whilst mastering a perfect application enables users to navigate the cyberspace safely.

Inhaltsverzeichnis

Abkürzungsverzeichnis

AES ..... Advanced Encryption Standard
DES ..... Data Encryption Standard
DH ..... Diffie Hellman
g^x, g^y ..... Diffie Hellman Parameter
HTTP ..... Hypertext Transfer Protocol
HTTPS ..... Hypertext Transfer Protocol Secure
K, K_1, K_2, K_3 ..... Symmetrischer Schlüssel
N, N_1, N_2, N_3 ..... Nodes
NSA ..... National Security Agency
p, g ..... Primzahlen
Pr ..... Private-Key
Pu ..... Public-Key
RSA ..... Rivest–Shamir–Adleman
Tor ..... The Onion Routing (Protokoll)
x, y ..... Zufällige Zahlen
Z, Y ..... Zufällige Zahlen

1 Einleitung

> "Tor Stinks - We will never be able to de-anonymize all Tor users all the time."[1]

Das soeben erwähnte Zitat stammt aus einem von Edward Snowden 2013 geleakten Dokument der NSA, in dem Methoden zum Umgehen oder Hacken von Tor dargelegt werden.

Die Entscheidung für dieses Thema gründet in meinem starken Interessensgebiet für Cyber-Security, insbesondere hinsichtlich Anonymität im Internet und kryptographischer Prozesse. Tor bündelt diese zwei Interessensgebiete, was die Recherche und die Arbeit für mich noch einmal spannender und interessanter macht.

Jedermann, der speziell im digitalen Raum auf Sicherheit achtet, sollte seine Tools für den Zugang zu und die Bewegung in diesem Raum immer auf deren Funktionsweise hin hinterfragen. Als Internetnutzer, der sehr auf seine Sicherheit im digitalen Raum bedacht ist, eröffnete sich mir im Zusammenhang mit der Verfassung der VWA die besondere Möglichkeit, diesbezüglich eingehende Kenntnisse zu gewinnen und gleichzeitig zu verstehen, warum vor allem Journalisten, Whistleblower, Spione und Geheimdienste Tor verwenden, wenn es ihnen darum geht, ihre Identität im Internet zu verschleiern.

Im Fokus der gegenständlichen Abhandlung steht die Frage, wie sicher Tor insgesamt ist. Um diese Frage beantworten zu können, müssen folgende drei Detailfragen geklärt werden:

1. die Frage, wie der Aufbau einer Verbindung zu einem Server genau bewerkstelligt wird (die Beantwortung dieser Frage ist für das Verständnis der Arbeitsweise von Tor von größter Bedeutung),
2. die Frage, wie der Tor-Traffic verschlüsselt und organisiert wird, und
3. die Frage nach möglichen Schwachstellen und Angriffsflächen von Tor.

Bedauerlicherweise ist die Klärung dieser Fragen alles andere als einfach, weil die Quellenlage zum Forschungsbereich „Tor" schlecht ist. Die meisten Forschungen werden von Geheimdiensten durchgeführt. Diese machen ihre Erkenntnisse jedoch

[1] The Guardian 2013, S.2

nicht öffentlich. Die Forschungen der Geheimdienste wiederum stützen sich vornehmlich auf die 2013 lancierten Leaks von Edward Snowden und damit auf eine ausgesprochen überschaubare und schon etwas „angegraute“ Erkenntnisgrundlage.
Vor diesem Hintergrund wurde bei der Quellensammlung für die VWA so vorgegangen, dass zunächst in den geleakten Dokumenten der NSA nach einschlägigen Informationen gesucht wurde. Erwiesen sich diese Informationen als unzulänglich, wurde eine spezielle Technik bei der Google-Suche, **Google-Dorking** genannt, benutzt, um die erforderlichen Quellen ausfindig zu machen. Falls sich auch diese Methode als unzureichend erwies, blieb als letzte Möglichkeit nur die strikte Analyse jedes resultierenden Eintrags einer „normalen“ Suche.

Die gegenständliche Arbeit ist in vier Hauptteile gegliedert. Zunächst wird erläutert, was das Internet ist und wie es unterteilt ist. Dann wird die Frage geklärt, wie Tor funktioniert und wie es im Internet etabliert ist. Anschließend wird näher auf die Verschlüsselung des Traffics in Tor eingegangen. Der letzte Teil beschäftigt sich mit den Schwachstellen von Tor und den Möglichkeiten zu deren Behebung.

2 Darstellung der methodischen Vorgangsweise

In diesem Kapitel werden die methodischen Vorgangsweisen in Bezug auf das Programmieren und Aufbauen des Scripts in Kap. 6.4 beschrieben.
Die Programmiersprache Python wurde für dieses Script aufgrund seiner hohen Lesbarkeit und seiner Fähigkeit, komplexe mathematische Operationen effizient umzusetzen, gewählt.
Anforderung war es, ein Programm zu entwickeln, welches die praktische Anwendung der Diffie-Hellman-Kryptographie zeigen soll.
Um den mathematischen Prozess nicht vollständig reimplementieren zu müssen, wurde die „cryptography“-Bibliothek verwendet – ein Projekt, das von dritten bereitgestellt, gewartet und veröffentlicht wird und viele vordefinierte Funktionen und Strukturen beinhaltet.
Der Aufbau des Scripts folgt den in der Dokumentation zu findenden Vorlagen und Kurzbeispielen. Wichtig zu erwähnen ist, dass die in den Mustern gezeigte Implementierung nicht für eine Live-Umgebung entwickelt wurde, d.h. nicht betriebstauglich ist.

3 Internet - Surface Web, Deep Web, Dark Web

Der Begriff „Internet“ ist in unserem Sprachgebrauch weit verbreitet. Allerdings ist bei der Verwendung dieses Begriffs meist nur ein kleiner Teil des Internets gemeint, das sogenannte Surface-Web. Darüber hinaus werden in Bezug auf das Internet häufig Begriffe und Definitionen wie z.B. Deep und Dark Web vermischt oder falsch verwendet.
Aus diesem Grund wird in den folgenden Kapiteln erklärt, wie das Internet strukturiert ist. Das Augenmerk liegt hierbei auf der Definition bzw. der Abgrenzung des Deep Webs und des Dark Webs.

3.1 Surface Web

Bereiche des Internets, die mithilfe von Browsern wie Google Chrome oder Firefox erreicht werden können, nennt man Surface Web, Clearnet, Oberflächenweb oder Visible Web. Die darin enthaltenen Webseiten bzw. Webanwendungen werden von **Crawlern** systematisch katalogisiert und indexiert, also in ein Register (Index) aufgenommen, auf das dann die Suchmaschine zugreifen kann.[2] Befindet sich eine Adresse nicht in diesem Index, wird diese auch nicht gefunden.

3.2 Deep Web

Mit dem „Deep Web“ ist der Teil des Internets gemeint, der den Crawlern unzugänglich ist. Die dortigen Adressen werden weder indexiert noch wird der Inhalt dieser Adressen analysiert. Kein Zugang zum „Deep Web“ besteht beispielsweise, wenn ein Password für den Zugriff gebraucht wird, der Inhalt mit einem **CAPTCHA** geschützt wird oder keinerlei Verknüpfungen durch Links im Surface Web existieren.[3] Konkrete Beispiele sind Warenkörbe von Usern, Cache-Server (Server, die als Zwischenspeicher fungieren), E-Mail-Konten, Regierungsnetze und Intranetze.

[2] Vgl. Pike 2021

[3] Vgl. Grömer 2020, S.12ff

Abbildung 1: Illustration der verschiedenen Bereiche des Internets anhand des Eisbergmodells (Bundeskriminalamt 2019)

Eine Darstellungsart – nämlich ein Eisberg - taucht im Zusammenhang mit dem Surface Web und Deep Web immer wieder auf. Der Eisberg soll das Ausmaß der beiden Bereiche veranschaulichen, wobei dessen herausragender Teil für das Surface Web und der verborgene, im Wasser liegende Teil für das Deep Web steht (siehe die obige Abbildung). Wie klar erkannt werden kann, ist das Deep-Web um einiges größer als das Clearnet.

3.3 Dark Web

Das Dark Web oder auch Darknet stellt lediglich einen kleinen Ausschnitt des Deep Webs dar. In diesem Bereich sind sämtliche Verknüpfungen und Informationen verschlüsselt und anonymisiert. Um auf das Dark Web zugreifen zu können, sind spezielle Softwareanwendungen wie Tor oder I2P erforderlich, da herkömmliche Webbrowser keinen Zugriff ermöglichen. Diese Besonderheit ist einer der Hauptgründe, weshalb das Dark Web nicht von herkömmlichen Crawlern indexiert

werden kann. Zudem sind Webseiten im Dark-Web extra für den Zweck, unsichtbar zu sein, angelegt worden, was das Auffinden noch schwieriger macht.
Für Journalisten, Whistleblower, politisch Verfolgte und generell für alle Personen, die an der Wahrung ihrer Privatsphäre und ihrer Anonymität interessiert sind, ist diese Faktenlage insofern vorteilhaft, als sie im Dark Web sicher und anonym kommunizieren können. Gleichzeitig ist dieser digitale Raum allerdings auch ein beliebter Ort für Kriminelle, weil diese dort anonym Straftaten begehen können.[4]

[4] Vgl. Wright 2020, S.9

4 TOR – Ursprung, Aktuelles

In diesem Kapitel wird eine allgemeine Übersicht über Tor vermittelt. Diese beinhaltet die Ursprungsgeschichte und wichtige Begriffsdefinitionen in Bezug auf Tor.

4.1 Ursprung und Geschichte von Tor

Als das Internet am 30. April 1993 erstmals für die Öffentlichkeit frei nutzbar wurde, waren die zu diesem Zeitpunkt geltenden Sicherheitsstandards – aus jetziger Sicht - ausgesprochen primitiv und alles andere als sicher.
Heutzutage nutzen Webseiten standardmäßig HTTPS als Verbindungsprotokoll, wobei Daten mit starken Verschlüsselungsalgorithmen wie z.B.: AES, RSA verschlüsselt werden.[5]
In den Anfangszeiten des Internets hingegen waren alle Verbindungen zu Servern im Internet entweder teilweise oder gar nicht verschlüsselt. Jedermann konnte den Traffic mitlesen und auch leicht den Standort eines Users bestimmen. Dieser Umstand erlaubte insbesondere Staaten, Benutzer auszuspähen und zu überwachen. Ein Beispiel für die Überwachung der Internetkommunikation ist das US-amerikanische Spionagesystem „Echelon“, an dem neben den USA auch Neuseeland, Kanada, Australien und Großbritannien beteiligt sind. Dieses System dient zum Abfangen und Analysieren jeglicher Kommunikation - unabhängig davon, ob diese via Telefon, Fax, E-Mail-Übermittlung oder die Abwicklung von Bank-Transaktionen erfolgt.[6] Es wird beispielsweise vermutet, dass die NSA Echelon 1994 dazu benutzte, um das deutsche Unternehmen Enercon auf dessen neue Erfindungen in Bezug auf alternative Energien hin auszuspähen. Es wird weiters angenommen, dass die dabei abgeschöpften Informationen an amerikanische Firmen weitergegeben wurden, was diesen die Möglichkeit eröffnete, die betreffenden Erfindungen für sich patentieren zu lassen. In praktischer Hinsicht wäre dieser Spionageangriff jedenfalls möglich gewesen, da Enercon damals über

[5] Vgl. Capers/Pavlakoudis 2021

[6] Vgl. Hearst magazines 2001, S.69f

Telekom-Leitungen, die nicht (ausreichend) geschützt bzw. verschlüsselt waren, kommunizierte.[7]

Aufgrund dessen entwickelten David Goldschlag, Mike Reed, und Paul Syverson, allesamt Forscher des U.S. Naval Research Lab, die Idee eines Netzwerkes, das seine User anonymisiert und den Traffic verschlüsselt. Dieses Konzept ist als Onion Routing Protokoll bekannt. The Onion Routing, kurz Tor, ist eine Erweiterung dieses Konzepts. 2003 wurde dann der „reine" Tor-**Proxy** unter einer **Open Source** Lizenz veröffentlicht. Den Tor-Proxy kann man sich wie einen Internetzugang vorstellen. Die Tools für den Zugriff auf und die Nutzung des Proxys mussten von den Usern selbst programmiert werden. Um auch technisch weniger erfahrenen Personen Zugang zu ermöglichen, wurde durch die wachsende Community 2008 die Entwicklung des Tor-Browsers vorangetrieben.[8]

4.2 Ereignisse gestützt durch Tor

Da Tor immer benutzerfreundlicher wurde und immer mehr an Beliebtheit gewann, wurde es schnell zum Instrument für Aktivisten, Journalisten und **Whistleblowern,** die etwa wegen der Aufdeckung von Missständen und Straftaten teils massive Sanktionen von bloßgestellten Staaten und anderen Kompromittierten fürchten müssen. Tor spielte durch seine Funktionen beispielsweise während der Protestaktionen des Arabischen Frühlings 2010 eine wichtige Rolle und versetzte das Hacker-Kollektiv Anonymous 2011 in die Lage, sich an der Tunesischen Revolution zu beteiligen (durch Ausführung von **DDoS** Attacken auf Regierungsserver und die Weitergabe der Tor-Software zusammen mit anderen Scripts an Demonstranten).[9] Edward Snowden wiederum nutzte Tor 2013 dazu, anonym mit Journalisten zu kommunizieren und geheime Dokumente zu veröffentlichen.[10]

[7] Vgl. DIE ZEIT 1998

[8] Vgl. The Tor Project, Inc 2023 (History)

[9] Vgl. Web Archive 2011

[10] Vgl. The Tor Project, Inc 2023 (History)

Eine besonders wichtige Rolle spielt Tor derzeit in China. Durch Tor wird es chinesischen Journalisten und Whistleblowern ermöglicht, trotz starker Zensur (chinesische Firewall genannt), mit der „Außenwelt“ frei zu kommunizieren.[11]

[11] Vgl. Thoma 2004

5 Funktionsweise von Onion Routing / Tor

Zur Klärung der Frage, wie Tor funktioniert, muss zuerst eine Unterscheidung zwischen Tor, dem Tor-Netzwerk und dem Tor-Browser gemacht werden.

> "The Tor Browser is a web browser that anonymizes your web traffic using the Tor network, making it easy to protect your identity online. "[12]

Wie das Zitat erklärt, bedient sich die Benutzeroberfläche Tor-Browser des Tor-Netzwerks, das Verbindungen von Usern verschleiert und verschlüsselt.
Der Begriff Tor wird in dieser Arbeit kontextabhängig entweder als Synonym für das Tor-Netzwerk verwendet oder als Bezeichnung für die Implementierung der Idee, auf der das System basiert – Onion Routing (Kap. 5.1).

In der Informatik, besonders in der Kryptographie und Netzwerksicherheit, werden die Aliasnamen Alice, Bob und Carol häufig verwendet. Sie dienen dazu, Personen zu repräsentieren, die in verschiedenen Szenarien interagieren. Alice steht dabei typischerweise für eine Person, die eine Kommunikation beginnt, Bob für den Empfänger der Mitteilung und Carol für eine dritte Partei. Durch die Verwendung der Aliasnamen können komplexe Themen anschaulich vermittelt werden, ohne konkrete Namen nennen oder Rollen erläutern zu müssen.

5.1 Onion-Routing

Das Onion-Routing-Protokoll ist eine Methode zur Anonymisierung von Traffic. Beim Onion-Routing wird ein Traffic im Wesentlichen in Layers (Schichten) verschlüsselt und anschließend durch mehrere sog. **Nodes** oder Onion Relays (OR's) geleitet, bevor er am Zielserver ankommt. Tor - die Abkürzung für „The Onion Routing" - stellt eine Implementierung dieser Idee dar, bei der drei Nodes verwendet werden.

Onion-Routing funktioniert laut Aditya Saxena[13] und Dr. Mike Pound[14] wie folgt (dargestellt mit ebenfalls drei Nodes):

---

[12] Porup 2019

[13] Vgl. Saxena 2023

[14] Vgl. Pound 2017 (Tor)

Wenn „Alice“ sich mit einer Webseite im Clearnet verbindet, ist die Verbindung sowohl zurückverfolgbar als auch für Außenstehende sichtbar. Onion-Routing verhindert dies mithilfe eines Netzwerks aus Nodes. Statt sich direkt mit dem Server zu verbinden, verbindet sich Alice mit einer Node. Diese Node leitet ihre Verbindung zur nächsten weiter, usw. Jeden „Sprung“ von einer Node zur anderen nennt man einen „Hop“ – in diesem Beispiel geschieht das drei Mal. Die letzte Node verbindet sich dann mit dem eigentlichen Zielserver. Dasselbe passiert umgekehrt, wenn der Server eine Antwort schickt.

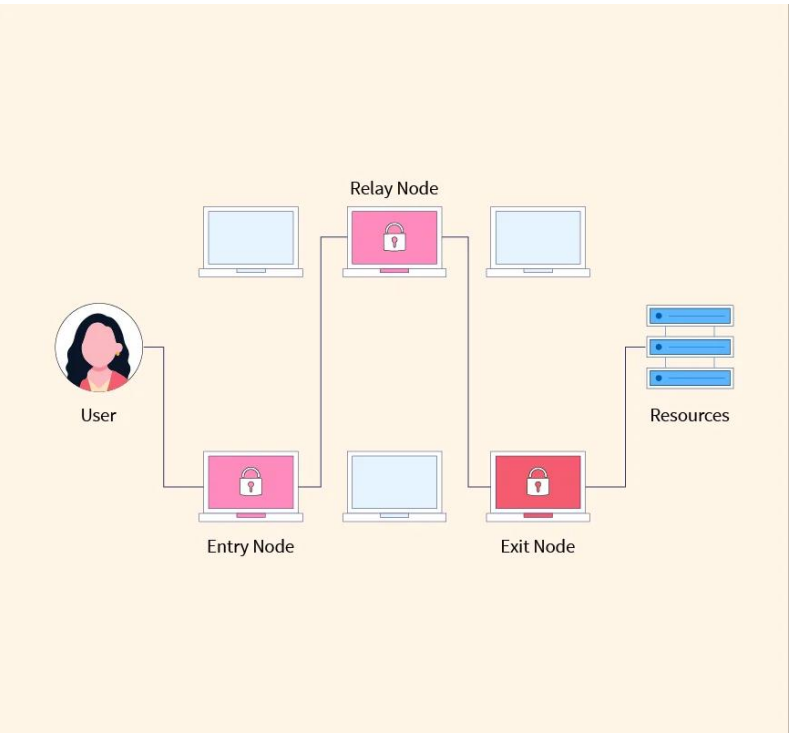

Abbildung 2: Verbindung über Onion-Routing-Protokoll (Saxena 2023)

Das Netzwerk aus Nodes bildet einen essenziellen Teil der Onion-Routing-Idee, doch allein reicht es nicht aus, um eine vollständige Anonymisierung der Verbindung zu gewährleisten. Um dieses Ziel zu erreichen, wird die Verbindung innerhalb des Netzwerks nochmal in mehreren Schichten (Layers) verschlüsselt. Dieses Verfahren gibt dem „Onion“-Routing auch ihren Namen – die Zwiebel (Onion) dient mit ihren Schichten als Symbolbild. Jede von Alice ausgewählte Node im Netzwerk teilt mit ihr einen Schlüssel, der zum Aufbau oder Abbau dieser Layers benutzt wird. Dadurch bleibt die Identität des Absenders geschützt und die Nachvollziehbarkeit des Ursprungs und des Ziels der Daten wird erschwert. Weitere Einzelheiten zur Funktionsweise dieser Verschlüsselungsprozesse finden sich in den Kapiteln 6.5 und 6.6.

5.2 Traffic über Tor ins Clearnet

Tor selbst besteht standardmäßig aus drei Onion-Routern mit **statischen IP-Adressen**. Man unterscheidet dabei drei Hauptkategorien von Onion-Routern. [15]

1. Guard oder Entry-Relay/Node
2. Middle-Relay/Node
3. Exit-Relay/Node

Der Verbindungsaufbau über Tor erfolgt in zwei Hauptschritten. Im ersten Schritt bezieht der Tor-Client von einem sog. „Directory Server“ eine Liste aller „Nodes”, also der oben genannten Onion-Routern. Dieser Prozess wird in der folgenden Grafik veranschaulicht.

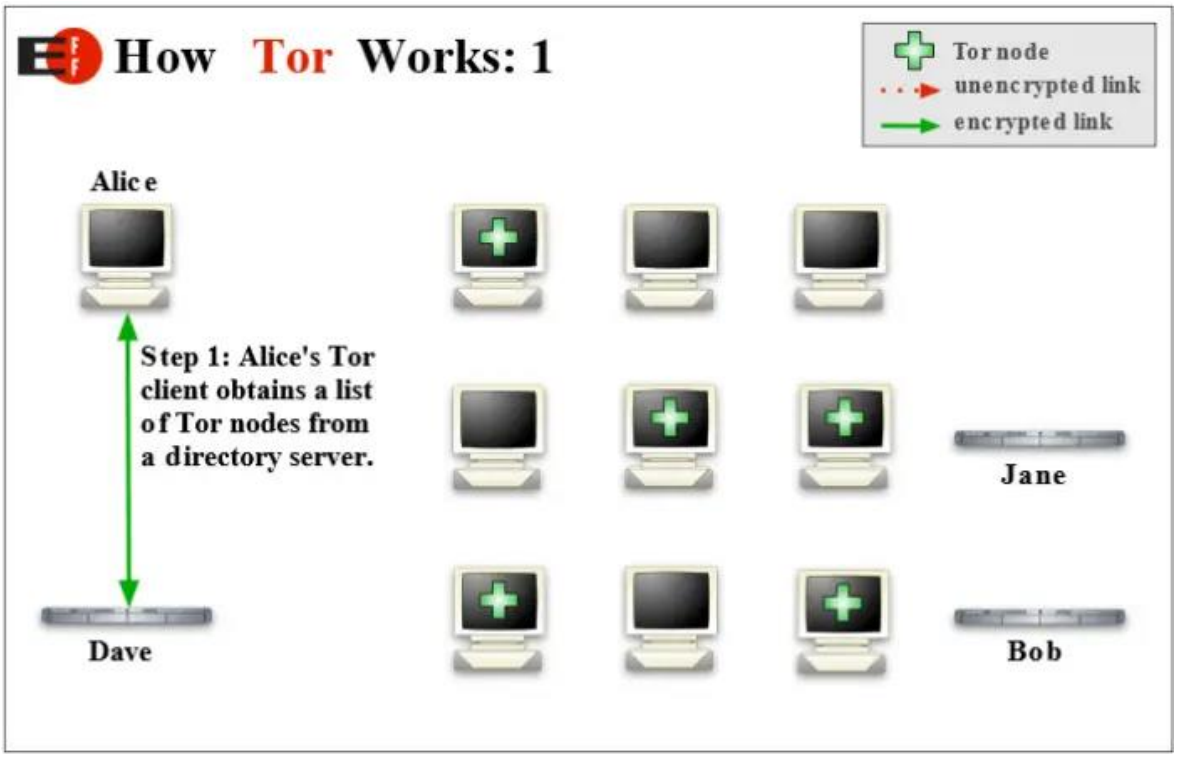

Abbildung 3: How Tor Works: Der Tor-Client holt sich die Liste der Tor-Nodes (Porup 2019)

Im zweiten Schritt beginnt der eigentliche Verbindungsaufbau. Der Tor-Client wählt aus der empfangenen Liste der Nodes nun drei Nodes - eine Guard-, eine Middle- und eine Exit-Node - aus. Danach wird sich mit dem Guard verbunden, von dem aus eine Verbindung zur Middle-Node aufgebaut wird. Die Middle-Node verbindet sich zum Schluss mit der Exit-Node, die den Endpunkt der Tor-Verbindung darstellt. Von dort aus geht der Traffic anschließend ins Clearnet.

[15] Vgl. The Tor Project, Inc 2023 (Types Of Relays)

Dieser Pfad wird auch „Circuit“ genannt. Da der Traffic über mehrere Nodes läuft, bevor er am Ziel ankommt, werden die IP-Adresse, der Standort und viele weitere **Metadaten** des eigentlichen Users verborgen. Angezeigt werden nur die Daten der Exit-Node.

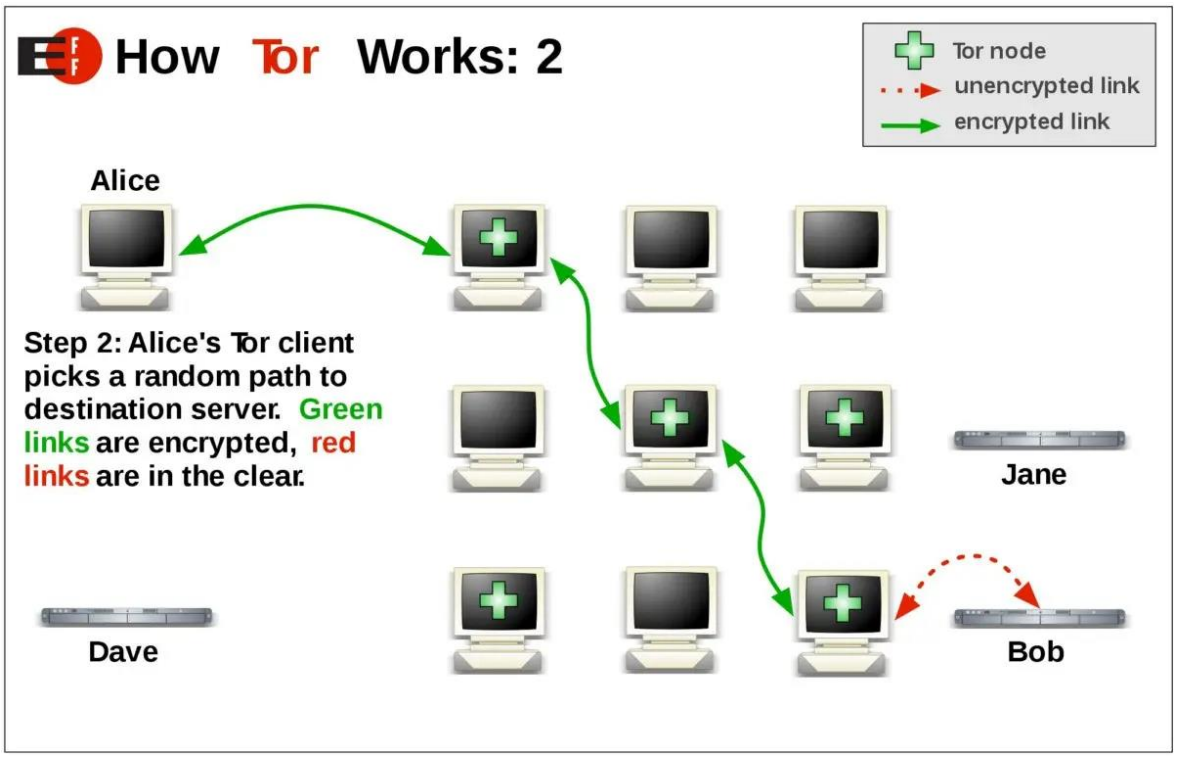

Abbildung 4: How Tor Works: Verbindungsaufbau über Tor mit einem Clearnet-Server (Quelle siehe Abb.3)

Ein Circuit ist nur für eine einzige Verbindung gültig. Wenn ein User eine andere Seite aufruft, also eine neue Verbindung zu einem neuen Host aufbaut, wird der zweite Schritt wiederholt. Das heißt, es wird ein neuer, zufälliger Circuit erstellt – die Identität wird mit dem Aufruf jeder neuen Seite verändert[16].

[16] Vgl. Porup 2019

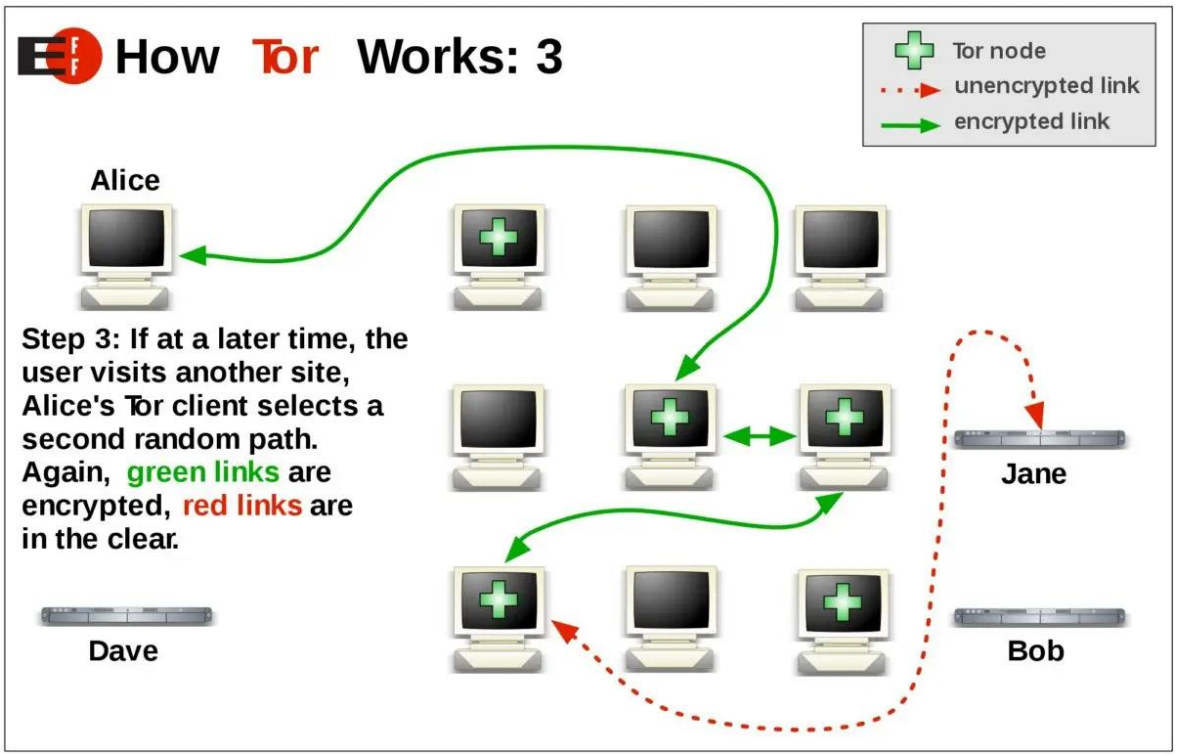

Abbildung 5: How Tor Works: Illustration eines neuen Verbindungsaufbaus (Quelle siehe Abb.3)

Hierbei muss beachtet werden, dass sich die Zielserver außerhalb des Tor-Netzwerks befinden. Dieser Umstand ist wichtig, da die Art bzw. Komplexität der Verbindung je nach Ziel (Clearnet oder Darknet) verschieden ist, wie im folgenden Kapitel erläutert wird.

5.3 Traffic über Tor ins Darknet

Der Verbindungsaufbau über Tor zu einem „Darknet-Server“, also einem sog. Onion-Server, funktioniert - vereinfacht gesagt - ähnlich wie eine Verbindung zu einem Clearnet-Server. Jedoch liegen Onion-Server innerhalb des Tor-Netzwerks, haben also auch einen eigenen Circuit. Daraus resultiert, dass der Traffic hierbei über sechs Nodes geleitet wird. Die ersten drei schützen die Identität des Users, die weiteren drei schützen die Identität des Onion-Servers. Somit können Daten wie die IP-Adresse, Standort, etc. weder vom User noch vom Server ermittelt werden.[17]

[17] Vgl. Aragon 2023

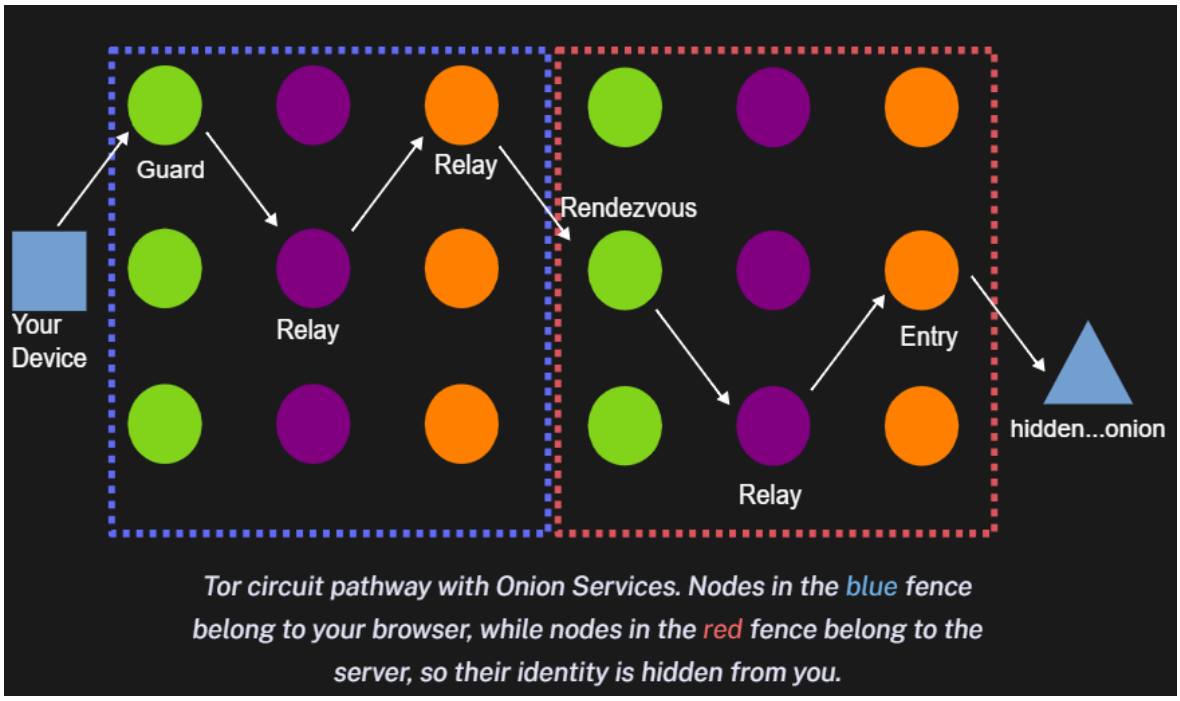

Abbildung 6: Verbindung zu einem Onion-Server (Aragon 2023)

Diese sechs Nodes sind u.a. auch ein Grund dafür, warum Onion-Server langsamer als Server im Clearnet sind - Tor-Nodes verfügen meist nur über eine geringe Bandbreite, die bei sechs Nodes mehr zum Tragen kommt als bei drei. Zudem befinden sich die einzelnen Nodes oft in verschiedenen Kontinenten und Staaten (z.B. einer in Österreich, einer in den USA, einer in Japan), was die Verbindung zusätzlich verlangsamt. Hinzu kommt, dass die Mehrheit der Onion-Server nicht in großen Rechenzentren operiert und daher noch weniger Ressourcen zur Verfügung stehen.

6 Verschlüsselung des Traffics von Tor

Tor vereint symmetrische und asymmetrische Verschlüsselung mit dem Diffie-Hellman-Schlüsselaustausch und dem TLS-Protokoll zu einer mächtigen Kombination. Wie in Kapitel 5.1 Onion-Routing schon erklärt wurde, wird der Traffic durch Nodes geschickt, bevor er beim Zielserver ankommt. Da das reine Durchsenden des Traffics keine vollständige Anonymität bietet, wird dieser noch in mehreren Layern verschlüsselt.

Um zu verstehen, wie die Verschlüsselung genau erfolgt, werden die oben genannten Verschlüsselungsarten und Protokolle in den vier folgenden Kapiteln (6.1 bis 6.4) näher erklärt.

6.1 Symmetrische Verschlüsselung / Secret-Key - AES

Die symmetrische Verschlüsselung braucht zum Verschlüsseln und Entschlüsseln nur einen Schlüssel, der auch als Secret-Key bezeichnet wird. Diese Verschlüsselung ist sehr sicher, wenn der Schlüssel ausreichend lang ist (z.B. >= 256 Bits), da die Stärke der Verschlüsselung im Verhältnis zum aufgewandten Rechenaufwand weit besser ist relativ zu anderen Verschlüsselungsarten (siehe Kap. 6.2).
Beispiele sind AES (Advanced Encryption Standard), DES (Data Encryption Standard) und Blowfish[18]. Am verbreitetsten ist der AES-Algorithmus. In der folgenden Grafik ist der kryptografische Prozess dargestellt. Mit **Plaintext** ist hierbei ein unverschlüsselter Text gemeint. **Ciphertext** ist der verschlüsselte Text.

[18] Vgl. Elektronik Kompendium (Symmetrische Kryptografie)

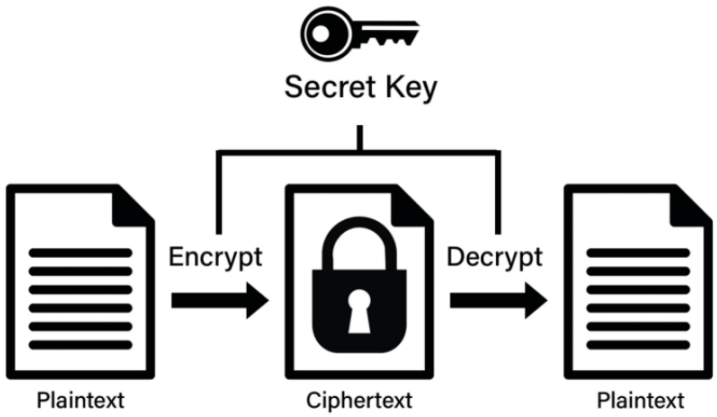

Abbildung 7: Symmetrisches Verschlüsselungsverfahren (Baka, Schatten 2020, S.6)

Zur besseren Veranschaulichung folgt hier ein Beispiel für eine symmetrische Verschlüsselung:

Alice und Bob schreiben sich oft Nachrichten, wollen dabei aber nicht, dass Carol ihrer Konversation folgen kann. Deshalb treffen sie sich persönlich und einigen sich auf einen Schlüssel: „Geheim23“. Somit können die beiden ab nun untereinander Nachrichten verschlüsseln, entschlüsseln und lesen. Carol hingegen kann ihrer Konversation nicht mehr länger folgen und auch nicht daran teilnehmen, da sie nicht über den festgelegten Schlüssel verfügt. Um Carol an der Kommunikation teilhaben zu lassen, müssten sich alle drei Beteiligten persönlich treffen und einen neuen Schlüssel vereinbaren oder müsste Carol z. B. über WhatsApp der Schlüssel bekanntgegeben werden.[19] Wie soll aber eine Einbeziehung von Carol in die Kommunikation vonstattengehen, wenn ein Treffen nicht möglich ist, weil Carol z.B. auf einem anderen Kontinent lebt, oder man WhatsApp nicht vertraut?

Es stellt sich also folgende Frage: Wie gelangen zwei Instanzen an denselben Schlüssel, wenn keine oder keine sichere Kontaktaufnahme möglich ist?

6.2 Asymmetrische Verschlüsselung / Public-Private-Key - RSA

Asymmetrische Verschlüsselung ist der Gegensatz zur symmetrischen Verschlüsselung. Diese funktioniert mit jeweils zwei Schlüsseln pro Instanz, einem Public-Key zum Verschlüsseln von Daten und einem Privat-Key zum Entschlüsseln.

[19] Vgl. Das, Veni Madhavan 2009, S.4

Das häufigste Verfahren hierbei ist der RSA-Algorithmus. Die folgende Grafik veranschaulicht dieses Verfahren.

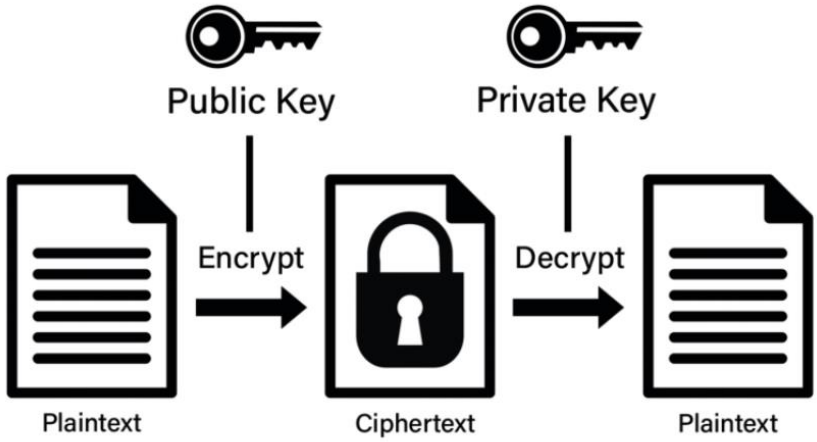

Abbildung 8: Asymmetrischer Verschlüsselungsverfahren (Quelle: siehe Abb.7)

Public- und Privat-Keys sind zwei unterschiedliche Schlüssel. Sie hängen insofern zusammen, als dass beide den jeweils anderen zum Funktionieren benötigen. Gerät der eine Schlüssel in Verlust, ist der andere wertlos. Selbiges gilt, wenn der Private-Key nicht mehr nur einem User zugänglich ist, sondern mehreren. Denn dann ist jegliche Kommunikation kompromittiert, da auch Fremde mitlesen könnten. Dies könnte zum Beispiel im Zuge eines Hackerangriffs passieren.
Die asymmetrische Verschlüsselung löst somit das Problem der Schlüsselübertragung. Alice muss nur ihren Public-Key Pu^A öffentlich machen. Daten werden anschließend von Bob mit Pu^A verschlüsselt und können danach mit dem Private-Key Pr^A von Alice wieder entschlüsselt werden. Diese Verschlüsselungsart löst somit das Problem der Schlüsselübertragung, wodurch eine nahtlos verschlüsselte Kommunikation ermöglicht wird.[20]

Das Verfahren sei an folgendem Beispiel veranschaulicht:
Alice möchte ihrem Freund Bob abhörsicher mitteilen, wann sie in München ankommt. Sie kontaktiert ihn also und teilt ihm mit, sie müsse ihm eine geheime Botschaft zukommen lassen. Bob generiert daraufhin einen Privat-Key Pr^B und einen Public-Key Pu^B. Dann schickt er Pu^B Alice. Alice verschlüsselt ihre Nachricht mit Pu^B und schickt Bob ihren **Ciphertext**. Anschließend entschlüsselt Bob den

[20] Vgl. Das, Veni Madhavan 2009, S.4

Ciphertext mit Pr^B und kann nun die Nachricht im **Plaintext** lesen. Will Alice nun Nachrichten von Bob empfangen, muss sie denselben Prozess wie Bob durchlaufen.

6.3 Diffie-Hellman-Merkle-Schlüsselaustausch

Eine andere, sehr beliebte Methode, die Prinzipien des asymmetrischen Verfahrens nutzt, ist der Diffie-Hellman-Merkle-Schlüsselaustausch (kurz DH).

Der Diffie-Hellman-Merkle-Schlüsselaustausch ist laut Luber:

> „[…] ein Verfahren, mit dem sich ein gemeinsamer Sitzungsschlüssel zwischen zwei Kommunikationspartnern sicher über ein potenziell unsicheres Übertragungsmedium vereinbaren lässt."[21]

Im Gegensatz zu RSA dient diese Methode aber ausschließlich zur Übertragung eines symmetrischen Schlüssels.

Der Ablauf gestaltet sich – im Großen und Ganzen – wie folgt:[22]

1. Alice und Bob einigen sich auf zwei Primzahlen, p und g, auch Diffie-Hellman-Parameter genannt, wobei $g < p$ sein muss.
2. Gleichzeitig wählen beide jeweils eine zufällige Zahl $x < p$ und $y < p$.
3. Nun berechnet Alice $A = g^x \bmod p$
   und Bob $B = g^y \bmod p$.
4. Anschließend werden *A* und *B* untereinander ausgetauscht - sie stellen daher aufgrund ihrer Funktion Public-Keys dar.
5. Dann berechnet Alice $K_1 = B^x \bmod p$
   und Bob $K_2 = A^y \bmod p.$
   Die Ergebnisse *K1* und *K2* beider Rechnungen sind immer identisch und bilden somit den symmetrischen Schlüssel zur Verschlüsselung jeder weiteren Kommunikation.

Wichtig zu beachten ist, dass in Texten und Berichten meist alternative Begriffe und Schreibweisen für den DH angewendet werden. So werden in Bezug auf die Mathematik g^x/g^y häufig als Private-Keys, *A/B* häufig als Public-Keys bezeichnet. Gängig ist auch die Verallgemeinerung bzw. Verkürzung der mathematischen

[21] Luber 2019

[22] Vgl. Elektronik Kompendium (Diffie-Hellman-Merkle-Schlüsselaustausch)

Formeln von A/B zu lediglich g^x/g^y. Welche Schreibweiße gewählt wird, ist sowohl vom Kontext als auch von der Tiefe des gegenständlichen Textes abhängig.

6.4 Diffie-Hellman in Python

Das folgenden Script – geschrieben in Python – beinhaltet eine simple Implementierung eines 1024-Bits-DH-Schlüsselaustausches. Generatoren und Funktionen werden hierbei von der „cryptography"-Bibliothek bereitgestellt, die vor Anwendung mit „*pip install cryptography*" im Terminal installiert werden muss.[23] [24]
Der Ablauf gestaltet sich wie folgt: Zuerst werden die DH-Parameter definiert bzw. generiert (Z.7). Dann werden die Private- und Public-Keys berechnet (Z.10-13). Im Anschluss müssten die Public-Keys eigentlich untereinander ausgetauscht werden, was hier aus Gründen der Übersicht ausgelassen wurde. Stattdessen berechnen Alice und Bob direkt die symmetrischen Schlüssel K_1 und K_2 (Z.17,21), die am Schluss mit „*assert*" auf ihre Übereinstimmigkeit hin überprüft werden (Z.24). Wenn K_1 ungleich K_2 ist (K_1 *!=* K_2), dann wird eine **Exception** mit benutzerdefinierter Fehlermeldung ausgegeben. In diesem Fall: *"Fatal-Error: K1 != K2"*.

```
1  # Importieren der Kryptographie - Bibliothek
2  from cryptography.hazmat.primitives.asymmetric import dh
3
4
5  # Generieren der großen DH-Primzahl p
6  # g wird hier vordefiniert --> g = 2
7  dh_parameter = dh.generate_parameters(generator=2, key_size=1024)
8
9  # Berechnung der Private- und Public-DH-Keys von Alice und Bob
10 alice_gX_private_key = dh_parameter.generate_private_key()     # g^x
11 alice_A_gXmodp_public_key = alice_gX_private_key.public_key() # A = g^x mod p
12 bob_gY_private_key = dh_parameter.generate_private_key()       # g^y
13 bob_B_gYmodp_public_key = bob_gY_private_key.public_key()     # B = g^y mod p
14
15 # Berechnung von K1 durch Alice mithilfe Bobs Public-Key
16 # K1 = B^x mod p
17 alice_K1_shared_key = alice_gX_private_key.exchange(bob_B_gYmodp_public_key)
18
19 # Berechnung von K2 durch Bob mithilfe von Alices Public-Key
20 # K2 = A^y mod p
21 bob_K2_shared_key = bob_gY_private_key.exchange(alice_A_gXmodp_public_key)
22
23 # Prüfen ob K1 == K2
24 assert alice_K1_shared_key == bob_K2_shared_key, "Fatal-Error: K1 != K2"
```

[23] Vgl. Kehrer 2023

[24] Vgl. Cryptography Docs 2023

6.5 HTTPS / TLS (Transport Layer Security)

Sichere Dienste, wie zum Beispiel HTTPS, nutzen eine erweiterte Diffie-Hellman-Methode, um einen symmetrischen Schlüssel für eine Sitzung zu übertragen. Asymmetrische Verfahren sind mathematisch komplexer als symmetrische, wodurch diese auch mehr Zeit bzw. Rechenleistung in Anspruch nehmen. Große Datenmengen können daher nicht schnell genug verarbeitet werden, weshalb zum Verschlüsseln von Daten ausschließlich symmetrische Verschlüsselungen verwendet werden. Die modernste Implementierung eines Protokolls, das diese Umstände berücksichtigt, ist TLS (Version 1.3).
Verbindet man sich als Client mit einer HTTPS Webseite, findet im Hintergrund ein TLS-Handshake statt. Dieser Handshake ist für den Austausch der Sitzungsschüssel, die später für die Verschlüsselung der Sitzung genutzt werden, zuständig. Ein standardmäßiger TLS-Handshake läuft laut Pound[25] und Baka/Schatten[26] folgendermaßen ab:

1. Der Client sendet ein sog. „Client Hello“ zum Server, das u.a. die unterstützten Verschlüsselungsversionen (AES, RSA, TLS…) sowie eine zufällige Zahl *Z* beinhaltet.
2. Daraufhin antwortet der Server mit einem „Server Hello“. In dieser Nachricht befinden sich drei Informationen - die vom Server ausgewählten Verschlüsselungsversionen, das Zertifikat des Servers, das auch den Public-Key des Servers beinhaltet (meistens RSA), und eine zufällige Zahl *Y*.
3. Der nächste Schritt wurde von Baka/Schatten nur sehr kurz angeschnitten bzw. wenig detailreich dargestellt, da dies nicht im Hauptfokus des Buches lag. Für Pound hingegen ist dieser Schritt essenziell, deshalb erklärt er ihn im Detail:
   Anschließend sendet der Server eine „Server-Key Exchange“ Nachricht. In dieser liegen optional die Diffie-Hellman Parameter g und p [meist sind jene Parameter immer dieselben], sowie der DH-Parameter des Servers g^x.
4. In weiterer Folge sendet der Server die Nachricht „Server Hello Done“.

[25] Vgl. Pound 2020 (TLS)

[26] Vgl. Baka/Schatten 2020, S.8ff

5. Der Client prüft nun, ob der Server wirklich vertrauenswürdig ist, indem er das Zertifikat auf seine Gültigkeit hin überprüft [die Darlegung des genauen Ablaufs dieser Nachfrage würde den Rahmen dieses Kapitels sprengen, weshalb darauf verzichtet wurde]. Nur wenn sich das Zertifikat als vertrauenswürdig erweist, sendet der Client eine „Client-Key Exchange“ Nachricht. In dieser steckt der DH-Parameter des Clients g^y.
6. Abschließend sendet auch der Client eine „Client Hello Done“ Nachricht.
7. Beide, der Server und der Client, berechnen nun aus g^x und g^y das sog *Pre*-Master-Secret. Für erweitere Sicherheit werden dann noch die beiden zufälligen Zahlen *Z* und *Y* mit dem Pre-Master-Secret kombiniert. Output dieser Kombination ist das Master-Secret *K*.

   Hier endet die Erklärung von Baka/Schatten, allerdings folgen laut Pound noch zwei weitere Schritte:

8. Zunächst sendet der Client „Change Cipher Spec“, was bedeutet, dass die nächste Nachricht des Clients mit dem Master-Secret *K* verschlüsselt sein wird. Danach sendet der Client eine Art Zusammenfassung der bisherigen „Konversation“ – verschlüsselt mit *K*.
9. Daraufhin sendet der Server gleich wie der Client ein „Change Cipher Spec“ mit einer ebenfalls verschlüsselten Zusammenfassung der Kommunikation.

6.6 Verbindungsaufbau mit einzelnen Onion-Routern

In diesem Abschnitt wird erläutert, wie der Tor-Client die kryptographischen Schlüssel der einzelnen Nodes erhält, dabei die Anonymität bewahrt und gleichzeitig sicherstellt, dass die Schlüssel nicht kompromittiert werden.

Um eine sichere und einwandfrei funktionierende Verbindung aufzubauen, nutzt Tor die Kombination aus TLS-Protokoll und dem Diffie-Hellman-Schlüsselaustausch. TLS sorgt für eine nahtlose Verschlüsselung zwischen den einzelnen Nodes (zwischen den Hops), Diffie-Hellman für die sichere Übertragung der Daten zwischen Tor-Client und Ziel. Der ganze Prozess gestaltet sich wie folgt:

Die Datenpakete, die das Tor Netzwerk passieren, sind klar strukturiert und organisiert. Sie werden Zellen (Cells) genannt und sind immer 512 Bytes groß. Insgesamt gibt es zwei Cell-Typen, aber der Einfachheit halber stellt man sich den Aufbau beider als zwei Abschnitte vor. Der erste Abschnitt enthält den Befehl und der zweite Abschnitt enthält die Daten – Payload genannt.
Die vier wichtigsten Befehle sind in der folgender Tabelle dargestellt. Sie werden im Laufe dieses Kapitels näher erläutert.

Befehl	Beschreibung
create	Fordert eine neue Node auf, sich dem Circuit anzuschließen → Erweiterung des Circuits
relay-extend	Fordert eine Node auf, sich mit einer weiteren zu verbinden und diese aufzufordern, sich dem Circuit anzuschließen.
Created / *relay-extended*	Statusrückmeldungen über das (erfolgreiche) Eingliedern einer Node in den Circuit.

Der Tor Client, nennen wir ihn Alice, erstellt Circuits schrittweise – ein **Hop** nach dem anderen. Dazu verbindet sich Alice zuerst mit einem Directory-Server und holt sich eine Liste mit verfügbaren Nodes. In der Liste befindet u.a. auch der RSA-Public-Key jeder Node – Onion-Key genannt.
Alice wählt nun eine Guard-, eine Middle-, und eine Exit-Node aus der Liste aus. Im weiteren Verlauf baut Alice eine TLS-Verbindung zur Guard-Node – Bob – auf. Wenn die Verbindung steht, sendet Alice ihm eine *create-cell*, als Payload den ersten Diffie-Hellman Parameter g^{x1} verschlüsselt mit dem Onion-Key von Bob.
Dieser muss nun beweisen, dass er wirklich der ist, für den er sich ausgibt. Aus diesem Grund wurde der DH-Parameter mit dem Onion-Key von Bob verschlüsselt, von dem nur er das Gegenstück, also den Private-Key, besitzen sollte. Dadurch wird einerseits seine Identität bestätigt, andererseits verhindert, dass andere Nodes sich als ihn ausgeben. Kann der Payload erfolgreich entschlüsselt werden, antwortet Bob mit einer *created-cell*, als Payload den zweiten Diffie-Hellman Parameter g^{y1} sowie den **Hash** des symmetrischen Schlüssels $K_1=g^{x1y1}$. Jetzt können Alice und Bob mithilfe von K_1 kommunizieren.

Um den Circuit um eine Node zu erweitern, sendet Alice eine *relay-extend-cell* zu Bob. In der Zelle inkludiert ist die Adresse der nächsten Node, Carol, und als Payload sendet Alice den DH-Parameter g^{x2} verschlüsselt mit Carols Onion-Key. Bob führt den darin enthaltenen Befehl *relay-extend* aus, indem er den Payload in eine *create-cell* kopiert und an Carol weiterleitet – das ganze ebenfalls über eine TLS-Verbindung zwischen Bob und Carol. Auch sie muss zuerst mithilfe ihres Private-Keys beweisen, dass sie „die echte Carol" ist. Darauffolgend antwortet sie nun mit einer *created*-cell, als Payload sendet sie den zweiten DH-Parameter g^{y2} und den **Hash** von $K_2=g^{x2y2}$. Im Anschluss kopiert Bob den Payload wieder in eine *relay-extended-cell* und schickt diese zurück zu Alice. Jetzt können Alice und Carol mithilfe von K_2 kommunizieren.[27]

> "To extend the circuit to a third node or beyond, Alice proceeds as above, always telling the last node in the circuit to extend one hop further."[28]

Wie das Zitat erklärt, kann durch Wiederholung des oben beschriebenen Prozesses der Circuit immer um eine Node erweitert werden. Bei Tor wird dieser Schritt standardmäßig zweimal wiederholt, am Ende besteht der Circuit also aus drei Nodes. Obwohl es theoretisch möglich ist, mehr als drei Nodes in einem Circuit zu verwenden, wird dies nicht empfohlen. Es wird durch dieses Verfahren nämlich kein höheres Maß an Anonymität erreicht. Überdies tritt eine weitere Verlangsamung des Datenverkehrs ein.

6.7 Verschlüsselung des Traffics über Tor

Will sich Alice also mit dem Tor-Netzwerk verbinden, wählt sie drei Nodes (N_1, N_2, N_3) aus. Dann werden ihr von jeder Node die jeweiligen Schlüssel übermittelt (K_1, K_2, K_3), die zum Verschlüsseln des Traffics genutzt werden.
Den Prozess hinter dem Senden von Traffic durch dieses Netzwerk kann man sich wie das Schälen einer Zwiebel vorstellen:

1) Alice, die in Besitz aller drei Schlüssel ist, verschlüsselt ihre Anfrage nacheinander mit K_3, K_2, K_1 und sendet sie der ersten Node (N_1).

[27] Vgl. The Tor Project, Inc (Onion Router)

[28] Vgl. The Tor Project, Inc (Onion Router)

2) N_1 entschlüsselt die erste Schicht der Nachricht mit ihrem K_1. Allerdings ergibt der entschlüsselte Text keinen Sinn, also schickt N_1 das Ergebnis weiter zur zweiten Node N_2.
3) Dasselbe wie N_1 in Schritt 2 führt nun auch N_2 aus. Aber auch hier ergibt das Ergebnis keinen Sinn, es wird also zur dritten Node N_3 weitergeleitet.
4) N_3 entschlüsselt die dritte Schicht des Payloads mithilfe ihres K_3. Aber anders als bei N_1 oder N_2 ergibt der entschlüsselte Text jetzt Sinn, da dieser z.B. eine HTTP-Anfrage enthält.

   Alle sich in dem Payload befindlichen Anweisungen werden nun von N_3 nacheinander abgearbeitet bzw. ausgeführt.

Wenn der Server Alice eine Antwort schickt, werden diese Schritte in umgekehrter Reihenfolge ausgeführt. Bildlich kann man sich diesen Vorgang wie den Wiederaufbau einer Zwiebel vorstellen.

1) N_3 verschlüsselt die Antwort des Servers mit ihrem K_3 und schickt sie zu N_2.
2) N_2 und N_1 wiederholen nun nacheinander denselben Prozess wie N_3 in Schritt 1, jeweils ihre Schicht der Verschlüsselung dranhängen.
3) Letztendlich kommt die Nachricht bei Alice an. Da Alice im Besitz aller drei Schlüssel ist, ist sie in der Lage, die Nachricht komplett zu entschlüsseln und zu lesen.

7 Schwachstellen von Tor

Tor ist ein Netzwerk, das den Traffic der User anonymisiert und es ihnen so ermöglicht, sowohl im Clearnet als auch im Darknet sicher zu surfen. Es spielt daher eine wichtige Rolle für Aktivisten in autoritären Regimen. Gleichzeitig kann es jedoch leicht für die Vornahme illegaler Aktivitäten missbraucht werden.
Aus diesem Grund gibt es viele Instanzen, die zur Ausübung von Zensur oder zur Aufdeckung von im Web begangenen Straftaten daran interessiert sind, User zu de-anonymisieren. Derartige Instanzen sind beispielsweise die NSA, das FBI, Interpool und das BKA. Um die De-Anonymisierung zu bewerkstelligen, müssen Schwachstellen im System gefunden und angegriffen werden.
In diesem Hauptkapitel werden die Schwachstellen des Tor-Netzwerks aufgezeigt sowie mögliche Angriffspunkte analysiert. Um den Inhalt verständlicher zu machen, wird in diesem Kapitel der Angreifer „Eve“ genannt.

7.1 Endpoint Traffic Correlation

Ein Verfahren, mit dem User de-anonymisiert werden können, ist Endpoint Traffic Correlation. Hierbei wird der Traffic von Nodes auf das Vorliegen bestimmter Muster und Gemeinsamkeiten hin verglichen. Um dies bewerkstelligen zu können, muss Eve in der Lage sein, sowohl den Traffic zwischen User und Guard-Node als auch den Traffic zwischen Exit-Node und Zielserver mitzuschneiden.[29] Ein Beispiel zeigt folgendes Szenario:
Alice schickt Bob über Tor drei Nachrichten von jeweils 10 KB innerhalb von sechs Sekunden (eine Nachricht je zwei Sekunden). Der mögliche Traffic einer Guard-Node wird in der folgenden Grafik vereinfacht dargestellt. Alices Nachrichten sind bei der 14., 16. und 18. Sekunde zu finden. Die Sendezeitpunkte sind rot markiert.

[29] Vgl. Pound 2017 (Tor)

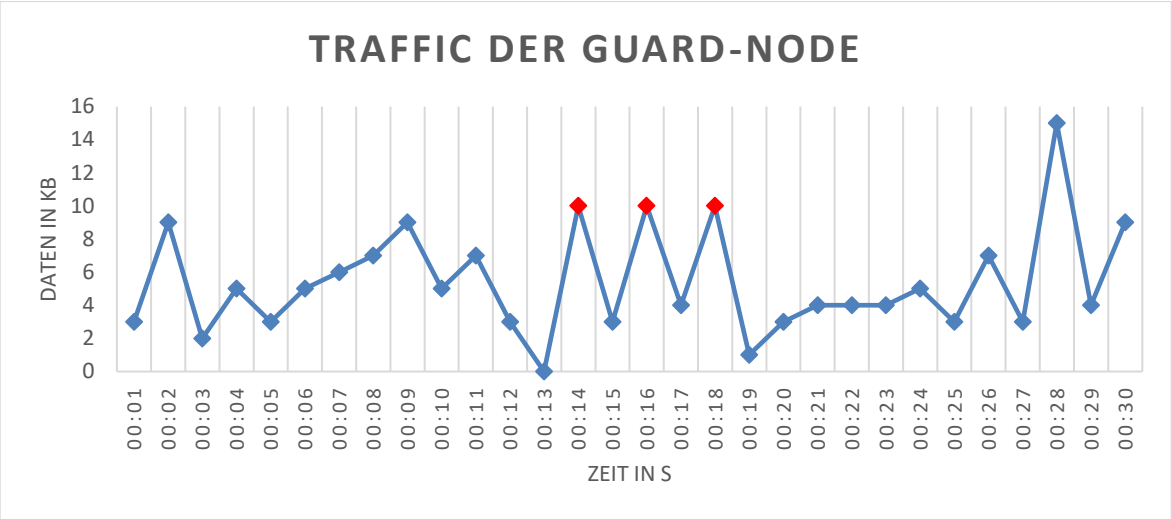

Abbildung 9: Traffic der Guard-Node (Eigendarstellung)

Der mögliche Traffic der Exit-Node wird in der nächsten Grafik dargestellt. Alices Nachrichten - nun bei der 15., 17. und 18. Sekunde - sind schwarz markiert. Die Zeitverschiebung um eine Sekunde resultiert aus der etwas langsamen Datenübertragung des Tor-Netzwerks.

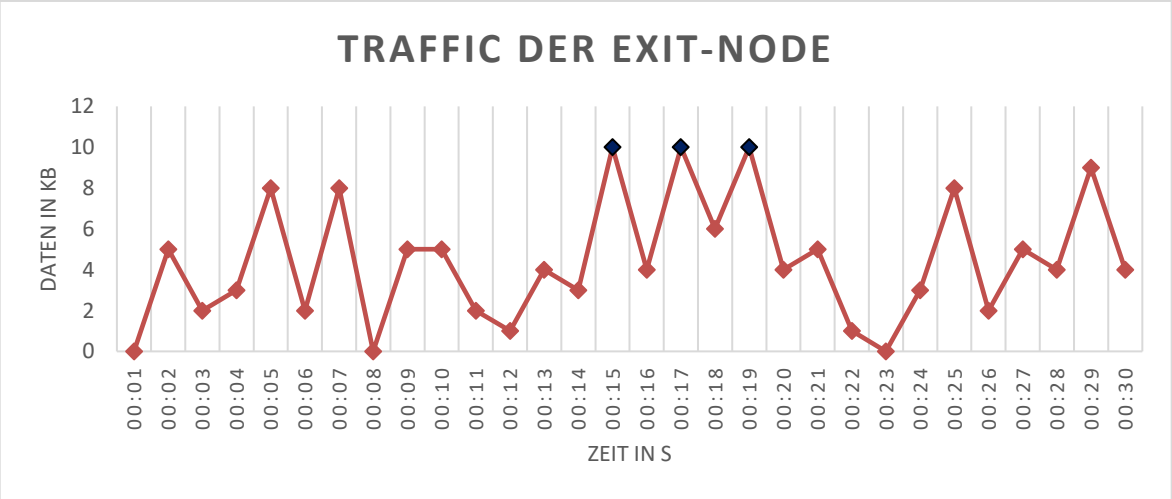

Abbildung 10: Traffic der Exit-Node (Eigendarstellung)

Wenn Eve diese beiden Traffic-Analysen nun miteinander vergleicht, kann eine Ähnlichkeit entdeckt werden, die sich im Zeitfenster zwischen Sekunde 14 und 19 zeigt. Hier nämlich sind drei Extrema der Guard-Node identisch mit drei Extrema der Exit-Node. Siehe die folgende Grafik:

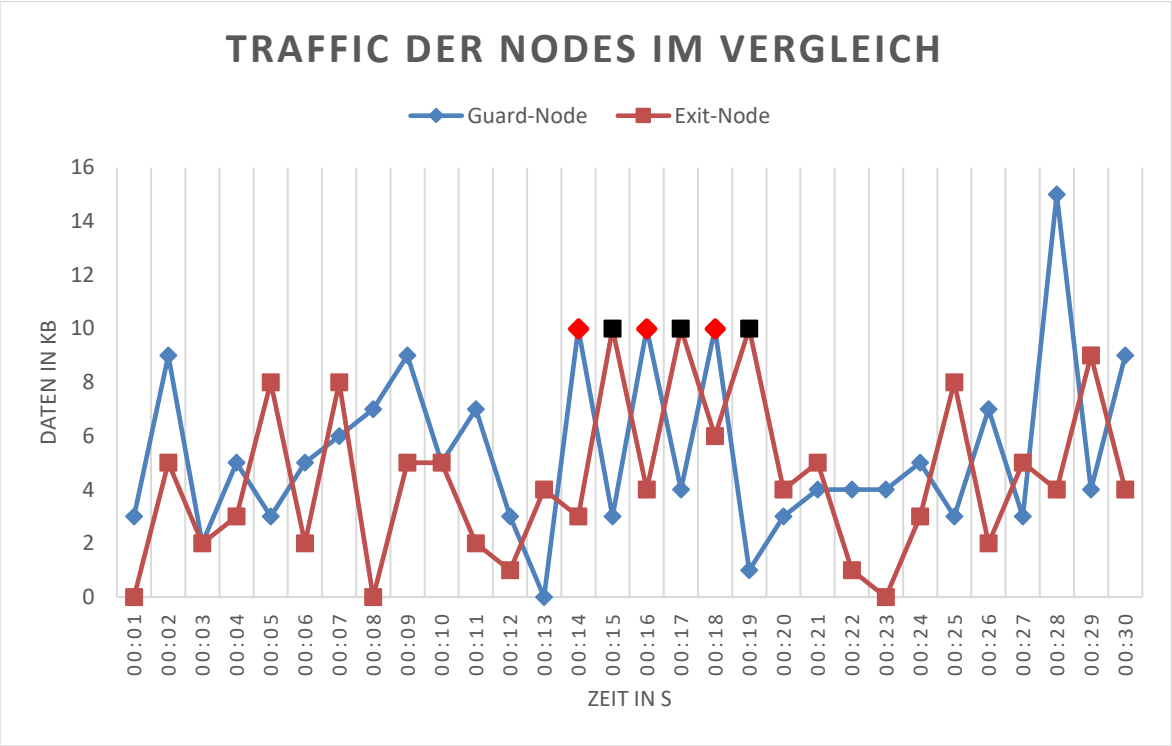

Abbildung 11: Traffic der Nodes im Vergleich (Eigendarstellung)

Jetzt muss Eve nur in den Traffic-Aufzeichnungen der Guard-Node nachschlagen, welche IP-Adresse die drei Extrema verursacht hat, und schon ist die IP-Adresse von Alice kompromittiert. Den Nachrichteninhalt kann Eve allerdings nicht lesen, da ihm die zu dessen Entschlüsselung benötigten kryptographischen Schlüssel fehlen (siehe Kap. 7.9).

7.2 Sybil Angriff

Der Sybil-Angriff ist eine Art Erweiterung der Endpoint Traffic Correlation. Die Bezeichnung leitet sich von einer Novelle von Flora Schreiber mit dem Titel „Sybel“ ab, welche eine Frau mit multipler Persönlichkeitsstörung beschreibt. Hierbei muss Eve einen großen Teil der Tor-Nodes kontrollieren können.[30] Was dabei erreicht werden soll, wird im folgenden Zitat dargelegt:

> “These nodes may cooperate by sharing traffic information and correlating and routing traffic to achieve easier anonymity compromise.”[31]

[30] Vgl. Binance Acadamy 2018

[31] Årnes 2022, S.119

Der Vorteil dieses Angriffs ist, dass weder ein direkter Zugang zum Traffic des Clients bzw. zu seinem System noch ein Aufenthalt in dessen Nähe erforderlich ist.

7.3 Tor Node Availability - Snowflakes

Diese Angriffstaktik zielt nicht darauf ab, Tor direkt zu attackieren, sondern es von vornherein gar nicht zu einer Verbindung kommen zu lassen. Wie in Kapitel 5.2 beschrieben, muss sich Alice mit einem Directory-Server verbinden, um die Adressen der Nodes downzuloaden. Der Nachteil hierbei ist, dass diese Liste somit öffentlich bzw. komplett einsehbar für jeden ist. Staaten, die eine weitgehende Zensur ausüben, können daraufhin alle dort gelisteten Nodes für ihr Land sperren, sodass das Tor-Netzwerk nicht mehr erreichbar ist. Die Lösung hierfür sind Bridges, gängig auch unter dem Namen Snowflakes bekannt, die nicht öffentlich in den Directory-Servern gelistet werden und somit auch in zensierten Ländern eine Verbindung zu Tor gewährleisten. Sie fungieren wortwörtlich als Brücke zwischen dem User und dem eigentlichen Tor-Netzwerk. Wegen ihrer „Unsichtbarkeit“ ist es sehr schwer, sie alle zu blocken, zumal im Gegensatz zu Nodes die IP-Adressen der Bridges meist **dynamisch** sind.[32]

7.4 Zensurfreie und sichere Veröffentlichung

Ein grundsätzliches Problem bei der Verbreitung von Software stellen deren Zugänglichkeit und Echtheit dar. Server, welche Tools wie Tor veröffentlichen und zum Download bereitstellen, können landesintern durch die Justiz im Auftrag des Staates leicht ausfindig gemacht und abgeschaltet werden. Befinden sich die Server in einem anderen Land, können diese von Zensur ausübenden Staaten gesperrt oder durch Cyberangriffe funktionsunfähig gemacht werden. Eine andere Gefahr besteht darin, dass sich Geheimdienste Zugang zum Server, der die Software anbietet, verschaffen und versuchen könnten, diese für ihre Zwecke zu manipulieren.[33]

[32] Vgl. Årnes 2022, S.119f

[33] Vgl. Årnes 2022, S.119f

Ein sicheres Verbreitungsmedium für die Software zu finden, ist daher eine der wichtigsten Komponenten zur Gewährleitung der Sicherheit des Tor-Users.

7.5 HTTP-Verbindungen

Die Nutzung von **HTTP**-Verbindungen ist mitunter ein Grund dafür, warum User trotz Benutzung von Tor zurückverfolgt werden können. Verbindet man sich nämlich mit einer HTTP-Webseite, werden die übermittelten Daten nicht verschlüsselt, sondern im **Plaintext** gesendet. Auch bei Verwendung von Tor ändert sich dies nicht, da der Traffic nur innerhalb von Tor – also nur von Node zu Node - verschlüsselt wird. Angreifer, die zwischen der Exit-Node und dem Server andocken, können den Traffic ohne Probleme mitschneiden und auch verändern. Wenn dann userspezifische Informationen wie Mail-Adressen oder Standorte gesendet werden, wird die Schutzfunktion von Tor außer Kraft gesetzt.

7.6 JavaScript Zero-Day-Schwachstellen

JavaScript, kurz JS, ist eine Java-verwandte Programmiersprache, die für Webservices verwendet wird. Der wichtige Punkt hierbei ist, dass JS eine clientseitige Scriptsprache ist - d.h., sie wird auf dem Computer des Clients ausgeführt.[34] Dies bringt in der der Webentwicklung viele Vorteil mit sich, in Sachen Sicherheit jedoch viele Nachteile. Von Angreifern richtig programmiert wird JS leicht zum Einfallstor. Es ist zum Beispiel möglich, durch JS eine Verbindung mit einem Server aufzubauen, ohne den Traffic dabei durch Tor gehen zu lassen. Auf diesem Wege wird dann die „wahre“ IP-Adresse des Users übermittelt, wodurch seine Anonymität nicht mehr gewährleistet ist.

Es ist auch möglich, den auf der Website verwendeten JS-Code zu hacken. Genauer gesagt werden sog. Zero-Day-Schwachstellen, also Sicherheitslücken in verwendeten JS-Scripts und Bibliotheken, die bekannt, aber noch nicht **gepatcht** sind, ausgenutzt.[35] Ein Beispiel für eine solche Sicherheitslücke ist Log4J - eine

[34] Vgl. AWS-Amazon

[35] Vgl. Ablon, Bogard 2017, S.iX

Schwachstelle in Apaches Protokollier-Script, welche es Angreifern 2022 ermöglichte, Pakete mit eigenem Code intern in den Server einzuschleusen und dort auszuführen.[36]
Ziel all dieser Attacken ist es, Informationen über den User herauszufinden, um diesen eventuell belangen bzw. bestrafen zu können.

7.7 Firefox Zero-Day-Schwachstellen

Der Tor-Browser ist wie in Kap. 5 beschrieben eine Nutzeroberfläche, die sich des Tor Netzwerks bedient. Die Benutzeroberfläche an sich besteht aus einer abgewandelten Version des Firefox-Browsers mit zusätzlichen Funktionen. Wegen der einfachen Bedienung der Oberfläche ist Tor für technisch weniger erfahrene Personen leichter zugänglich. Allerdings öffnet sich hier auch eine weitere Angriffsfläche, wodurch Tor-User de-anonymisiert werden können. 2013 gelang es dem FBI, Eric Eoin Marques, den Betreiber von Freedom Hosting, dem damals größten Host von Darknet-Seiten, zu verhaften.[37] Nach seiner Verhaftung war auf allen gehosteten Seiten folgende Fehlermeldung zu sehen: „Down for maintenance“. Im Hintergrund wurde währenddessen schädlicher JavaScript-Code ausgeführt, der durch eine Schwachstelle im Firefox-Browser selbst die wahren IP-Adressen, die MAC-Adressen sowie Seriennummern von Geräten der Besucher offenlegte. Später gab das FBI in einer Mitteilung bekannt, dass es den Hack ausgeführt habe.[38] In der Folge wurden zahlreiche Besucher und auch einige Hoster dieser Webseiten verhaftet.[39]

7.8 „Tor Stinks“ – NSA und Tor

Die NSA ist als Sicherheitsbehörde der USA sehr daran interessiert, Tor-User zu de-anonymisieren und hat demensprechend weitgehende Forschungen zum Thema „Tor“ betrieben. Eines der veröffentlichten Dokumente der Edward Snowden Leaks beinhaltet eine Top-Secret Präsentation der NSA über den damals aktuellen

[36] Vgl. Gallo 2022

[37] Vgl. The Infographics Show 2020, TC 00:00:46-00:02:20

[38] Vgl. Poulsen Kevin 2013

[39] Vgl. The Infographics Show 2020, TC 00:00:46-00:02:20

Forschungsstand, die in weiterer Folge von der Zeitung „The Guardian“ veröffentlich wurde.[40] In den folgenden Unterkapiteln werden einzelne Punkte der Präsentation vorgestellt, allerdings sind die betreffenden Informationen teilweise unvollständig, da Abkürzungen und Codenamen verwendet werden, welche der Öffentlichkeit noch nicht bekannt sind.

So schreibt die NSA:

> „We will never be able to de-anonymize all Tor users all the time.
> With manual analysis we can de-anonymize a very small fraction of Tor users, however, no success de-anonymizing a user […] on demand.“[41]

7.8.1 Cookie Leakage

Wenn eine Internetseite besucht wird, werden standardmäßig sogenannte Cookies übermittelt. Cookies sind kleine Textdateien, die lokal auf dem Computer des Users gespeichert werden. Sie können verschiedene Arten von Informationen enthalten, z. B. Anmeldeinformationen, Präferenzen, Einstellungen und Verhalten des Nutzers.[42] Die NSA fand nun heraus, dass manche Cookies eine Tor-Sitzung „überleben“, also nach dem Beenden von Tor immer noch auf dem Computer gespeichert bleiben. Auf diesem Wege könnte man, sobald der User nicht mehr den Tor-Browser - sondern z.B. Chrome – verwendet, nach diesen Cookies suchen und nach deren Auffindung die wahre IP-Adresse aufdecken. Ein Beispiel für so ein überlebendes Cookie ist „DoubleClickID“.

Allerdings wird diese Schwachstelle erst zur Gefahr, wenn veraltete Versionen des Tor Browsers oder eigene Programme verwendet werden, um Tor zu nutzen. Torbutton und Tor Browser (Bundle) löschen Cookies nach jeder Sitzung automatisch. Eine weitere Möglichkeit zur Verhinderung derartiger Angriffsflächen ist das Benützen eines „sicheren“ Betriebssystems wie Tails oder Qubes OS, bei denen nach jedem Herunterfahren alle Daten permanent gelöscht werden.[43]

---

[40] Vgl. The Guardian 2013

[41] The Guardian 2013, S.2

[42] Vgl. Searchmetrics

[43] Vgl. Geiger 2022

7.8.2 Ländergebundene Analyse

Unter „Goes Inta Goes Outta/Low Latency“ wird im geleakten Dokument eine Möglichkeit beschrieben, wie durch Überwachung des Datenverkehrs in einem Land Tor-User aufgedeckt werden können. Wird eine für die NSA verdächtige Person, die Tor nutzt, ausfindig gemacht, werden im vermuteten Aufenthaltsland des Users alle Verbindungen in das Tor-Netzwerk analysiert und dokumentiert. Tritt nun irgendwo im Internet ein Ereignis auf, dessen Urheber der Verdächtige sein könnte, wird die Zeit des Auftretens des Ereignisses mit den Zeiten der dokumentierten Tor-Verbindungen verglichen. Ziel ist es, Muster und Übereinstimmungen zu finden und Aktionen nachzuvollziehen. Um dies bewerkstelligen zu können, braucht es aber viel Überwachung, die nur von staatlichen Instanzen realisiert werden kann.

7.8.3 Tor Node Überflutung

Eine weitere Idee der NSA, die darauf abzielt, das Tor-Netzwerk an sich anzugreifen, ist es, viele Nodes aufzustellen, die vorgeben, eine sehr hohe Bandbreite zu haben. Tatsächlich haben diese Nodes nur eine sehr geringe Bandbreite und könnten somit das gesamte Netzwerk schwächen. Die NSA gibt aber selbst zu bedenken, dass es vielleicht kontraproduktiv wäre, Tor-User zu verschrecken und sie zum Umsteigen auf Tor-Alternativen zu bewegen. Aus diesem Grund richten sich Attacken gegen Tor deshalb eher auf einzelne Tor-User als gegen das gesamte Tor-Netzwerk.

7.9 Harvest now – Decrypt later

> “…refers to a proactive approach taken by potential adversaries who […] collect encrypted data today with the intention of decrypting it in the future when quantum computers are powerful enough to break existing encryption methods.“[44]

Wie das hier aufgeführte Zitat erklärt, besteht eine Angriffsmöglichkeit darin, verschlüsselten Traffic abzufangen und ihn dann in Zukunft mittels Quantencomputer zu entschlüsseln. Quantencomputer sind aufgrund ihrer speziellen Funktionsweise, die sich von der eines „klassischen“ Computers

[44] Vgl. AuCloud 2023

unterscheidet, in der Lage, alle heute angewendeten, asymmetrischen Verschlüsselungsmethoden wie RSA oder DH zu knacken. Die Details würden den Rahmen sprengen, grundlegend geht es aber darum, dass Quantencomputer **Primfaktorzerlegungen** um einiges schneller berechnen können als heutige Computer. Ein 2048-Bit-RSA-Schlüssel *RK*, der einfach ausgedrückt mit *RK=p*g* berechnet wird, kann dadurch um einiges schneller in seine zwei Primfaktoren p und g zerlegt werden.[45] Als Vergleich: zum Knacken von *RK* braucht ein klassischer Computer circa 300 Billionen Jahre.[46] Ein Quantencomputer mit 10.000 **QBits** hingegen braucht nur 104 Tage[47] – wobei 10.000 nicht viel sind. IBM zum Beispiel forscht schon an einem 100.000 QBits Quantencomputer.[48]

2009 war die allgemeine Meinung der Kryptografie-Community, dass dieser Angriffsvektor nie realistisch anwendbar sei. Gestützt wurde diese Meinung u.a. auch von der Tatsache, dass die Entwicklung eines funktionierenden Quantencomputers bis dato noch weit in der Zukunft lag. Schätzungen zufolge werde diese Technologie den Geheimdiensten dagegen schon in den nächsten drei bis fünf Jahren zur Verfügung stehen – in 10 Jahren auch eingeschränkt der Öffentlichkeit.[49] Dementsprechend gehört dieser Angriff heutzutage zu den größten Cyber-Bedrohungen, da die erfolgreiche Ausführung eines solchen Angriffs immer wahrscheinlicher wird.[50] Betrachtet man nun ein Szenario, in welchem Tor mit dieser Methode angegriffen wird, wären die Folgen katastrophal. Einerseits wäre der gesamte Traffic offengelegt und somit die Geheimhaltung verloren. Andererseits würde jeder User seine Anonymität verlieren, da man mithilfe des entschlüsselten Traffics die genaue Route des Circuits ermitteln könnte.

Allerdings stellen sich folgende Fragen:

- Wie wahrscheinlich ist es wirklich, dass funktionierende Quantencomputer innerhalb der nächsten Jahre entwickelt werden?

[45] Vgl. Collins 2023

[46] Vgl. QuintessenceLabs 2019

[47] Vgl. Gent 2023

[48] Vgl. Brooks 2023

[49] Vgl. Baker 2022

[50] Vgl. QNu Labs 2020

- Werden die jetzt gesammelten Daten bis zum Zeitpunkt, ab dem sie entschlüsselt werden können, noch brauchbar sein?

Duran setzte sich mit diesen Fragen auseinander, kam aber zu dem generellen Schluss, dass man auf jene Fragen noch keine Antworten geben, sondern nur spekulieren kann.[51]

7.10 Menschliche Bedienungsfehler – Kompetenzschwächen

> "Die größte Schwachstelle ist der Mensch"[52]

In den meisten Fällen werden Nutzer nicht direkt aufgrund von Systemschwächen in Tor zurückverfolgt. Oftmals geben sie selbst den Angreifern durch eigene Fehler Gelegenheiten, sie aufzuspüren. Die häufigsten von Usern begangenen Fehler sind:

- Preisgabe von persönlichen Daten:
  Der fatalste Fehler, der begangen werden kann, ist es, persönliche Daten preiszugeben, die Rückschlüsse auf die wahre Identität des Users erlauben. Angreifer besitzen dann nämlich eine Spur, die direkt zum Benutzer führt, und müssen dafür nicht einmal in Cyber-Security gut aufgestellt – was eine große Herausforderung für viele Angreifer darstellt. Beispiele für persönliche Daten sind u.a. der echte Name, die Wohnadresse, das Geschlecht, das Alter, Kreditkarteninformationen und persönliche Bilder.
- Aktivierung von JavaScript:
  Zwei der großen Angriffspunkte bei Tor – JS- und Firefox-Zero-Day-Schwachstellen – können für Attacken nur genutzt werden , wenn JavaScript im Browser aktiviert ist. Extra hierfür existiert ein eigener Schalter (siehe Kap. 8). Das Problem hierbei ist, dass dieser Schalter oft von sowohl unerfahrenen Personen durch Unwissen als auch von erfahrenen aus Nachlässigkeit oder Vergesslichkeit nicht benützt wird. Der Benutzer allein trägt daher die gesamte Verantwortung für die Durchführung dieses Schrittes sowie das damit verbundene Risiko bei Unterlassung.

[51] Vgl. Duran 2023

[52] Vgl. Alperovitch 2019

- Bezahlung mit Kryptowährungen:
  Im Darknet erfolgen Zahlungen fast ausschließlich mit Kryptowährungen, meist mit Bitcoin. Viele User erachten die Zahlungen mit einer Kryptowährung als sicher. Jedoch können die Zahlungen von Ermittlungsbehörden zurückverfolgt werden. Die Details über die Bewerkstelligung der Rückverfolgung würden den Rahmen sprengen, deshalb wird diese Thematik an dieser Stelle nicht näher erörtert. Hinzu kommt, dass die Methode jener Zurückverfolgung von Bezahlvorgängen auch auf viele andere Kryptowährungen übertragbar ist, wodurch ein Umstieg auf eine andere Währung ebenfalls keine Lösung ist.[53] [54]
- Einloggen mit persönlichen Konten:
  Das Angeben seiner persönlichen Accounts wie z.B. Gmail-, Facebook- oder AppleID-Konten kann u.a. zu Rückschlüssen auf die wahre Identität des Benutzers führen. Gleich wie bei der Preisgabe von persönlichen Daten macht die Preisgabe persönlicher Konten den durch Tor gegeben Schutz obsolet.

[53] Vgl. Biselli 2019

[54] Vgl. Bitpanda o.J.

8 Abwehr von Schwachstellen

„If they want to get you, they get you in time."[55]

In dieser Arbeit bezieht sich der Begriff "Abwehr von Schwachstellen" auf potenzielle Sicherheitslücken, die durch Handlungen des Benutzers, wie beispielsweise das regelmäßige Aktualisieren von Software, proaktiv vermieden werden können.

Vor dem Hintergrund dieses Verständnisses können nur wenige der in Kapitel 7 aufgelisteten Schwachstellen abgewendet werden.

Um *Endpoint Traffic Correlation* zu unterbinden, müsste jede Anfrage durch den Client und den Server speziell so bearbeitet werden, dass insgesamt keine Muster erkennbar sind. Ein von der Community vorgestelltes Projekt wie LoopTor löst dieses Problem, indem Anfragen noch auf dem Rechner des Clients so bearbeitet werden, dass diese alle gleich viele Daten enthalten und somit identisch aussehen.[56] Dieses Projekt ist jedoch noch nicht in Tor implementiert, sondern bleibt vorerst nur ein Vorschlag - der Angriffspunkt bleibt also weiterhin bestehen und kann durch den User nicht geblockt werden. Dasselbe gilt in Bezug auf den *Sybil Angriff*, die *Tor Node Availability*, die *Zensur-freie Veröffentlichung*, die *Harvest now – Decrypt Later* - Methode und alle *Tor-Stinks*-Schwachstellen.

Schließlich bleiben drei Schwachstellen übrig, die der User vermeiden kann: *JavaScript*- und *Firefox-Zero-Day-Schwachstellen* sowie *Bedienungsfehler*. Um die ersten beiden Schwachstellen zu beheben, reicht es, JavaScript im Browser vollständig zu blockieren. Um dies zu ändern, muss (auf der PC-Version des Tor-Browsers) das Schutzschild-Icon rechts oben gedrückt, und auf „Am sichersten" gestellt werden. Danach sollte der Schild schwarz ausgefüllt sein. Bedienungsfehlern hingegen kann größtenteils nur durch ein vorsichtiges Handeln, die eingehende Befassung mit den existierenden Schwachstellen sowie durch Weiterbildung entgegengewirkt werden.

Dennoch gibt es für zwei angesprochene Probleme in Kap. 7.10 Lösungen. Erstens, Bezahlungen im Darknet. Obwohl sie zu vermeiden sind, werden sie immer mehr

[55] Snowden 2013, TC 00:06:16–00:06:20

[56] Vgl. Planas i Planas 2020, S.23

durch einen bestimmten Coin angeboten – Monero. Diese Kryptowährung ist speziell für den Zweck entwickelt worden, Transaktionen und Benutzer praktisch nicht zurückverfolgbar zu machen. Daher gilt sie momentan als eine sichere Methode zur Bezahlung im Darknet.[57] Was das Problem der Verwendung eigener Accounts betrifft, so lässt sich dieses durch die Verwendung von anonymen Konten, die speziell für die Nutzung von Tor geschaffen wurden, lösen. Derartige anonyme Konten werden z. B. von Proton-Mail oder Mail2Tor angeboten.[58]

[57] Vgl. Biselli 2019

[58] Vgl. Stosh 2023

9 Zusammenfassung

Aufgrund der durch Tor bewerkstelligten Verschlüsselung des Traffics wird es außenstehenden Personen unmöglich gemacht, diesen mitzulesen. Zudem werden die IP-Adressen der User geschickt verschleiert, was ein Zurückverfolgen beinahe unmöglich macht. Allerdings gibt es auch einige Schwachstellen und Angriffsflächen, mit denen User dennoch de-anonymisiert werden können. Diese betreffen aber fast nie direkt das Tor Netzwerk, sondern die Tools, mithilfe deren auf das Tor-Netzwerk zugegriffen wird. Ein Beispiel hierfür ist der Tor-Browser. Ungeachtet dessen ist in erster Linie der Benutzer selbst zu einem großen Teil für seine Sicherheit verantwortlich, da die meisten Angriffsflächen in Bezug auf die Einstellungen von Tools, die Benutzer nicht richtig konfiguriert haben, gegeben sind.

Insgesamt erweist sich Tor als eine perfekte und sichere Lösung, um sowohl im Clearnet als auch im Darknet anonym zu browsen. Dies gilt jedoch ausschließlich unter den Voraussetzungen, dass sämtliche Sicherheitsmaßnahmen getroffen werden.
Zwei Fragen müssen jedoch ungeklärt belassen werden.

1. Wie genau finden die Nodes heraus, ob eine Zelle noch verschlüsselt ist oder nicht und
2. wie schafft es Tor, dass sich die Größe der jeweiligen Zellen nicht verändert, wenn ein Layer Verschlüsselung dazugegeben oder entfernt wird.

Die Beantwortung dieser Fragen war nicht möglich, weil die Dokumentation von Tor größtenteils komplizierte und teils unverständliche Erklärungen aufweist. Hinzu kommt, dass keine anderen Quellen gefunden wurden, die diesbezüglich detaillierte Erläuterungen enthalten.
Um Tor noch besser verstehen zu können, wäre es wichtig, die Beweggründe für die Auswahl bestimmter Verschlüsselungsmethoden und die Ursachen für Unterschiede in ihrer Effizienz zu untersuchen. Dafür müsste jedoch eine eingehende Beschäftigung mit den mathematischen Grundlagen der kryptografischen Prozesse erfolgen. Eine derartige Beschäftigung würde allerdings den Rahmen dieser Arbeit weit überschreiten und wurde daher nicht ausgeführt.

10 Literaturverzeichnis

Ablon Lillian, Bogart Andy: Zero Days, Thousands of Nights. The Life and Times of Zero-Day Vulnerabilities and Their Exploits, Santa Monica - California – USA: RAND Corporation 2017, https://www.google.at/books/edition/Zero_Days_Thousands_of_Nights/UaaSDgAAQBAJ?hl=de&gbpv=1&dq=zero%20day%20vulnerabilities&pg=PR9&printsec=frontcover (Zugriff: 10.8.2023)

Alperovitch Dimitri: "Die größte Schwach-stelle ist der Mensch", https://www.sueddeutsche.de/wirtschaft/internetsicherheit-die-groesste-schwach-stelle-ist-der-mensch-1.4338184, 20.2.2019 (Zugriff: 27.8.2023)

Aragon Jonah: Tor Overview, https://www.privacyguides.org/en/advanced/tor-overview/, 12.3.2023 (Zugriff: 27.6.2023)

Årnes André: Cyber Investigations, USA: John Wiley & Sons, Inc 2023, https://books.google.at/books?id=c9GPEAAAQBAJ& (Zugriff: 9.8.2023)

AuCloud: “Harvest Now – Decrypt Later” Threat: Safeguarding Today’s Data for Tomorrow’s Quantum Era and the Role of Symmetric Keys. https://www.australiacloud.com.au/blogs/harvest-now-decrypt-later-threat/ 14.8.2023 (Zugriff: 18.10.2023)

AWS-Amazon: Was ist JavaScript?, https://aws.amazon.com/de/what-is/javascript/ (Zugriff: 10.8.2023)

Baka Paul, Schatten Jeremy: SSL/TLS Under Lock and Key. A Guide to Understanding SSL/TLS Cryptography, Sydney Australien, Keyko 2020, https://www.google.at/books/edition/SSL_TLS_Under_Lock_and_Key/344OEAAAQBAJ?hl=de&gbpv=0 (Zugriff: 28.6.2023)

BBC: Anonymous activists target Tunisian government sites, https://www.bbc.com/news/technology-12110892, 4.1.2011 (Zugriff: 23.8.2023)

Binance Acadamy: Sybil Attacks Explained, https://academy.binance.com/en/articles/sybil-attacks-explained, 7.12.2018 (Zugriff: 9.8.2023)

Biselli Anna: Blockchain-Forensik: Wer steckt hinter einer Bitcoin-Zahlung? https://netzpolitik.org/2019/blockchain-forensik-wer-steckt-hinter-einer-bitcoin-zahlung/, 2.8.2019 (Zugriff: 9.9.2023)

Bitpanda: Was ist das Darknet und was hat es mit Bitcoin zu tun? https://www.bitpanda.com/academy/de/lektionen/was-ist-das-darknet-und-was-hat-es-mit-bitcoin-zu-tun/ (Zugriff: 9.9.2023)

Brooks Michael: IBM wants to build a 100,000-qubit quantum computer, https://www.technologyreview.com/2023/05/25/1073606/ibm-wants-to-build-a-100000-qubit-quantum-computer/, 25.5.2023 (Zugriff: 1.11.2023)

Bundeskriminalamt (BKA; Deutschland): Die Tiefen des Internets, https://www.bka.de/SharedDocs/Downloads/DE/AktuelleInformationen/Infografiken/Infografiken_Internetkriminalitaet/infografikDieTiefenDesInternets.html, 12.10.2016 (Zugriff: 31.5.2023)

Capers, Zach/Pavlakoudis, Rosalia: 4 moderne Verschlüsselungsverfahren in der Übersicht. So schützt du sensible Daten vor neugierigen Blicken, https://www.getapp.de/blog/2307/moderne-verschlusselungsverfahren-ubersicht, 16.11.2021 (Zugriff: 25.2.2023)

Collins Misty: How Quantum Computers Break Encryption - Shor's Algorithm, https://www.qubitlogger.com/post/can-quantum-computers-really-break-encryption, Feber 2023 (Zugriff: 10.10.2023)

Cryptography Docs: Cryptography Docs, https://cryptography.io/en/latest/, 2023 (Zugriff: 1.11.2023)

Das Abhijit, Veni Madhavan C. E.: Public-key Cryptography: Theory and Practice, Dheli Indien, Dorling Kinderslay (India) 2009, https://www.google.at/books/edition/Public_key_Cryptography/fzoiOeUf8fIC?hl=de&gbpv=0 (Zugriff: 30.6.2023)

DIE ZEIT: Hintertür für Spione. https://www.zeit.de/1998/39/199839.c_krypto_.xml, 17.9.1998 (Zugriff: 23.8.2023)

Duran Jeffrey: Guest Post: Harvest Now, Decrypt Later? The Truth Behind This Common Quantum Theory, https://thequantuminsider.com/2023/02/07/guest-post-harvest-now-decrypt-later-the-truth-behind-this-common-quantum-theory/, 7.2.2023 (Zugriff: 18.10.2023)

Elektronik Kompendium: Diffie-Hellman-Merkle-Schlüsselaustausch, https://www.elektronik-kompendium.de/sites/net/1909031.htm (Zugriff: 26.7.2023)

Elektronik Kompendium: Diffie-Hellman-Merkle-Schlüsselaustausch, https://www.elektronik-kompendium.de/sites/net/bilder/19090311.gif (Zugriff: 26.7.2023)

Elektronik Kompendium: Perfect Forward Secrecy, https://www.elektronik-kompendium.de/sites/net/1809181.htm (Zugriff: 17.10.2023)

Elektronik Kompendium: Symmetrische Kryptografie (Verschlüsselung), https://www.elektronik-kompendium.de/sites/net/1910101.htm (Zugriff: 28.6.2023)

Evans Jaq: What is Perfect Forward Secrecy? https://www.extrahop.com/company/blog/2017/what-is-perfect-forward-secrecy/, 9.8.2017 (Zugriff: 17.10.2023)

Gallo Katlyn: Log4J Vulnerability Explained: What It Is and How to Fix It, https://builtin.com/cybersecurity/log4j-vulerability-explained, 9.23.2022 (Zugriff: 12.8.2023)

Geiger Jörg: Der Anti-NSA-PC: Tails Linux perfekt einrichten, https://www.chip.de/artikel/Tails-Linux-Der-Anti-NSA-PC-System-einrichten_139935821.html, 20.12.2022 (Zugriff: 10.8.2023)

Gent Edd: Quantum Computers Could Crack Encryption Sooner Than Expected With New Algorithm, https://singularityhub.com/2023/10/02/quantum-computers-could-crack-encryption-sooner-than-expected-with-new-algorithm/, 2.10.2023 (Zugriff: 1.11.2023)

Grömer Julia: Anonymität im Darknet: Nutzer und Usability im Vergleich der drei Hauptvertreter, Bachelorarbeit, Anhalt University of Applied Scieneces (Deutschland) 2020, https://opendata.uni-halle.de/bitstream/1981185920/12783.2/2/Bachelorarbeit%202020%20Julia%20G%C3%B6rmer.pdf (Zugriff: 8.1.2024)

Kehrer Paul: Cryptography, https://pypi.org/project/cryptography/, 24.10.2023 (Zugriff: 1.11.2023)

Luber Stefan: Definition Diffie-Hellman key exchange, Was ist der Diffie-Hellman-Schlüsselaustausch? https://www.security-insider.de/was-ist-der-diffie-hellman-schluesselaustausch-a-799443/, 18.02.2019 (Zugriff: 25.7.2023)

Hearst Magazines: How our government spies on you. (= Popular Mechanics Band 178, Nr. 4), New York USA, Hearst Magazines 2001, https://books.google.at/books?id=hM8DAAAAMBAJ&pg=PA68 (Zugriff: 21.8.2023)

Pike Sarah: Darknet, Darkweb, Deep Web und Surface Web — was ist der Unterschied? https://www.kaspersky.de/blog/deep-web-dark-web-darknet-surface-web-difference/26169/, 2.2.2021 (Zugriff: 31.5.2023)

Planas i Planas Jaume: LOOPTOR. Fighting traffic correlation on the Tor network, Bachelorarbeit, Escola T`ecnica d'Enginyeria de Telecomunicaci´o de Barcelona Universitat Polit`ecnica de Catalunya 2020, https://upcommons.upc.edu/bitstream/handle/2117/329579/TFG_JAUME_PLANAS.pdf?sequence=3&isAllowed=y (Zugriff: 10.9.2023)

Porup J.M.: What is the Tor Browser? And how it can help protect your identity. https://www.csoonline.com/article/3287653/what-is-the-Tor-browser-how-it-works-and-how-it-can-help-you-protect-your-identity-online.html, 15.10.2019 (Zugriff: 1.6.2023)

Poulsen Kevin: FBI Admits It Controlled Tor Servers Behind Mass Malware Attack, https://www.wired.com/2013/09/freedom-hosting-fbi/, 13.9.2013 (Zugriff: 24.8.2023)

Pound Mike: How TOR Works- Computerphile [YouTube-Video], https://www.youtube.com/watch?v=QRYzre4bf7I&ab_channel=Computerphile, 31.05.2017 (Zugriff: 31.7.2023)

Pound Mike: TLS Handshake Explained – Computerphile [YouTube-Video], https://www.youtube.com/watch?v=86cQJ0MMses&ab_channel=Computer5645456564phile, 06.11.2020 (Zugriff: 2.7.2023)

QNu Labs: Dark Side of Quantum Computers A Lurking Threat to National Security. https://www.qnulabs.com/dark-side-of-quantum-computers-a-lurking-threat-to-national-security/ 18.12.2020 (Zugriff: 18.10.2023)

QuintessenceLabs: Breaking RSA Encryption - an Update on the State-of-the-Art, https://www.quintessencelabs.com/blog/breaking-rsa-encryption-update-state-art, 13.6.2019 (Zugriff: 1.10.2023)

Saxena Aditya: Onion Routing, https://www.scaler.com/topics/onion-routing/, Bearbeitet: 4.5.2023 (Zugriff: 31.7.2023)

Searchmetrics: Cookies. Was sind Cookies und wozu werden sie eingesetzt?, https://www.searchmetrics.com/de/glossar/cookies/, (Zugriff: 10.8.2023)

Snowden Edward: Edward Snowden, NSA files source: 'If they want to get you, in time they will' [Video], The Guardian 2013 (Zugriff: 9.9.2023)

Stosh Brandon: List Of Secure Dark Web Email Providers In 2023, https://freedomhacker.net/list-of-secure-dark-web-email-providers-in-2016-4946/, 1.9.2023 (Zugriff: 10.9.2023)

THE CYBERSECURITY MAN, HTTPS: The TLS Handshake Using Diffie-Hellman Ephemeral, https://thecybersecurityman.com/2018/04/25/, 25.4.2018 (Zugriff: 2.8.2023)

theguardian.com: Tor Stinks, www.theguardian.com/world/interactive/2013/oct/04/tor-stinks-nsa-presentation-document, Freitag 4 Oktober 2013 (Zugriff:10.8.2023)

The Infographics Show: This Is How the FBI Will Catch You on the Dark Web [YouTube], https://youtu.be/p2IC4VQcxyo?t=46, 27.06.2020 (Zugriff: 24.8.2023)

The Tor Project, Inc: History, https://www.Torproject.org/about/history/ (Zugriff: 25.2.2023)

The Tor Project, Inc: Types Of Relays On The Tor Network, https://community.torproject.org/relay/types-of-relays/ (Zugriff: 27.6.2023)

The Tor Project, Inc: Tor: The Second-Generation Onion Router, https://svn-archive.torproject.org/svn/projects/design-paper/tor-design.html#subsec:circuits (Zugriff: 4.8.2023)

Thoma Jürgen: Tor und die große chinesische Firewall, https://www.golem.de/news/internetzensur-tor-und-die-grosse-chinesische-firewall-1410-109828.html, 14. Oktober 2014 (Zugriff: 6.8.2023)

Web Archive: Arab springtime: Is the web reaching new heights? https://web.archive.org/web/20160303181007/http://en.rsf.org/the-new-media-between-revolution-11-03-2011,39764.html, 11.3.2011 (Zugriff: 23.8.2023)

Wright A. J.: The Dark Web. The Unseen Side of the Internet, United States: A. J. Wright 2020, https://www.google.at/books/edition/The_Dark_Web/jPTKDwAAQBAJ?hl=de&gbpv=1 (Zugriff: 1.6.2023)

11 Abbildungsverzeichnis

12 Glossar

Begriff	Definition
CAPTCHA	Ein Sicherheitsmechanismus, der entwickelt wurde, um zwischen menschlichen Benutzern und automatisierten Programmen zu unterscheiden.
Ciphertext [/ˈsaɪ.fər/]	Kryptographisch verschlüsselter Text
Crawler	Ein automatisiertes Programm zur Indexierung von Webseiten (für Suchmaschinen).
DDoS	Kurz für Distributed Denial of Service. Sind Cyberangriffe auf Web-Services, mit dem Ziel, diese ausfallen zu lassen.
Dynamische IP	Sind IPs, die sich nach jedem Verbindungsaufbau ändern.
Exception	Eine unerwartete Störung im Programmablauf, die durch Fehler oder ungewöhnliche Situationen auftritt.
Google-Dorking Google-Hacking	Google Dorking oder Google Hacking bezieht sich auf die Nutzung spezieller Suchparameter in Google, um versteckte oder anfällige Informationen im Internet zu finden. Ein Beispiel für eine Anwendung, um speziell nur Seiten von bg-sillgasse.tsn.at anzuzeigen, ist die Query: *„site:bg-sillgasse.tsn.at“.*
Hop	Der „Sprung“ von einer Node zur anderen
Hash	Ein Hash Algorithmus wandelt Daten in eine eindeutige, nicht umkehrbare Zeichenfolge um.
Metadaten	Sind Informationen wie Titel, Beschreibung, Schlüsselwörter, Sprache, Encodierung
Node	Server im Tor-Netzwerk, der Datenverkehr anonymisiert, indem er als Eingangs-, Mittel- oder Ausgangsknoten agiert. Von der Funktionsweiße fast dasselbe wie ein Proxy.

Open Source	Bezieht sich auf Software, deren Quellcode für die Öffentlichkeit frei verfügbar ist und von der Gemeinschaft weiterentwickelt, überprüft und verbessert werden kann.
Plaintext	Kryptographisch unverschlüsselter Text
Proxy	Ein Vermittler zwischen einem Client und einem Zielserver. Agiert als Zwischenstation und nimmt Anfragen vom Client entgegen, leitet sie an den Zielserver weiter und gibt die Antwort des Servers an den Client zurück. Dabei verbirgt der Proxy die tatsächliche IP-Adresse des Clients.
QBit	QBit (Quanten-Bit): Die grundlegende Informationseinheit in der Quanteninformatik. Im Gegensatz zum klassischen Bit kann ein QBit mehrere Zustände gleichzeitig repräsentieren, was Quantencomputern ihre Schnelligkeit verleiht.
Sitzungsschlüssel	Ist ein zufällig generierter kryptographischer Schlüssel, der zur Verschlüsselung des Traffics zwischen zwei Instanzen verwendet wird. Der Schlüssel ist einmalig, wird als nur für eine einzige Sitzung verwendet.
Statische IP	Sind IPs, die immer gleichbleiben
Whistleblower	Eine Person, die für die Öffentlichkeit wichtige Informationen aus einem geheimen oder geschützten Zusammenhang veröffentlicht.

Selbstständigkeitserklärung

Ich erkläre, dass ich diese vorwissenschaftliche Arbeit eigenständig angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benutzt habe.

_____ _____

Ort, Datum Unterschrift

Zustimmung zur Aufstellung in der Schulbibliothek

Ich gebe mein Einverständnis, dass ein Exemplar meiner vorwissenschaftlichen Arbeit in der Schulbibliothek meiner Schule aufgestellt wird.

_____ _____

Ort, Datum Unterschrift